



HEITOR SILVÉRIO LOPES E  
RICARDO HIROSHI CALDEIRA TAKAHASHI

# Computação Evolucionária em Problemas de Engenharia



[www.omnipax.com.br](http://www.omnipax.com.br)

**omnipax**  
editora

Heitor Silvério Lopes  
Ricardo Hiroshi Caldeira Takahashi  
(Editores)

# Computação Evolucionária em Problemas de Engenharia



2011

Copyright ©2011 Omnipax Editora Ltda  
Caixa Postal: 16532 - 81520-980 Curitiba, PR



A editora disponibiliza por acesso livre a versão eletrônica deste livro no *site*: <http://www.omnipax.com.br>, sob uma licença *Creative Commons Attribution 3.0*.

**Capa:**

Sérgio Alexandre Prokofiev

**Projeto gráfico e editoração:**

Omnipax Editora Ltda

**Impressão:**

Pix Bureau Gráfica e Editora

**Ficha catalográfica:**

Adriano Lopes (CRB9/1429)

Dados Internacionais de Catalogação na Publicação

C738 Computação evolucionária em problemas de engenharia/  
organização: Heitor Silvério Lopes e Ricardo Hiroshi Cal-  
deira Takahashi. – Curitiba, PR: Omnipax, 2011.  
385 p.: il.

Vários autores

Inclui biografias

ISBN: 978-85-64619-00-5

e-ISBN: 978-85-64619-04-3

1. Inteligência artificial. 2. Computação evolutiva. 3.  
Algoritmos genéticos. 4. Algoritmos de computador. 5. En-  
genharia auxiliada por computador. I. Lopes, Heitor Silvério,  
org. II. Takahashi, Ricardo Hiroshi Caldeira, org. III. Título

CDD (22. ed.) 006.3

CDU 006.32

005.1

O projeto de sistemas de engenharia é um campo que requer, por um lado, o conhecimento relacionado com cada disciplina de engenharia (por exemplo: mecânica dos fluidos, eletricidade, cálculo estrutural, etc), que constituem a base para a construção de dispositivos ou sistemas que satisfaçam às necessidades de cada projeto. Por outro lado, requer também o uso de técnicas capazes de tratar o grande número de possíveis soluções de projeto, trazendo à luz as poucas soluções possíveis que podem ser chamadas de ótimas, ou seja, que representam sistemas com bom desempenho, econômicos e confiáveis. Estas técnicas são chamadas de técnicas de otimização. À medida em que avança a tecnologia, torna-se cada vez mais importante o uso da otimização, devido ao aumento da complexidade dos sistemas a serem projetados, o que torna virtualmente impossível o projeto sem o uso direto ou indireto destas técnicas.

Nas últimas décadas, pesquisadores procuraram desenvolver novas abordagens para problemas de otimização, tendo como inspiração os mecanismos de adaptação biológica presentes da natureza. Foram desenvolvidos, por exemplo, novos algoritmos inspirados na teoria da evolução das espécies, ou na observação de colônias de insetos buscando alimento, ou ainda no sistema imunológico dos mamíferos defendendo o organismo contra invasores. Tais algoritmos, coletivamente conhecidos como técnicas de computação evolucionária, tem revelado grande eficiência para resolver satisfatoriamente inúmeros problemas de projeto de engenharia. Atualmente, tais algoritmos fazem parte do conjunto de ferramentas que se associam a praticamente todo projeto de sistemas de “alta tecnologia”.

Dentre as técnicas utilizadas para a resolução de problemas de engenharia destacam-se os Algoritmos Genéticos. Entretanto, outras técnicas evolucionárias (evolutivas) também tem sido utilizadas em alguns casos, podendo-se citar: Otimização por Colônias de Formigas (ACO), Otimização por Enxame de Partículas (PSO), Colônia de Abelhas Artificiais (ABC), Busca Harmônica (HS) e Evolução Diferencial (DE), além de diversos algoritmos evolutivos híbridos.

No Brasil há um grande número de grupos de pesquisa em diferentes instituições acadêmicas aplicando técnicas de computação

evolucionária em problemas de projeto de engenharia. Isto mostra a maturidade da área no Brasil, e este livro apresenta um panorama de trabalhos recentes desenvolvidos por pesquisadores brasileiros nesta área estratégica do conhecimento, incluindo aplicações nas engenharias: elétrica, nuclear, mecânica, de controle e de produção.

Heitor Silvério Lopes – UTFPR

Ricardo Hiroshi Caldeira Takahashi – UFMG

# Sumário

---

1	Aplicação do algoritmo de colônia de formigas com informações dos multiplicadores de Lagrange na programação de sistemas termoeletrônicos de geração . . . . .	1
	<i>Ivo C. Silva Junior, Flávia R. Nascimento, Edimar J. Oliveira, André L.M. Marcato e Bruno H. Dias</i>	
2	Ferramenta computacional baseada em algoritmos genéticos aplicada ao planejamento da operação de sistemas hidrotérmicos . . . . .	19
	<i>Patricia T.L. Asano, Thiago R. de Alencar, Adriano A.F.M. Carneiro, André Carlos P.L.F. de Carvalho e Sadao Massago</i>	
3	Algoritmos genéticos aplicados ao planejamento da distribuição de energia elétrica em Curitiba e região metropolitana . . . . .	55
	<i>Arleide C. Alves, Maria T.A. Steiner e Nelson H. Mussi</i>	
4	Otimização multiobjetivo da operação de uma usina hidroelétrica . . . . .	73
	<i>Luis A. Scola, Oriane M. Neto, Ricardo H.C. Takahashi e Sérgio A.A.G. Cerqueira</i>	
5	Aplicação <i>on-line</i> de algoritmo genético na otimização de um regulador de velocidade <i>fuzzy</i> para acionamentos elétricos . . . . .	89
	<i>Wander G. da Silva, Bernardo Alvarenga, Paul P. Acarnley e John W. Finch</i>	
6	Comparação de técnicas de computação evolucionária para o projeto de filtros digitais de resposta finita ao impulso . . . . .	107
	<i>Daniel R. de Oliveira, Marlon H. Scalabrín, Fábio R. Teodoro, Rafael S. Parpinelli e Heitor S. Lopes</i>	
7	Caracterização e otimização de câmaras reverberantes para testes de compatibilidade eletromagnética . . . . .	129
	<i>Sérgio L. Ávila, Djonny Weinzierl, Mário A. Santos Jr., Luiz Lebensztajn, Carlos A.F. Sartori, José R. Cardoso e Laurent Krähenbühl</i>	

8	Computação evolucionária aplicada ao problema da recarga de reatores nucleares . . . . .	147
	<i>Roberto Schirru, Alan M.M. Lima, Andressa S. Nicolau, Ioná M.S. de Oliveira e Márcio H. da Silva</i>	
9	Resolução de problemas inversos em processos difusivos e transferência radiativa usando o algoritmo de evolução diferencial . .	173
	<i>Fran S. Lobato, Valder Steffen Jr. e Antônio J. Silva Neto</i>	
10	Algoritmo genético com interação social na resolução de problemas de otimização global com restrições . . . . .	197
	<i>Otávio N. Teixeira, Walter A.L. Lobato, Hitoshi S. Yanagibashi, Rodrigo V. Cabvalcante, Deam J.A. da Silva e Roberto C.L. de Oliveira</i>	
11	Controlador preditivo neural do nível do molde do lingotamento contínuo sintonizado via algoritmo genético . . . . .	225
	<i>Fábio B. Sanchotene, Gustavo M. de Almeida e José L.F. Salles</i>	
12	Algoritmo evolutivo híbrido para escalonamento integrado na agroindústria . . . . .	251
	<i>Ademir A. Constantino, Dario Landa-Silva e Wesley Romão</i>	
13	Heurísticas evolutivas híbridas para o problema de escalonamento de projetos com restrição de recursos dinâmicos . . . . .	273
	<i>André R.V. da Silva e Luiz S. Ochi</i>	
14	Algoritmos evolucionários na solução do problema do caixeiro alugador . . . . .	301
	<i>Marco C. Goldberg, Paulo H.A. da Silva e Elizabeth F.G. Goldberg</i>	
15	Algoritmo micro-genético aplicado ao <i>scheduling</i> de uma rede de distribuição de derivados de petróleo . . . . .	331
	<i>Henrique Westphal, Flávio Neves Jr. e Lúcia V.R. de Arruda</i>	
16	Aplicação de algoritmos bio-inspirados para a síntese e otimização de sistemas para produção de petróleo <i>offshore</i> . . . . .	355
	<i>Beatriz S.L. Pires de Lima e Breno P. Jacob</i>	

# Aplicação do Algoritmo de Colônia de Formigas com Informações dos Multiplicadores de Lagrange na Programação de Sistemas Termoeletrônicos de Geração

Ivo C. Silva Junior\*, Flávia R. Nascimento, Edimar J. Oliveira,  
André L. M. Marcato e Bruno H. Dias

---

**Resumo:** Este capítulo propõe a determinação e a utilização de uma ordem de mérito para as unidades termoeletrônicas de geração como informação adicional no processo de busca bio-inspirado conhecido como colônia de formigas. Esta ordem de mérito é baseada nas informações fornecidas pelos multiplicadores de Lagrange associados às variáveis discretas de decisão, as quais são representadas através de uma função contínua. Com a ordem de mérito determinada, um percentual de indivíduos da colônia faz uso destas informações no processo de busca visando a minimização da programação diária de operação. Os resultados alcançados através das simulações indicam que as informações dos multiplicadores de Lagrange aumentaram a eficiência da colônia em seu processo de busca.

**Palavras-chave:** Multiplicadores de Lagrange, Operação de unidades termoeletrônicas, Despacho Térmico, Otimização por colônia de formigas.

**Abstract:** *This chapter proposes the use of a merit order as additional information in the ant colony optimization process applied to thermoelectric generation systems. This merit order is based on information provided by the Lagrange multipliers related to the discrete decision variables. Those variables are represented by a continuous function. After determining the merit order, a set of the ant colony individuals make use of that information aiming at minimizing the daily dispatch of the system. Results indicate that information obtained from Lagrange multipliers increase the efficiency of the ant colony search process.*

**Keywords:** *Lagrange coefficients, Thermal unit commitment, Thermal dispatch, Ant colony optimization.*

---

\* Autor para contato: ivo.junior@ufjf.edu.br

## 1. Introdução

A programação diária da operação da geração é um dos problemas mais relevantes do setor de energia elétrica. Nessa tarefa, a meta é o atendimento da demanda ao menor custo possível através da otimização dos recursos disponíveis (Fortunato et al., 1990). A programação da operação torna-se cada vez mais relevante, não apenas para promover economia e segurança do sistema, mas também pelas seguintes razões:

- As decisões de partida, parada e as dinâmicas quanto à reativação das modernas instalações geradoras são complexas e dispendiosas;
- No atual modelo competitivo do setor elétrico, ganhos de pequena porcentagem tornaram-se economicamente relevantes;
- Há uma variação acentuada entre as demandas em horários de pico e fora do pico.

A programação diária da operação pode ser definida como sendo a determinação de uma estratégia de operação, cujo objetivo é indicar dentre todas as unidades geradoras existentes no sistema quais devem ser colocadas em operação e suas respectivas potências horárias de saída, de modo a atender a demanda de energia, satisfazendo as restrições operacionais e funcionais do sistema elétrico (Wood & Wollenberg, 1996; Padhy, 2004).

O problema pode ser dividido em dois subproblemas: (i) referente à determinação das unidades que devem estar em operação ou não mediante a demanda solicitada; (ii) referente à determinação da potência gerada por cada uma das unidades colocadas em operação pelo subproblema (i), o despacho econômico. A diferença entre ambos os problemas é a questão relacionada à variação da demanda ao longo do tempo. O problema do despacho visa determinar unicamente a potência elétrica de saída de cada unidade termoeleétrica, portanto, não decide sobre quais unidades devem estar em operação ou não e quando isso deve ocorrer. A variação da curva de carga ao longo do tempo requer o acionamento e/ou desligamento de determinadas unidades geradoras, o que envolve novos custos e restrições aumentando a complexidade do problema. Quanto às escalas de tempo envolvidas, a programação diária da operação cobre o escopo das decisões de operação do sistema elétrico de potência a cada hora, dentro do horizonte de um dia a duas semanas.

Classificado matematicamente como um problema de programação não linear inteira mista, o problema apresenta as seguintes dificuldades: (i) Região de solução não convexa, o que permite a existência de várias soluções e conduz grande parte dos algoritmos a convergirem em direção de mínimos locais; (ii) Natureza combinatória do processo de decisão, que leva ao fenômeno da explosão combinatória referente às alternativas de operação, acarretando elevado tempo computacional. A Tabela 1 apresenta o número total de combinações de operação em relação a um determinado número

de Unidades Termoeletricas de Geracao (UTES) para um periodo de vinte e quatro horas de operacao; (iii) Natureza dinamica do processo de decisao, que se por um lado limita as opcoes de decisao, por outro ocasiona antagonismo em relacao ao despacho economico.

Tabela 1. Natureza combinatoria do problema.

Número de UTES	Número de Combinações
7	$3,0995 \times 10^{50}$
10	$1,7259 \times 10^{72}$
40	$9,7453 \times 10^{288}$

Fazendo uma análise da literatura pode-se constatar que há necessidade tanto de aperfeiçoamentos dos algoritmos existentes como da elaboração de novas técnicas para a resolução do problema em questão. O que se observa, no geral, é que por um lado, um algoritmo pode ser simples, rápido, mas que apresenta mínimos locais de baixa qualidade (alto custo operacional). Por outro lado, têm-se algoritmos complexos, lentos, mas que apresentam soluções sub-ótimas ou ótimas (baixo custo operacional).

Diante das dificuldades e dos cenários apresentados, surge a motivação do estudo, utilização e aperfeiçoamento das técnicas de otimização bio-inspiradas (Belede et al., 2009). Assim, o objetivo deste capítulo é a determinação ótima da programação diária de operação de unidades termoeletricas de geracao via algoritmo de colônia de formigas considerando as informações dos multiplicadores de Lagrange associados às variáveis discretas de decisão no processo de busca da colônia.

Os resultados para quatro sistemas termoeletricos de geracao amplamente difundidos na literatura serão apresentados de modo a verificar a eficiência das informações dos multiplicadores de Lagrange no critério de busca do algoritmo de colônia de formigas.

## 2. Colônia de Formigas

As formigas, em sua busca por comida, deixam trilhas bem definidas entre a colônia e a fonte de alimento, isso acontece devido ao depósito de uma substância chamada feromônio. Embora o processo de busca se inicie de forma aleatória, se uma formiga encontra um caminho mais curto entre a fonte de comida e o ninho, esta, ao retornar para o ninho, reforçará a trilha de feromônio deixada por ela na ida. Como seu caminho é mais curto, ela percorrerá o trajeto de ida e volta mais vezes do que as outras, que levarão mais tempo para sair e voltar ao ninho, reforçando mais a concentração de feromônio. À medida que outras formigas encontrarem essa trilha com

maior quantidade de feromônio, estas continuarão depositando a substância, aumentando ainda mais a sua concentração, chegando a um ponto em que praticamente todas as formigas serão atraídas a ir somente por aquele caminho ótimo. O algoritmo de Colônia de Formigas (Dorigo et al., 1996; Bonabeau et al., 1999; Dorigo & Caro, 1999; Dorigo et al., 1999; Dorigo & Stützle, 2003, 2004) nada mais é do que um modelo matemático que representa este comportamento com o objetivo de encontrar uma solução para um determinado problema, seja este de otimização ou não.

### 3. Formulação do Problema da Programação Diária da Operação de Sistemas Termoelétricos de Geração

De maneira geral, existem dois tipos de restrições na formulação (Hobbs et al., 2001): (i) restrições sistêmicas, tais como atendimento à demanda, reserva girante e limites de transmissão. Estes tipos de restrições impõem alguma dificuldade ao problema, visto que acoplam as diversas termoelétricas existentes no sistema; (ii) restrições locais, tais como limites operacionais, tomadas e retomadas de carga e tempos mínimos de parada e partida. Estas restrições interferem apenas na operação das unidades termoelétricas de geração individualmente, ou seja, são restrições inerentes a cada unidade geradora.

A seguir será apresentada a notação utilizada e a formulação do problema referente à programação da operação de sistemas termoelétricos de geração.

Notação utilizada:

$N$	Número total de unidades térmicas;
$T$	Período total de operação;
$i$	Índice da unidade térmica;
$t$	Índice da hora;
$DO_i(t)$	Variável discreta $[0, 1]$ de decisão “ON/OFF”, da unidade térmica $i$ na hora $t$ ;
$P_i(t)$	Potência ativa gerada pela unidade térmica $i$ na hora $t$ ;
$P_i^{max}$	Limite máximo de potência ativa gerada pela unidade térmica $i$ ;
$P_i^{min}$	Limite mínimo de potência ativa gerada pela unidade térmica $i$ ;
$D(t)$	Demanda solicitada na hora $t$ ;
$r(t)$	Reserva girante solicitada na hora $t$ ;
$T_i^{on}$	Tempo mínimo de partida da unidade térmica $i$ ;
$T_i^{off}$	Tempo mínimo de parada da unidade térmica $i$ ;

$X_i^{on}(t)$	Tempo em que a unidade térmica $i$ está ligada;
$X_i^{off}(t)$	Tempo em que a unidade térmica $i$ está desligada;
$SC_i(t)$	Custo de partida da unidade térmica $i$ na hora $t$ ;
$SD_i(t)$	Custo de parada da unidade térmica $i$ na hora $t$ ;
$a_i, b_i, c_i$	Coefficientes referentes aos custos do combustível da unidade térmica $i$ ( $\$/h, \$/MWh, \$/MW^2h$ );
$hc_i$	Custo de partida quente da unidade térmica $i$ ;
$cc_i$	Custo de partida fria da unidade térmica $i$ ;
$csh_i$	Tempo de partida fria da unidade térmica $i$ ;
$\pi_p(t)$	Multiplicador de Lagrange associado ao limites de potência ativa no instante $t$ ;
$\lambda_p(t)$	Multiplicador de Lagrange associado à equação de balanço de potência no instante $t$ ;
$\lambda_r(t)$	Multiplicador de Lagrange associado à restrição de reserva girante no instante $t$ ;

A função objetivo (FOB), Equação 1, consiste na minimização da soma do custo total de operação ( $A$ , Equação 2), dos custos de partida ( $B$ , Equação 3) e de parada ( $C$ , Equação 4) das unidades geradoras durante o período de operação em estudo.

$$\min FOB = A + B + C \quad (1)$$

onde:

$$A = \sum_{t=1}^T \sum_{i=1}^N [a_i + b_i \cdot P_i(t) + c_i \cdot P_i^2(t)] \cdot DO_i(t) \quad (2)$$

$$B = \sum_{t=1}^T \sum_{i=1}^N SC_i(t) \quad (3)$$

$$C = \sum_{t=1}^T \sum_{i=1}^N SD_i(t) \quad (4)$$

É importante mencionar que o custo de partida das unidades geradoras depende do tempo que a unidade esteve parada anteriormente e do fato de se manter ou não as caldeiras quentes durante o período de parada. Desta forma, o custo de partida pode ser definido pela Equação 5.

$$SC = \left\{ \begin{array}{l} hc_i : T_i^{off} \leq X_i^{off}(t) \leq T_i^{off} + csh_i \\ cc_i : X_i^{off} > T_i^{off} + csh_i \end{array} \right\} \quad (5)$$

O custo de parada (SD) é sempre dado como um valor constante para cada unidade geradora e foi considerado, com na literatura, nulo nas simulações aqui realizadas.

A restrição de balanço de potência ativa, Equação 6, analisa, diretamente, o estado de equilíbrio do sistema elétrico a todo instante de tempo.

$$\sum_{i=1}^N DO_i(t) \cdot P_i(t) = D(t) \dots \lambda_d(t) \quad (6)$$

É necessário prever uma folga, designada de reserva girante, entre a carga prevista e a potência total disponível entre as unidades geradoras em serviço para suprir aumentos inesperados de carga ou desvios de previsão. A Equação 7 traduz esta folga.

$$\sum_{i=1}^N DO_i(t) \cdot P_i^{max} \geq D(t) + r(t) \dots \lambda_r(t) \quad (7)$$

Em relação às unidades geradoras foram consideradas as seguintes restrições: (i) tempos mínimos de partida e parada, Equações 8 e 9 respectivamente; (ii) limites máximos e mínimos de produção, Equação 10. Destaca-se que os limites inferiores são sempre positivos e não nulos.

$$X_i^{on}(t) \geq T_i^{on}(t) \quad (8)$$

$$X_i^{off}(t) \geq T_i^{off}(t) \quad (9)$$

$$P_i^{min} \leq P_i(t) \leq P_i^{max} \dots \pi_{pi}(t) \quad (10)$$

#### 4. Metodologia Proposta

A metodologia proposta baseia-se na utilização dos multiplicadores de Lagrange, obtidos da solução contínua do problema da programação diária da operação de sistemas termoeletricos de geração, como informações adicionais no processo de busca do algoritmo de colônia de Formigas. Para tanto, a metodologia proposta utiliza três etapas distintas: (a) Obtenção da solução relaxada do problema, com objetivo de determinar o limite inferior do custo de operação e assim, dar uma sensibilidade ao processo de busca no que diz respeito à distância das soluções obtidas pela colônia em relação ao limite inferior encontrado; (b) Obtenção dos multiplicadores de Lagrange associados aos estados operativos de cada uma das unidades geradoras, durante o período de estudo, com o objetivo de elaborar de uma ordem de mérito que auxiliará no processo de busca via colônia de formigas; (c) As informações oriundas dos multiplicadores de Lagrange são inseridas na colônia de modo auxiliar no processo de busca.

#### 4.1 Limite inferior do custo de operacao

Nesta etapa, o problema e relaxado com o objetivo de se determinar o limite inferior do custo operacional do sistema termoeletrico de geracao. Esta solucao do problema e obtida: (i) negligenciando as restricoes de parada e partida das unidades geradoras; (ii) permitindo que a variavel discreta de decisao assuma valores contnuos entre o intervalo  $[0, 1]$ . Desta forma, resolve-se o problema de otimizacao e obtm-se o custo total de operacao, sendo este, o menor custo possivel para o problema. Ou seja, a solucao discreta, procurada, nunca apresentara um custo menor do que o obtido atraves das consideracoes realizadas, solucao contnuua (Goldbarg & Luna, 2005). O objetivo e dar uma sensibilidade ao processo de busca da colnia, no que diz respeito a distancia entre solucoes obtidas pela colnia em relacao ao Limite Inferior de Custo .

#### 4.2 Multiplicadores de Lagrange

Enquanto problemas contnuos encontram uma serie de algoritmos robustos e eficientes, problemas discretos nao compartilham de tais algoritmos (Puccini & Pizzolato, 1990). Em virtude dessa realidade, surgiram varias tecnicas computacionalmente eficientes, mas que nao garantem a otimalidade de problemas discretos (Goldbarg & Luna, 2005). De modo a evitar as dificuldades peculiares da resolucao de problemas de programacao discreta, a metodologia proposta permite que a variavel discreta de decisao ( $DO$ ), seja representada por uma funcao denominada de Funcao Decisao de Operacao ( $FDO$ ) e assuma valores contnuos dentro do intervalo discreto de decisao  $[0, 1]$ . Assim, o problema que originalmente e de programacao inteira passa a ser formulado como um problema de programacao contnuua.

Teoricamente qualquer funcao contnuua poderia ser utilizada na modelagem da  $FDO$ . Entretanto, optou-se por uma funcao contnuua da familia das funcoes sigmoides. As funcoes sigmoides sao adequadas no contexto do problema, ja que possuem dinamica similar ao da funcao degrau unitario na representacao dos dois estados de decisao "ON-OFF" inerentes ao problema (Figura 1).

As funcoes sigmoides sao comumente empregadas como possiveis funcoes de ativacao na modelagem de neuronios artificiais em trabalhos envolvendo redes neurais (Haykin, 2007).

As Equacoes 11 e 12 sao referentes a funcao sigmoide adotada e a canalizacao do argumento da funcao, respectivamente.

$$FDO_i(x_i^t) = \frac{e^{\alpha \cdot x_i^t} - 1}{e^{\alpha \cdot x_i^t} + 1} \quad (11)$$

$$x_i^{min} \leq x_i^t \leq x_i^{max} \dots \pi_{x_i}(t) \quad (12)$$

onde:

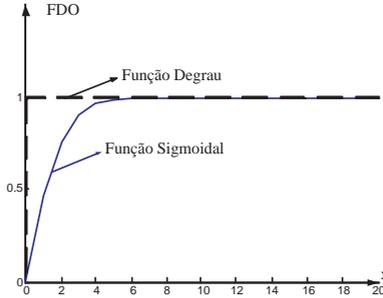


Figura 1. Função decisão de operação.

$FDO_i(x_i^t)$	Função contínua sigmóide que representa a decisão de operação “ON/OFF” da unidade térmica $i$ na hora $t$ ;
$x_i^t$	Argumento da função sigmóide da unidade térmica $i$ na hora $t$ ;
$x^{min} . x^{max}$	Limite mínimo e máximo do argumento da função sigmóide;
$\pi_{x_i}(t)$	Multiplicador de Lagrange associado ao argumento da função sigmóide da unidade $i$ na hora $t$ .

Assim como em [de Oliveira et al. \(2005\)](#), vários valores para o parâmetro foram testados e pode-se verificar que o valor unitário representa um bom compromisso entre a convergência e a sensibilidade desejada.

Com a função decisão de operação ( $FDO$ , Equação 11), assume valores contínuos, verificou-se que a condição inicial, do argumento da função sigmóide, consiste em um fator complicador, visto que esta interfere diretamente na solução devido à não convexidade da região de solução. Assim, com o objetivo de evitar esta situação indesejável, ao invés de se utilizar diretamente o valor da  $FDO$  como critério para a elaboração da ordem de mérito, optou-se por investigar os multiplicadores de Lagrange associados às funções de decisão. Para tanto, os seguintes aspectos são considerados:

1. Intervalo rígido de canalização para o argumento da  $FDO$ ,  $0 \leq x_i^t \leq 0,0001$ , de todas as unidades geradoras existentes no sistema elétrico. Isto porque se pretende avaliar a sensibilidade da função objetivo em relação à tendência de acionamento de cada unidade termoeletrica diante da demanda solicitada, ou seja,  $FDO(x_i^t) \rightarrow 1$ .
2. A inserção de uma nova unidade geradora, fictícia, denominada unidade de convergência ( $Pg_C$ ). Esta nova unidade tem como objetivo possibilitar a convergência do problema de otimização devido ao estreitamento da restrição de canalização do argumento da  $FDO$ . Além disso, esta unidade tem como característica, alto custo operacional

quando comparado com as demais unidades existentes e capacidade de atender sozinha a demanda e reserva do sistema.

Diante da inserção desta nova unidade, uma parcela adicional ( $D$ ), Equação 13, é incluída na *FOB* tradicional, Equação 1, do problema com objetivo de possibilitar a convergência e a obtenção dos multiplicadores de Lagrange associados às variáveis discretas de decisão  $[0 - 1]$  das unidades geradoras.

$$D = \sum_{t=1}^T \mu \cdot Pg_C(t) \quad (13)$$

onde:

- $\mu$  Custo operacional da unidade termoeletrica de convergência (R\$/MWh);
- $Pg_C(t)$  Potência ativa (MW) gerada pela termoeletrica de convergência na hora  $t$ .

Diante das considerações anteriores apresentadas, o processo de otimização, inicialmente, apresentará a tendência em colocar as unidades geradoras mais econômicas em serviço, porém com a imposição  $0 \leq x_i^t \leq 0,0001$ , as restrições de balanço de potência e reserva girante, Equações 6 e 7, só poderão ser atendidas através da unidade de convergência. Portanto, cabe a esta unidade suprir sozinha a demanda e a reserva girante solicitada pelo sistema, apesar do alto custo associado à mesma. Desta forma, tem-se como solução do problema de otimização os valores para os multiplicadores de Lagrange associados aos argumentos da função decisão de operação de cada uma das unidades geradoras existentes.

Com os multiplicadores conhecidos é possível obter uma Matriz de Sensibilidade ( $MS$ ), Equação 14. Esta matriz é formada pelos multiplicadores de Lagrange associados às *FDO* e traduz a sensibilidade da função objetivo em relação à tendência de acionamento de cada uma das termoeletricas em relação à demanda horária solicitada ao longo do período de operação.

$$MS = \begin{bmatrix} \pi_{x_1}(t) & \pi_{x_2}(t) & \dots & \pi_{x_N}(t) \\ \pi_{x_1}(t+1) & \pi_{x_2}(t+1) & \dots & \pi_{x_N}(t+1) \\ \vdots & \vdots & \dots & \vdots \\ \pi_{x_1}(T) & \pi_{x_2}(T) & \dots & \pi_{x_N}(T) \end{bmatrix} \quad (14)$$

Como estes multiplicadores são de caráter local, optou-se por calcular a média destes coeficientes de Lagrange para cada uma das unidades, Equação 15. O objetivo é obter uma sensibilidade geral e com isso uma ordem de mérito única para todo o período de operação.

$$\bar{\pi}_{x_i} = \frac{\sum_{t=1}^T \pi_{x_i}(t)}{T}, i = 1, \dots, N \quad (15)$$

Como os valores dos multiplicadores são negativos, já que estes retratam a redução da função objetivo em relação à tendência da colocação das unidades em serviço, a ordem de mérito é obtida através da ordenação crescente dos valores médios dos multiplicadores de Lagrange.

### 4.3 Processo de busca via colônia de formigas

O processo de busca é inicializado com uma população de indivíduos (formigas) que representam informações referentes à solução do problema. Esse conjunto de indivíduos, que será um conjunto de soluções, é denominado de colônia.

Cada solução é representada por uma matriz cuja dimensão é dada pelo número de horas de operação (linhas da matriz) por número de unidades termoeletricas existentes (colunas da matriz). Os elementos desta matriz representam as decisões horárias de operação (0 – 1) de todas as unidades térmicas de geração para todo o período de operação.

O processo de busca tem início de forma completamente aleatória, sendo as soluções iniciais escolhidas ao acaso. Entretanto, de modo à garantir a viabilidade, o processo de construção das soluções é feito de hora em hora, de modo que a soma horária das capacidades máximas de geração das unidades termoeletricas colocadas em operação seja maior do que a demanda horária a ser atendida e a reserva horária prevista. Depois de gerada a solução discreta do problema, deve-se verificar se há ou não violações dos tempos de parada e partida das unidades. Estas violações, caso existam, são tratadas através de procedimentos heurísticos descritos em (da Silva Jr., 2008). Assim, garante-se a viabilidade das soluções geradas.

O passo seguinte é avaliar as soluções iniciais obtidas. Essa avaliação é feita com base no valor numérico da função objetivo associada a cada indivíduo existente na colônia. Se, por exemplo, a função objetivo for de minimização, quanto menor o valor desta, maior deverá ser a quantidade de feromônio associada a esta solução. Ou seja, de maneira geral, as melhores soluções terão maior valor de feromônio associado. Desta forma, calcula-se o valor da Função Objetivo (*FOB*) do problema de otimização em estudo, Equação 1. Esta informação é importante na determinação da intensidade de feromônio a ser depositada por cada indivíduo da colônia.

Através do valor da *FOB* pode-se formar uma matriz de orientação para a colônia, denominada de Matriz de Feromônio (*MF*). Esta matriz possui a mesma estrutura utilizada na representação de cada indivíduo (solução) da colônia. É importante mencionar que a *MF* é única para todo o processo de busca, sendo atualizada a cada iteração da seguinte forma: (a) soluções cujos custos operacionais forem próximos do limite inferior, contribuem de forma mais contundente na matriz de feromônio;

(b) solucoes cujos custos operacionais forem distantes do limite inferior, contribuem de forma superficial na matriz de feromônio. Desta forma, o feromônio depositado, na  $MF$ , pelo indivíduo  $k$  para a solucao  $m$  na iteracao  $i$  é dado pela Equacao 16.

$$\Delta MF_k^i(m) = \frac{1}{|FOB_k^i(m) - LIC|} \quad (16)$$

onde:

- $FOB_k^i(m)$  Valor da funcao objetivo referente à solucao  $m$ , obtida pela formiga  $k$ , na iteracao  $i$ ;
- $LIC$  Valor do limite inferior de custo para o sistema elétrico em análise;
- $\Delta MF_k^i(m)$  Depósito de feromônio da formiga  $k$ , referente a solucao  $m$ , na iteracao  $i$ .

Esse depósito é importante a fim de guardar as informacoes das solucoes encontradas por todos os individuos da colonia. É a chamada inteligencia coletiva. Sendo assim, o depósito é acumulativo, tornando a quantidade de feromônio mais acentuada nas melhores solucoes encontradas pela colonia.

A quantidade de feromônio depositada por cada indivíduo, na  $MF$ , é atualizada pela adicao do depósito de feromônio de todas as formigas na iteracao atual, conforme a Equacao 17.

$$MF^i(m) = MF^{i-1}(m) + \Delta MF^i(m) \quad (17)$$

onde:

- $MF^i(m)$  Quantidade total de feromônio depositado, na  $MF$ , para a solucao  $m$  na iteracao  $i$ ;
- $MF^{i-1}(m)$  Quantidade total de feromônio depositado, na  $MF$ , para a solucao  $m$  na iteracao  $i-1$ ;
- $\Delta MF^i(m)$  Depósito de feromônio, na  $MF$ , de todas as formigas para a solucao  $m$ , na iteracao  $i$ , conforme a Equacao 18, onde  $nf$  é o número de formigas existentes na colonia:

$$\Delta MF^i(m) = \sum_{k=1}^{nf} \Delta MF_k^i(m) \quad (18)$$

Outro aspecto que deve ser levado em conta a fim de tornar o algoritmo mais eficiente é o fenômeno da evaporacao do feromônio. Sem uma taxa de decaimento da quantidade de feromônio, o algoritmo não terá um bom desempenho. A evaporacao permite que solucoes ruins sejam “esquecidas”,

além de permitir que a busca não fique presa em ótimos locais e possa, assim, varrer mais amplamente o espaço de soluções (Bonabeau et al., 1997). Desta forma, introduzindo um coeficiente de evaporação  $\rho$ , que pode variar entre zero e o valor unitário, a Equação 17 pode ser reescrita como:

$$MF^i(m) = (1 - \rho) \cdot MF^{i-1}(m) + \Delta MF^i(m) \quad (19)$$

O algoritmo aqui proposto utiliza uma taxa de evaporação de 10%.

Com as quantidades de feromônio determinadas, os indivíduos escolhem suas próximas decisões, normalmente, baseadas na concentração deste hormônio. Ou seja, o sorteio das novas soluções será ponderado pela quantidade de feromônio correspondente a cada solução. Isso é feito por meio de uma regra de transição, que fornece a probabilidade de cada formiga  $k$  escolher a solução  $m$ , conforme a Equação 20.

$$P_k(m) = \frac{[MF(m)]^\varphi \cdot [\eta(m)]^\beta}{\sum([MF]^\varphi \cdot [\eta]^\beta)} \quad (20)$$

onde:

- $P_k(m)$  Probabilidade de que a formiga  $k$  escolha a solução  $m$ ;
- $MF(m)$  Representa a quantidade de feromônio da solução  $m$ ;
- $\eta(m)$  É uma informação prévia (heurística) do problema, um índice de atratividade de escolha pela solução  $m$ . Esta informação pode existir ou não, depende do problema em estudo;
- $\varphi$  e  $\beta$  Parâmetros de controle que determinam o peso relativo da influência da concentração de feromônio ou da informação heurística do problema.

Entretanto, a colônia aqui proposta é composta por três tipos de formigas: (a) Soldados - Estas percorrem a região de solução de forma completamente aleatória e correspondem a 10% da composição da colônia; (b) Obreiras - Estas formigas percorrem a região de solução com base nas informações dos multiplicadores de Lagrange (ordem de mérito) e correspondem a 10% da composição da colônia; (c) Operárias - Estas formigas percorrem a região de solução com base nas informações dos feromônios depositados por todos os indivíduos da colônia e correspondem a 80% da colônia. Ressalta-se ainda, que não foi utilizado nenhum fator de atratividade, sendo este, um ponto futuro a ser explorado. Desta forma, utilizou-se dos seguintes valores: influência da concentração de feromônio,  $\varphi = 1$  e influência de informações heurísticas,  $\beta = 0$ .

Sorteada a nova colônia deve-se novamente avaliar as soluções obtidas. Antes, porém, deve-se verificar o critério de convergência do processo de

busca. Este passo consiste em verificar se o critério de parada do algoritmo foi atendido, como, por exemplo, o número máximo de iterações previamente determinado ou ainda o fenômeno da estagnação. Neste último, o processo é finalizado quando um percentual da colônia apresenta a mesma solução, ou seja, percorre a mesma trilha de feromônio. Enquanto um dos critérios não ocorre, o processo continua, retornando a etapa de avaliação das soluções geradas. No presente trabalho, adotou-se o número máximo de 50 iterações como critério de convergência do processo de busca.

## 5. Resultados

Com o objetivo de verificar a eficiência das informações dos multiplicadores de Lagrange no processo de busca pelo algoritmo de colônia de formigas foram realizadas simulações de um sistema com quatro unidades termoeletricas de geracao e um período de programação de oito horas (Valenzuela & Smith, 2002) e sistemas com dez, vinte e cem unidades geradoras, com um período de programação de vinte e quatro horas (Senjyu et al., 2003). O número total de combinações para os sistemas em análise é apresentado na Tabela 2, ilustrando a dificuldade da obtenção da melhor solução para o problema, ponto de mínimo global.

Tabela 2. Número total de combinações.

Número de UTEs	Número de Combinações
4	$(2^4)^8$
10	$(2^{10})^{24}$
20	$(2^{20})^{24}$
100	$(2^{100})^{24}$

A metodologia proposta foi implementada em MATLAB® e o ambiente computacional utilizado foi um computador Pentium Dual Core, 1.86 GHz e 2 GB RAM.

As Tabelas 3 e 4 apresentam, respectivamente, os custos operacionais e os tempos aproximados de processamento obtidos pela metodologia proposta.

Em negrito, destacam-se os resultados obtidos com a inserção dos multiplicadores de Lagrange, através das formigas Obreiras, no processo de busca da colônia. Os demais resultados não fazem uso das informações dos multiplicadores. Entretanto, são realizadas variações no tamanho da colônia de modo a verificar a eficiência do processo de busca sem estas informações adicionais.

É possível verificar que na colônia onde as informações baseadas nos multiplicadores de Lagrange foram inseridas, o processo de busca se apresentou mais eficiente. Ou seja, foi possível obter soluções de melhor quali-

Tabela 3. Resultados obtidos: custo total de operação.

Formigas	Sistemas em Estudo			
	4UTES	10UTES	20UTES	100UTES
<b>50</b>	<b>\$74.476</b>	<b>\$563.937</b>	<b>\$1.123,581</b>	<b>\$5.607.429</b>
50	\$74.476	\$576.228	\$1.166.425	\$5.951.426
100	\$74.476	\$574.817	\$1.162.181	\$5.916.357
450	\$74.476	\$573.759	\$1.149.894	\$5.915.437

Tabela 4. Tempo computacional (segundos).

Formigas	Sistemas em Estudo			
	4UTES	10UTES	20UTES	100UTES
<b>50</b>	<b>120</b>	<b>180</b>	<b>240</b>	<b>420</b>
50	120	180	240	420
100	240	300	360	780
450	960	1500	1800	3960

dade, menor custo operacional, com um número reduzido de indivíduos e, conseqüentemente, em um menor tempo de processamento.

Os resultados encontrados, pela metodologia proposta, são comparados com alguns resultados divulgados literatura especializada, os quais também utilizam técnicas de otimização bio-inspiradas, Tabela 5. São estas: (a) PSO - Otimização por Enxame de Partículas (Logenthiran & Srinivasan, 2010); (b) GA - Algoritmo Genético (Sun et al., 2006); (c) EP - Programação Evolucionária (Juste et al., 1999); (d) ACO - Otimização por Colônia de Formigas (Nualhong et al., 2004).

Fazendo a análise comparativa dos resultados pode-se verificar que a metodologia proposta apresentou resultados competitivos em relação a outras metodologias utilizadas na resolução do problema referente à programação de operação de unidades termoeletricas de geração. Apesar da diferença percentual entre os resultados serem relativamente pequenas, deve-se lembrar que os custos reais de operação são na ordem de milhões ou até bilhões de reais e que uma pequena redução representa uma cifra bastante significativa.

Considerando a natureza combinatória do problema, um aspecto relevante é a questão relacionada ao tempo de processamento. Entretanto, a falta de informações sobre os computadores (processadores e memórias) na execução das metodologias encontradas na literatura (Tabela 5), inviabilizam uma análise direta e correta entre os tempos computacionais. Entretanto, a Tabela 6 apresenta os tempos computacionais relatados na literatura e os obtidos pela metodologia proposta.

Tabela 5. Comparação de resultados.

Métodos	Custo Total de Operação		
	10UTES	20UTES	100UTES
EP	\$564.551	\$1.125.494	\$5.623.885
PSO	\$563.999	\$1.125.510	\$5.611.294
GA	\$564.467	\$1.126.199	\$5.616.957
ACO	\$563.977	\$1.123.989	\$5.608.630
<b>Proposto</b>	<b>\$563.937</b>	<b>\$1.123.581</b>	<b>\$5.607.429</b>

Tabela 6. Tempo Computacional.

Métodos	Tempo em segundos		
	10UTES	20UTES	100UTES
EP	100	340	6120
PSO	-	-	-
GA	221	733	15733
ACO	157	352	4965
<b>Proposto</b>	<b>180</b>	<b>240</b>	<b>420</b>

É importante ressaltar que nenhum processo de busca garante o ponto de mínimo global, ponto este que só é garantido através da enumeração completa das alternativas de operação, o que é inviável para sistemas de grande porte devido ao grande número de combinações possíveis de operação.

## 6. Conclusões

Este capítulo apresentou uma opção de representação para a variável discreta de decisão, inerente a programação diária da operação de sistemas termoeletricos de geracao e, conseqüentemente, a utilização das informações dos multiplicadores de Lagrange, associados as decisões de operação, com o objetivo de melhorar a eficiência do processo de busca baseado no comportamento de colônias de formigas. Diante dos resultados obtidos, os seguintes aspectos podem ser enfatizados:

- A metodologia proposta mostra-se competitiva em relação às principais técnicas de otimização bio-inspiradas, conciliando soluções de qualidade ao número reduzido de indivíduos;
- Apesar das pequenas diferenças econômicas entre as metodologias, deve-se ter em mente que para sistemas reais e/ou de grande porte, o montante financeiro envolvido é de ordem elevada e, portanto, quaisquer que sejam as economias obtidas são estas desejadas;

- A garantia de obtenção do ponto de mínimo global só pode ser obtida através da enumeração de todas as combinações possíveis de operação, o que nem sempre é possível devido ao elevado tempo de processamento;
- A utilização dos multiplicadores de Lagrange associados à variável de controle da função sigmóide mostrou-se um indicador eficiente das unidades termoeletricas relevantes ao problema;
- O aumento do número de indivíduos, na colônia, tende a melhorar a qualidade das soluções obtidas. No entanto, o tempo de simulação computacional aumenta consideravelmente, inviabilizando essa alternativa, principalmente, em sistemas de médio-grande porte e/ou com períodos grandes de programação;
- Uma opção de redução do tempo computacional seria a utilização de processamento paralelo (Roosta, 1999). O processamento paralelo permite que um algoritmo, que exija um elevado tempo computacional, tenha sua execução distribuída entre vários processadores e memórias, permitindo uma considerável redução no tempo de simulação.

## Referências

- Belede, L.; Jain, A. & Gaddam, R., Unit commitment with nature and biologically inspired computing. In: *Proceedings of World Congress on Nature and Biologically Inspired Computing*. Piscataway, EUA: IEEE Press, p. 824–829, 2009.
- Bonabeau, E.; Dorigo, M. & Theraulaz, G., *Swarm Intelligence: From Natural to Artificial Systems*. New York, EUA: Oxford University Press, 1999. 320 p.
- Bonabeau, E.; Theraulaz, G.; Deneubourg, J.; Aron, S. & Camazine, S., Self-organization in social insects. *Trends in Ecology and Evolution*, 12(5):188–193, 1997.
- da Silva Jr., I., Planejamento da Operação de Sistemas Termoeletricos Utilizando Análise de Sensibilidade Associada a Procedimentos Heurísticos. Tese de doutorado em engenharia elétrica, COPPE/UFRJ, Rio de Janeiro, RJ, 2008.
- de Oliveira, E.; da Silva Jr., I.; Pereira, J. & Carneiro Jr., S., Transmission system expansion planning using a sigmoid function to handle integer investment variables. *IEEE Transactions on Power Systems*, 20(3):1616–1621, 2005.
- Dorigo, M. & Caro, D., The ant colony optimization meta-heuristic. In: Corne, D.; Dorigo, M. & Glover, D. (Eds.), *New Ideas in Optimization*. New York, EUA, p. 11–32, 1999, 450 p.

- Dorigo, M.; Caro, D. & Gambardella, L., Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.
- Dorigo, M.; Maniezzo, V. & Coloni, A., The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 26(1):29–41, 1996.
- Dorigo, M. & Stützle, T., The ant colony optimization metaheuristic: Algorithms, applications, and advances. In: Glover, F. & Kochenberger, G. (Eds.), *Handbook of Metaheuristics*. New York, EUA: Springer, v. 57 de *Operations Research and Management Science*, p. 250–285, 2003.
- Dorigo, M. & Stützle, T., *Ant Colony Optimization*. Cambridge, EUA: MIT Press, 2004. 319 p.
- Fortunato, L.; Neto, T.; Albuquerque, J. & Pereira, M., *Introdução ao Planejamento da Expansão e Operação de Sistemas de Produção de Energia Elétrica*. Niterói, RJ: EDUFF, 1990.
- Goldberg, M. & Luna, H., *Otimização Combinatória e Programação Linear*. 2a edição. Rio de Janeiro, RJ: Editora Campus, 2005.
- Haykin, S., *Redes Neurais: princípios e práticas*. 2a edição. Rio de Janeiro, RJ: Editora Bookman, 2007. 900 p.
- Hobbs, B.; Rothkopf, M.; O'Neill, R. & Chao, H. (Eds.), *The Next Generation of Electric Power Unit Commitment Models*. 1a edição. New York, EUA: Kluwer Academic, 2001. 328 p.
- Juste, K.; Kita, H.; Tanaka, E. & Hasegawa, J., An evolutionary programming solution to the unit commitment problem. *IEEE Transactions on Power Systems*, 14(4):1452–1459, 1999.
- Logenthiran, T. & Srinivasan, D., Particle swarm optimization for unit commitment problem. In: *Proceedings of IEEE International Conference on Probabilistic Methods Applied to Power Systems*. p. 642 – 647, 2010.
- Nualhong, D.; Chusanapiputt, S.; Phoomvuthisarn, S.; Saengsuwan, T. & Jantarang, S., Diversity control approach to ant colony optimization for unit commitment problem. In: *Proceedings of IEEE Region 10 Emerging Technologies for Sustainable Development Conference*. v. 3, p. 488–491, 2004.
- Padhy, N., Unit commitment – a bibliographical survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205, 2004.
- Puccini, A. & Pizzolato, N., *Programação Linear*. 2a edição. Rio de Janeiro, RJ: Editora LTC, 1990. 250 p.
- Roosta, S., *Parallel Processing and Parallel Algorithms: Theory and Computation*. New York, EUA, 1999. 585 p.

- Senjyu, T.; Yamashiro, H.; Shimabukuro, K.; Uezato, K. & Funabashi, T., Fast solution technique for large scale unit commitment problem using genetic algorithm. *IEE Proceeding on Generation, Transmission and Distribution*, 150(6):753 – 760, 2003.
- Sun, L.; Zhang, Y. & Jiang, C., A matrix real-coded genetic algorithm to the unit commitment problem. *Electric Power System Research*, 76(9-10):716–728, 2006.
- Valenzuela, J. & Smith, A., A seeded memetic algorithm for large unit commitment problems. *Journal of Heuristics*, 8(2):173–195, 2002.
- Wood, A. & Wollenberg, B., *Power Generation, Operation and Control*. 2a edição. New York, EUA: J. Wiley & Sons, 1996. 592 p.

## Notas Biográficas

**Ivo Chaves da Silva Junior** é graduado e mestre em Eng. Elétrica pela UFJF (2001 e 2003), e Doutor em Eng. Elétrica pela UFRJ/COPPE (2008). Atualmente é professor adjunto do curso de Eng. Elétrica da UFJF e atua nos seguintes temas: otimização de sistemas elétricos de potência, expansão de sistemas de transmissão, operação de sistemas hidrotérmicos, técnicas heurísticas, metaheurísticas e mercado de energia.

**Flávia Rodrigues do Nascimento** é graduada (2008) em Eng. Elétrica pela UFJF. Atualmente é mestranda do Programa Pós-graduação em Eng. Elétrica da UFJF. Atua na área de operação de sistemas termoeletrônicos.

**Edimar José de Oliveira** é graduado em Eng. Elétrica pela UFJF (1985), Mestre pela UFU (1993) e Doutor em Eng. Elétrica pela Universidade Federal de Itajubá (1998). Desde 1989 trabalha no Depto. de Energia Elétrica da UFJF. Seu principal interesse está relacionado a pesquisas na área de métodos computacionais para análise de estabilidade, operação, expansão e controle de sistemas elétricos de potência.

**André Luís Marques Marcato** é graduado em Eng. Elétrica pela UFJF (1995), mestre e doutor em Eng. Elétrica pela PUC-Rio (1998 e 2002). É senior member do IEEE. Atualmente é professor Associado do Depto. de Energia da UFJF. Atua nas áreas de pesquisa operacional aplicada à coordenação hidrotérmica, planejamento da operação e expansão de sistemas elétricos e energéticos de grande porte.

**Bruno Henriques Dias** é graduado em Eng. Elétrica pela UFJF (2005), mestre e doutor em Eng. Elétrica pela PUC-Rio (2006 e 2010). Atualmente é Pós-doutorando do Programa Pós-graduação em Eng. Elétrica da UFJF. Atua na área planejamento da operação de sistemas elétricos e energéticos.

# Ferramenta Computacional Baseada em Algoritmos Genéticos Aplicada ao Planejamento da Operação de Sistemas Hidrotérmicos

Patricia Teixeira Leite Asano\*, Thiago Ribeiro de Alencar,  
Adriano Alber de França Mendes Carneiro,  
André Carlos Ponce de Leon Ferreira de Carvalho e Sadao Massago

---

**Resumo:** Este capítulo apresenta como a computação evolutiva pode ser utilizada na resolução do problema de Planejamento da Operação de Sistemas Hidrotérmicos. Com objetivo de atender as características do problema, foram feitas adaptações dos operadores genéticos tradicionais. Para validar a metodologia desenvolvida foi realizado um teste, adotando um sistema complexo de 35 usinas pertencentes ao Sistema Hidroelétrico Brasileiro. Os resultados experimentais mostraram o bom desempenho da metodologia proposta para determinar uma operação otimizada, garantindo, da melhor forma possível, o atendimento da demanda por um custo menor do que outras alternativas utilizadas em sistemas reais, mantendo a robustez.

**Palavras-chave:** Algoritmos genéticos, otimização, inteligência artificial, planejamento da operação de sistemas hidrotérmicos de potência.

**Abstract:** *This chapter shows how evolutionary computation can be used to solve the hydrothermal power systems planning problem. Aiming at attending the problem features, adaptations were done in the traditional genetic operators. To validate the developed methodology a test was performed using a complex system with 35 power plants from the Brazilian hydrothermal system. The experimental results showed the good performance of the proposed methodology to determine an optimized operation, leading to a solution with a lower cost than those of alternatives used in real systems, also keeping the robustness.*

**Keywords:** *Genetic algorithms, optimization, artificial intelligence, operation planning of hydrothermal power systems.*

---

\* Autor para contato: [patricia.leite@ufabc.edu.br](mailto:patricia.leite@ufabc.edu.br).

## 1. Introdução

Nos Sistemas Hidrotérmicos de Potência (SHP), a geração de energia elétrica pode ser obtida de forma hidroelétrica, quando resulta do aproveitamento da energia potencial da água, e térmica, através da queima de combustíveis diversos como carvão, óleo, gás ou fissão do urânio e, eventualmente, por importação de sistemas vizinhos, conforme apresentado na Figura 1. Toda a energia disponível é enviada através das linhas de transmissão para o atendimento da demanda. Portanto, faz-se necessário um Planejamento da Operação destes sistemas cada vez mais avançado, que englobe as fontes de geração de energia, com objetivo de garantir, da melhor forma possível, o atendimento da demanda. Este é um grande desafio para os gerenciadores do Setor Elétrico, caso não resolvido adequadamente, a uma crise de abastecimento de energia.

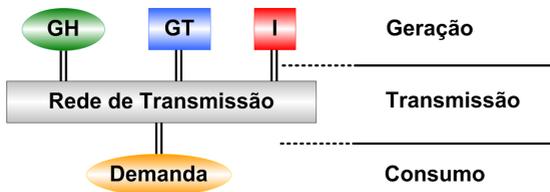


Figura 1. Sistema hidrotérmico, onde: GH é geração hidroelétrica, GT é geração térmica e I é importação de energia.

Além disso, em sistemas de energia elétrica que tem uma grande participação de geração hidroelétrica, como no caso do Brasil, aproximadamente 76,9% (EPE, 2010), pode-se utilizar a energia potencial da água armazenada nos reservatórios, gerenciando-a convenientemente, para atender a demanda e substituir de forma racional a geração dispendiosa das unidades térmicas.

Entretanto, o volume de água afluente aos reservatórios é desconhecido, pois depende basicamente das aflúncias que irão ocorrer no futuro. Além disso, a disponibilidade de energia hidroelétrica é limitada pela capacidade de armazenamento nos reservatórios. Isto introduz uma relação entre uma decisão de operação em uma determinada etapa e as consequências futuras desta decisão. Por exemplo, se a decisão for utilizar energia hidroelétrica para atender o mercado e no futuro ocorrer uma seca, poderá ser necessário utilizar geração térmica, de custo elevado, ou interromper o fornecimento de energia. Por outro lado, se a opção for o uso mais intensivo de geração térmica, conservando elevados os níveis dos reservatórios, e ocorrerem vazões altas no futuro, poderá haver vertimento no sistema, representando um desperdício de energia e, em consequência, um aumento desnecessário.

rio do custo de operação (Silva, 2001; MAE, 2002), conforme ilustrado na Figura 2

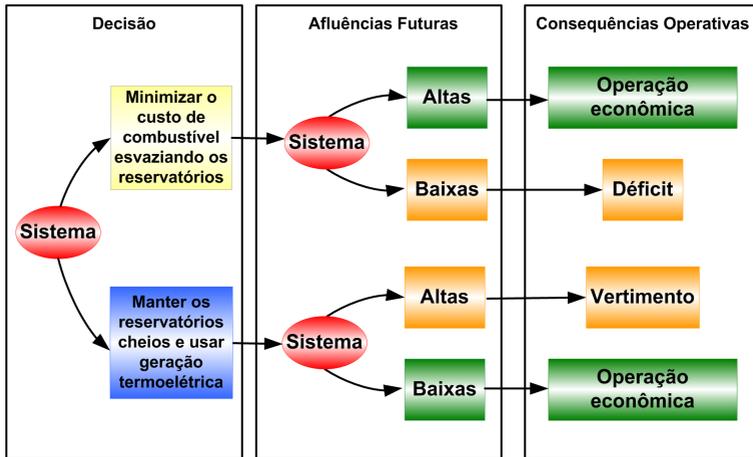


Figura 2. Cronograma de decisão para sistemas hidrotérmicos.

A decisão ótima deve equilibrar o compromisso entre o benefício presente do uso da água para a geração hidroelétrica e o benefício futuro esperado, advindo do seu armazenamento, tudo medido em termos de economia de combustível, caracterizando um problema dinâmico.

Devido às características do Sistema Brasileiro, o Planejamento da Operação de Sistemas Hidrotérmicos de Potência (POSHP) pode ser classificado como um problema de otimização dinâmico, de grande porte, com função objetivo não linear, não separável e não convexa, como mostrado por Soares & Carneiro (1991) e Leite (2003).

O objetivo do POSHP é determinar uma estratégia de geração para cada usina, que minimize o valor esperado dos custos operativos no período de planejamento e atenda a demanda dentro de um limite de confiabilidade. Desta forma, em sistemas com grande participação hidroelétrica, o objetivo econômico do planejamento da operação é substituir, na medida do possível, a geração de origem termoeletrica, de custo elevado, por geração de origem hidroelétrica, de custo praticamente nulo, de forma racional, como demonstrado por Pereira (1985), Carneiro et al. (1990) e Fortunato et al. (1985).

As técnicas clássicas de otimização para solução deste problema podem apresentar algumas dificuldades, principalmente devido à complexidade da função objetivo. Assim, a busca de melhorias nos métodos tradicionais, ou de abordagens alternativas, visa aperfeiçoar esta etapa vital no funci-

onamento dos Sistemas Hidrotérmicos de Potência (Macieira et al., 2003; Cicogna & Soares, 2003; Martinez, 2001; Cicogna, 2003).

Outras metodologias envolvendo técnicas de Inteligência Artificial (IA) vêm sendo investigadas e apresentam eficiência na solução de diversos problemas de otimização, dentre os quais pode-se destacar Chen & Chang (1996), Yang et al. (1996), Miranda et al. (1998), Chang & Chen (1998) e Gen & Cheng (1997).

Este trabalho trata da proposição de uma técnica de Inteligência Artificial no processo de otimização, baseada em Algoritmos Genéticos (AGs), voltada para o sistema Brasileiro, o qual possui características específicas que o diferencia dos demais sistemas no mundo, motivando o estudo e desenvolvimento de abordagens mais apropriadas.

A próxima seção apresenta a formulação matemática da operação ótima. Na Seção 3, alguns dos fundamentos da teoria de Algoritmos Genéticos são apresentados. Na Seção 4 é feita uma discussão do algoritmo proposto, bem como um breve comentário sobre as adaptações necessárias para utilizar os AGs na resolução do problema. A Seção 5 traz uma aplicação do método e a análise dos resultados obtidos. Finalmente, na Seção 6 são apresentadas as conclusões e observações finais do trabalho como um todo.

## 2. Formulação Matemática

O problema do Planejamento da Operação de Sistemas Hidrotérmicos de Potência (POSHP) pode ser formulado através de um modelo de otimização, onde o custo de operação dos sistemas hidrotérmicos é dado pelas termoeletricas, já que as usinas hidroelétricas têm custo de operação quase nulo, a menos de taxas, já que ela utiliza a energia potencial da água. Na realidade neste custo podem ser incluídas eventuais importações de energia e o custo do déficit, sem prejuízo das considerações e análises feitas.

A seguir apresenta-se uma análise sobre a modelagem do problema, bem como, as formas de obtenção do custo marginal de operação e do valor marginal da água, conforme definido em Carneiro (1991).

O custo de operação de uma usina termoeletrica  $C(G)$ , pode ser representado por uma função convexa crescente, próxima de um polinômio de segundo grau, dado pela Equação 1.

$$C(G) = a + b(G(t)) + c(G(t))^2 \quad (1)$$

onde:  $G$  é a potência gerada e  $a$ ,  $b$ ,  $c$  são parâmetros característicos da usina termoeletrica.

As restrições operativas são dadas pelos limites operacionais da usina, conforme a Equação 2:

$$\underline{G} \leq G(t) \leq \bar{G} \quad (2)$$

O conjunto de geração não hidráulico, que pode incluir a importação de energia e mesmo o corte de carga, pode ser modelado usando-se as funções de custo marginal para agregação em classes de térmicas. Desta forma, o custo de operação ótimo do conjunto não hidráulico é dado pela solução de um problema de despacho de geração, em um horizonte de tempo, que procura igualar os custos marginais entre os intervalos. Este custo é definido na seqüência de alocação da geração térmica e do déficit, pelo valor crescente dos respectivos custos marginais, das térmicas mais baratas para as mais caras (Carneiro, 1991; Martinez, 2001). Pode-se tomar, assim, uma função convexa crescente representando o custo de operação do conjunto das térmicas ou do sistema não hidráulico.

Para um subsistema composto por um conjunto de usinas térmicas sabe-se que o custo marginal em um intervalo é dado por  $\lambda$ , obtido pela agregação das disponibilidades térmicas. Portanto a função de custo marginal em um ponto é dada pela Equação (3).

$$\lambda = \left. \frac{\partial C[G(t)]}{\partial G(t)} \right|_{\tilde{G}(t)} \quad \forall t \quad (3)$$

Independentemente do combustível empregado nas usinas termoeletricas, o custo possui derivadas sempre positivas, conforme a Equação (4):

$$\frac{\partial C[G(t)]}{\partial G(t)} > 0 \quad (4)$$

O problema do Planejamento da Operação de Sistemas Hidrotérmicos de Potência, tem como custo de operação justamente o custo do sistema não hidráulico complementar, dado pela função  $C(\cdot)$ , o qual depende obviamente da demanda total  $D$  do sistema e também da geração hidroelétrica  $H$ , pois quanto maior esta geração, menor será a complementação térmica necessária, admitindo-se que  $D(t) = G(t) + H(t)$ , onde:

$$C = C[G(t)] \quad (5)$$

onde:

$$C(t) = C[D(t) - H(t)] \quad \text{para } D(t) \geq H(t), \quad (6)$$

$$C(t) = 0 \quad \text{para } D(t) < H(t). \quad (7)$$

Com este modelo adotado, o planejamento hidrotérmico passa a ser um problema tratado exclusivamente através das usinas hidroelétricas.

Neste capítulo, onde investiga-se Algoritmos Genéticos como técnica de resolução do POSHP, as suas variáveis são obtidas através de uma busca aleatória (Leite et al., 2002, 2006). Portanto, o cálculo do custo marginal de

operação pode ser feito da forma tradicional, derivando-se analiticamente a função objetivo, externamente aos AGs, e substituindo-se os valores ótimos encontrados pelo algoritmo proposto.

De  $D(t) = G(t) + H(t)$ , vem:

$$H[x(t), u(t)] = -G(t). \quad (8)$$

Logo:

$$\lambda(t) = - \frac{d CD(t) - H[x(t), u(t)]}{dH[x(t), u(t)]} \Big|_{\hat{H}(t)} = \frac{\partial BD(t) - H[x(t), u(t)]}{\partial H(t)} \Big|_{\hat{H}(t)}. \quad (9)$$

onde:  $B[G(t)]$  é o benefício resultante da utilização da água pela usina hidroelétrica, no intervalo  $t$ .

A Equação (9) expressa, ainda, o custo marginal de operação, agora como função da geração hidroelétrica, mostrando-o como o decréscimo de custo, ou o benefício, que ocorreria caso houvesse um aumento incremental da geração hidráulica (Carneiro, 1991).

Da mesma forma, o valor da água em cada usina pode ser facilmente encontrado, já que a abordagem usada representa individualmente as usinas. Assim basta ponderar o custo marginal de operação do sistema pela geração marginal da usina hidroelétrica em relação à turbinagem no ponto ótimo, onde:

$$\psi(t) = \lambda(t) \frac{\partial H[x(t), u(t)]}{\partial u(t)} \Big|_{\hat{u}(t)}. \quad (10)$$

Isto pode ser entendido como uma igualdade na operação ótima, entre o benefício marginal de armazenar a água e o benefício marginal de defluir. Maiores detalhes podem ser encontrados em Martinez (2001); Leite et al. (2002, 2006); Tavares et al. (1984).

Como exemplo, e sem prejuízo de generalidade, no caso de se usar uma função de custo do tipo  $C(G) = k[G(t)]^2$ , tem-se:

$$\frac{\partial C(G(t))}{\partial G(t)} = 2kG(t). \quad (11)$$

Em sistemas com uma porcentagem substancial de geração hidroelétrica, como no caso do Brasil, pode-se utilizar a energia armazenada nos reservatórios do sistema para atender a demanda, substituindo assim a geração dispendiosa das unidades térmicas. Neste caso o valor da água em si, ou seja valor dessa energia gerada, é determinado pela geração termoeleétrica, ou melhor, o valor da geração hidráulica em cada usina é dado pelo valor do que é substituído de energia termoeleétrica.

A Equação (12), que reafirma o conceito do valor marginal da água, pode ser expresso como o decréscimo que se obteria no custo de operação, devido a um aumento incremental no uso da água pela usina hidráulica, em cada instante de tempo, ou seja, o benefício advindo de um acréscimo marginal no uso do recurso hídrico. Daí o conceito de que este valor representa, em termos marginais, o que a água conseguiria reduzir de custo pela substituição da geração termoeletrica por hidroeletricidade, na condição de operação ótima (Carneiro, 1991).

$$\psi(t) = - \left. \frac{dCD(t) - H[x(t), u(t)]}{du(t)} \right|_{\hat{u}(t)} = \left. \frac{\partial BD(t) - H[x(t), u(t)]}{\partial u(t)} \right|_{\hat{u}(t)} \quad (12)$$

Portanto, a formulação geral do planejamento da operação energética de sistemas hidrotérmicos de potência não sofre nenhuma alteração. Esta pode ser estabelecida como a minimização do custo operacional do sistema, ao longo de todo o horizonte de planejamento  $[1, T]$ , dado por:

$$\text{Min} \sum_{t=1}^T [C(t)] \quad (13)$$

sujeito a:

$$H(t) = \sum_{i=1}^N \phi[x_i(t), q_i(t), z_i(t)] \quad (14)$$

$$q_i(\cdot) + z_i(\cdot) = u_i(\cdot) \quad (15)$$

$$\phi_i(x_i, q_i, z_i) = K_i[h_{1i}(x_i) - h_{2i}(q_i + z_i)]q_i \quad (16)$$

$$x_i(t+1) = x_i(t) + y_i(t) + \sum_{k \in \Omega_i} [q_k(t) + z_k(t)] - q_k(t) - z_k(t) \quad (17)$$

$$x_{i, \text{mim}}(t) \leq x_i(t) \leq x_{i, \text{máx}}(t) \quad (18)$$

$$q_i(t) \leq q_{i, \text{máx}}(t) \quad (19)$$

$$u_{i, \text{min}}(t) \leq q_i(t) + z_i(t) \quad (20)$$

onde:

- $N$  - número de usina hidroelétricas;
- $\phi_i(\cdot)$  - função de geração hidroelétrica da usina  $i$ ;

- $x_i(\cdot)$  - volume do reservatório  $i$  no intervalo  $t$ ;
- $q_i(\cdot)$  - vazão turbinada pela usina  $i$  no intervalo  $t$ ;
- $z_i(\cdot)$  - vazão vertida da usina  $i$  no instante  $t$ ;
- $u_i(\cdot)$  - vazão defluída da usina  $i$  no instante  $t$ ;
- $K_i$  - constante que engloba aceleração da gravidade, densidade da água, rendimento turbina-gerador e fatores de conversão de unidade;
- $h_{1i}$  - altura de montante, em relação ao nível do mar, função do volume do reservatório;
- $h_{2i}$  - altura de jusante, em relação ao nível do mar, função da vazão defluída total;
- $y_i(t)$  - afluência ao reservatório  $i$  no instante  $t$ ;
- $\Omega_i$  - conjunto de todas as usinas imediatamente a montante da usina  $i$ ;
- $u_{i,mim}$  - representa a defluência total mínima obrigatória.

Além disto, os volumes iniciais dos reservatórios são dados e os volumes finais podem ser fixados e/ou penalizados.

Como ferramenta de resolução do problema determinado pelas Equações de (12) a (20), tradicionalmente, adota-se um algoritmo de otimização não-linear (PNL), que pode ser baseado em técnica de fluxo em rede, onde preserva-se a representação individualizada das usinas (Carneiro et al., 1990; Carneiro, 1991; Carneiro et al., 1988; Carvalho, 1986; Cicogna & Soares, 2003; Rabêlo, 2010).

Entretanto, esta linha de abordagem apresenta imperfeições maiores ou menores, dependendo do sistema aplicado, que levam à procura de novos caminhos (Cicogna, 1999; Pereira et al., 1987; Tavares et al., 1984; Carneiro et al., 1988). Vários autores fizeram estudos comparativos entre as técnicas adotadas analisando as diversas formas de abordagens e mostrando as dificuldades de se abranger todos os aspectos da questão (Rosenthal, 1980; Fortunato et al., 1985; Pereira et al., 1987).

As técnicas de resolução de otimização tradicionais apresentam dificuldades inerentes aos próprios métodos, como a necessidade de cálculo de derivadas, a inversão de matrizes, problemas de convergência, além de outras dificuldades devido à complexidade da função objetivo, já relacionadas na introdução.

Uma nova abordagem baseada em Algoritmos Genéticos foi explorada, na tentativa de buscar técnicas que forneçam melhores resultados para otimização da operação, principalmente em casos de sistemas complexos e de difícil solução.

### 3. Algoritmos Genéticos

Os Algoritmos Genéticos (AGs) foram introduzidos em meados de 1976 por John Holland e seus colaboradores da Universidade de Michigan; mas seu pleno desenvolvimento só ocorreu a partir da década de 80 através do trabalho de Goldberg (Goldberg, 1989). Os AGs são algoritmos de busca inspirados na teoria da Seleção Natural. Eles atuam sobre uma população de indivíduos, baseados no fato de que indivíduos com boas características genéticas têm maiores chances de sobrevivência e de produzirem indivíduos cada vez mais aptos, enquanto indivíduos menos aptos tendem a desaparecer.

Algoritmos Genéticos baseiam-se inicialmente na geração de uma população formada por um conjunto aleatório de indivíduos, que podem ser vistos como possíveis soluções para um problema. Durante o processo evolutivo, esta população é avaliada: para cada indivíduo é dado um índice, refletindo sua habilidade de adaptação a um determinado ambiente. Uma porcentagem dos mais adaptados é mantida, enquanto os outros são descartados. Os membros selecionados podem sofrer modificações em suas características fundamentais através de mutações e cruzamentos genéticos, gerando descendentes para a próxima geração. Este processo, chamado de reprodução, é repetido até que um conjunto de soluções satisfatórias seja encontrado (Goldberg, 1989; Beasley et al., 1993).

### 4. Algoritmo Proposto

Para o desenvolvimento e aplicação do algoritmo proposto para a solução do problema do Planejamento da Operação de Sistemas Hidrotérmicos de Potência foi necessária uma preparação e adaptação dos AGs às condições encontradas no problema do POSHP (Leite, 2003).

Uma grande preocupação nesta preparação e adaptação foi a de respeitar as características físicas do problema original, tentando sempre observar a relação entre o real e o comportamento do algoritmo.

Atestada a possibilidade da aplicação de AGs, buscou-se aperfeiçoar a preparação e adaptação, aprofundar os estudos visando superar os obstáculos encontrados, tais como a dimensão dos indivíduos, aplicação da técnica a sistemas mais próximos da realidade, de grande porte, com grandes cascatas interligadas e com condições operativas mais severas, com também aperfeiçoar o desempenho do algoritmo com a introdução de novos recursos computacionais. A seguir serão apresentadas, no algoritmo proposto, todas as adaptações feitas no algoritmo tradicional para resolução do POSHP, e finalizando, será mostrado um Estudo, composto de 35 usinas hidroelétricas com o objetivo de verificar a viabilidade do algoritmo proposto.

No intuito de tornar bem claro o uso dos AGs no problema em questão, optou-se por apresentar os cálculos dos operadores genéticos com dados de uma usina real, partindo desde os dados básicos da usina, e das afluições, e desenvolvendo de forma detalhada as etapas do processo. Assim, apresenta-se a seguir o exemplo onde foi utilizada a usina hidroelétrica de Emborcação, pertencente ao Sistema Sudeste Brasileiro.

Nas Tabelas 1 e 2<sup>1</sup> são apresentados os dados da usina hidroelétrica de Emborcação utilizada na otimização.

Tabela 1. Dados da usina hidroelétrica de Emborcação.

Potência Instalada [MW]	Volume [ $hm^3$ ]			Vazão turbinada [ $hm^3/m$ ]		Taxa de juros[%]	Período de planejamento
	Mínimo	Máximo	Útil	Mínimo	Máximo		
1200	4669.0	17190.0	12521.0	202.4	2754.1	10	1 ano

Tabela 2. Dados de demanda e afluência para resolução do problema proposto.

	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr
Coefficiente de demanda	0,991	1,008	1,030	1,037	1,034	1,047	1,025	1,021	0,923	0,947	0,969	0,968
Afluência ( $m^3/s$ )	407,0	311,0	246,0	191,0	162,0	208,0	359,0	647,0	888,0	917,0	855,0	634,0

Na Figura 3, observa-se os passos utilizados para determinar o volume ótimo da usina nos 12 meses, através da aplicação de Algoritmos Genéticos.

Durante o desenvolvimento da ferramenta computacional algumas etapas do diagrama apresentado na Figura 3 foram modificadas. Entre estas, a população inicial e o critério de seleção, bem como o acréscimo dos seguintes operadores genéticos: mutação local, mutação direcionada, mutação induzida, mutação gradiente direcionada e mutação gradiente.

Outra melhoria significativa diz respeito à adoção de um gerador de números aleatórios mais eficiente (Matsumoto & Nishimura, 1998), o que resultou em um melhor desempenho do algoritmo proposto.

A seguir, apresenta-se a descrição do procedimento utilizado no algoritmo proposto.

#### 4.1 Codificação do problema

A codificação do problema de Planejamento da Operação Energética precisa ser realizada com cuidado, já que este é um problema interligado no

<sup>1</sup> O coeficiente de demanda multiplicado pela potência instalada, definida no arquivo de entrada, fornecerá a carga a ser atendida em cada mês do período de planejamento

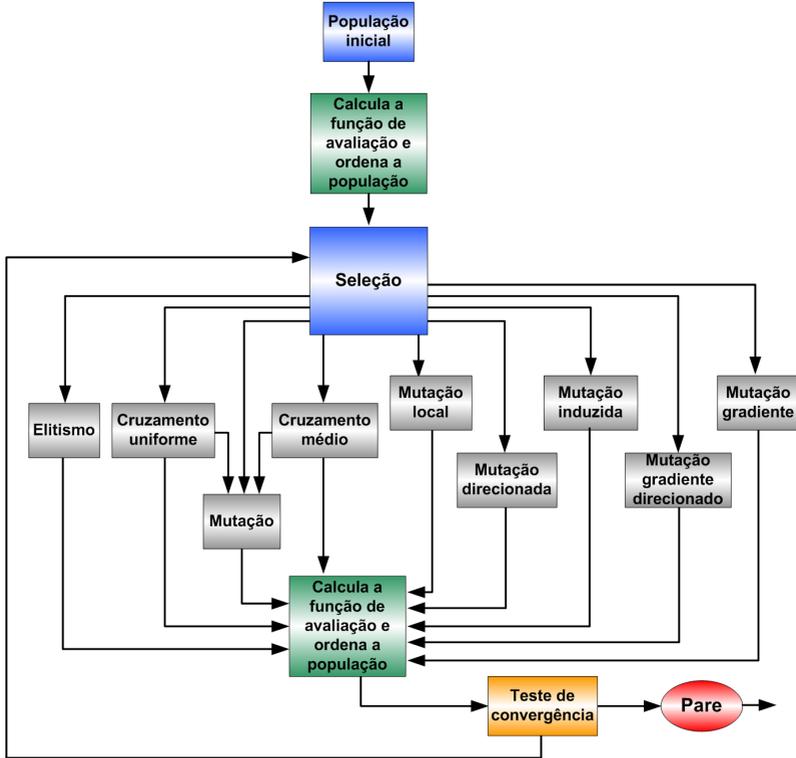


Figura 3. Algoritmo proposto.

tempo, pois a decisão tomada em um determinado período afeta diretamente todos os outros. Essa etapa é considerada a mais importante, pois a viabilidade do algoritmo depende da sua codificação. Para uma melhor representação do problema, adotam-se valores reais e não binários, como é usual, e o indivíduo representará o volume da usina em cada mês do planejamento.

O Sistema Brasileiro é composto por várias usinas em cascata, como já mencionado anteriormente. Portanto, para representar o volume de todas as usinas em um único vetor, ter-se-ia uma cadeia de características muito longa. Por exemplo, para o planejamento de 2 anos (24 características) com 35 usinas, o indivíduo teria 840 genes. Sendo assim, optou-se por uma representação matricial tridimensional. Nesta matriz, no eixo  $x$ , estão representados os genes de um indivíduo de acordo com o período de planejamento, neste caso 12 meses, no eixo  $y$ , representam-se os indivíduos

da população e no eixo z, está caracterizado o número de usinas envolvidas no sistema teste, conforme mostrado na Figura 4.

Por exemplo, em um sistema teste com 3 usinas, 5 indivíduos e cada indivíduo com 12 características (genes) de acordo com o período de planejamento, a população ficaria configurada, conforme apresentada na Figura 4.

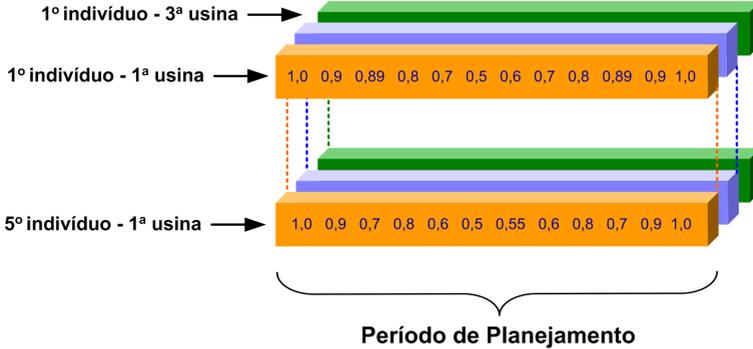


Figura 4. População.

O tamanho da população afeta diretamente o desempenho global e a eficiência dos AGs. Com uma população pequena o desempenho pode cair, pois a população fornece uma pequena cobertura do espaço do problema. Uma população grande geralmente fornece uma cobertura representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, para se trabalhar com populações grandes, são necessários maiores recursos computacionais, ou que o algoritmo trabalhe por um período de tempo muito maior, a fim de obter um resultado satisfatório (Rezende, 2003).

Esta representação reflete a realidade dos reservatórios, nos quais o  $i$ -ésimo elemento de um indivíduo representa o  $i$ -ésimo volume útil, isto é, o volume operativo no período  $i$  do planejamento, que é discretizado em meses.

Sabendo-se que: volume máximo operativo da 1º usina = 17190,0; volume mínimo operativo da 1º usina = 4669,0;  $VOL$  = volume da 1º usina, por unidade, em relação ao volume máximo operativo;  $VOL[i][j][k]$  = volume do indivíduo  $[i]$ , usina  $[j]$  e gene (mês) ou característica  $[k]$ ;  $VOL[1][1][1] = 1,00$ ;  $VOL[1][1][2] = 0,90$ , pode-se calcular o volume real que os genes 1 e 2 representam:

$$V_1 = V_{min} + (V_{max} - V_{min}) * VOL_1$$

$$V_1 = 4669,0 + (17190,0 - 4669,0) * 1,00$$

$$V_1 = 4660,0 + (12521,0) * 1,00$$

$$V_1 = 17190,0$$

Calculando o valor real do segundo gene têm-se:

$$\begin{aligned} V_2 &= V_{min} + (V_{max} - V_{min}) * VOL_2 \\ V_2 &= 4669,0 + (17190,0 - 4669,0) * 0,90 \\ V_2 &= 4660,0 + (12521,0) * 0,90 \\ V_2 &= 15937,90 \end{aligned}$$

Desta maneira, determina-se o volume real de todos os indivíduos da população. Após a codificação do problema, o próximo passo é a definição da população inicial.

## 4.2 Solução inicial

No início do desenvolvimento da ferramenta utilizou-se apenas solução inicial aleatória, pois era preciso confirmar a viabilidade da técnica quando aplicada ao POSHP, sem que fosse fornecido ao algoritmo um conhecimento prévio do comportamento das usinas para um determinado período de planejamento (Leite, 2003). Confirmada a viabilidade da técnica e na tentativa de diminuir o tempo computacional exigido para convergência, optou-se por fornecer ao algoritmo informações já conhecidas através dos estudos do POSHP, realizados utilizando uma técnica tradicional de otimização (Carneiro et al., 1990).

Após vários testes foram determinadas 5 formas de solução inicial. Na próxima seção esses testes serão mais detalhados para que se possa analisar melhor as investigações realizadas. Portanto, a população inicial poderá ser gerada adotando uma das soluções iniciais a seguir:

1. **Aleatória:** para cada indivíduo valores reais aleatórios entre 0 e 1 são gerados. Cada gene representa o volume em porcentagem para cada mês do período de planejamento.
2. **Volume sugerido:** nesse caso, o indivíduo pode ter os valores de suas características definidas pelo operador do sistema. Por exemplo, todos os genes iguais a 1,0, ou seja, 100% do volume útil. Neste caso, e em outros onde o volume permaneça constante durante todo o período de planejamento, justamente por não haver variação do volume, toda a água que chega na usina é turbinada ou vertida, caracterizando o que é chamado na literatura de solução a fio-d'água (Carneiro, 1991). Normalmente, é utilizada quando se faz estudos com a ferramenta baseada em programação não-linear. Foi isso que motivou o uso dessa solução nos testes com AGs.

Adicionada essa possibilidade ao algoritmo, pode-se variar as características dos genes com base no comportamento da usina operando em cascata, tendo eles valores iniciais iguais a 60%, 70%, 80%, etc.,

do volume útil do reservatório. Acredita-se que essa informação poderá acelerar a convergência do algoritmo.

3. **Volume sugerido + aleatória:** seguindo a idéia apresentada no item 2, adota-se como solução inicial para as usinas de cabeceira características aleatórias, pois sabe-se, através de estudos, que seus volumes devem oscilar para atender as variações do mercado e das afliências. Nas demais usinas, os indivíduos terão características iguais a 100%. Por exemplo, no caso de um sistema teste com três usinas, Emborcação, Itumbiara e São Simão, nas usinas 1 e 2 os genes iniciais serão aleatórios e na usina de São Simão iguais a 100%. Através das sucessivas gerações essas características poderão mudar conforme a necessidade de atendimento do mercado e o período da afliência, ou seja, esse comportamento inicial será apenas um indicativo de operação ótima.
4. **Solução inicial fornecida pelo algoritmo de otimização (PNL):** sendo uma das propostas do projeto de doutorado a de se trabalhar com algoritmos híbridos, ou seja, os AGs em conjunto com outras técnicas de otimização, foi utilizada a saída fornecida pelo PNL como solução inicial dos AGs. Essa alternativa vem mostrando um bom desempenho, o que pode ser visto nos testes apresentados na próxima seção.
5. **PNL + 100% (solução a fio-d'água):** em uma população de 20 indivíduos o primeiro será fornecido pelo PNL e os demais terão características iguais a 100%, ou seja, solução a fio-d'água.

### 4.3 Função de avaliação

A função de avaliação ou função de custo está relacionada à minimização ou maximização do valor esperado da função objetivo do problema.

No caso do planejamento da operação, a função de avaliação será o custo operativo no período de planejamento em conjunto com a adaptação do indivíduo, onde a cada restrição operativa do problema satisfeita, definidas na formulação matemática, é dado ao indivíduo um ponto. O mais pontuado será considerado mais apto, sendo esta uma maneira de avaliar os membros do espaço de busca.

Outra opção para verificar a adaptação do indivíduo seria a soma das falhas, onde considera-se a distância entre o valor da restrição operativa não atendida até o mínimo valor que esta restrição é aceita. Quanto menor for este somatório, mais apto será o indivíduo.

### 4.4 Ordenação da população

O algoritmo proposto classifica a população calculando a função de avaliação de todos os indivíduos e depois ordenando-os conforme o objetivo do problema. No exemplo mencionado, o objetivo será o de minimizar o custo

e atender as restrições operativas. Portanto, a população será ordenada do menor para o maior custo, sabendo que, o primeiro indivíduo terá o menor custo, desde que tenha atendido as restrições operativas.

#### 4.5 Seleção

A seleção é feita para determinar em quais e em quantos indivíduos serão aplicados os diversos tipos de operadores genéticos utilizados, conforme Figura 5.

Tela de configuração dos parâmetros do Algoritmo Genético e do Planejamento da Operação

Nome do Arquivo: Teste 35 usinas  
 Número de Gerações: 900000  
 Tamanho da População: 24

Tempo de Planejamento:  1 ano  2 anos  3 anos  4 anos  5 anos

Tipo de População Inicial:  
 Aleatória  
 Volume Sugerido (%)  
 Volume Sugerido (%) + Aleatório | 100  
 Solução Inicial Fornecida pelo PNL  
 PNL + 100%  
 Endereço do arquivo TXT de entrada da solução inicial do PNL:

Distribuição da População:  
 Elitismo: 2  
 Cruzamento Uniforme: 1  
 Cruzamento Uniforme + Mutação: 4  
 Cruzamento Médio: 1  
 Cruzamento Médio + Mutação: 1  
 Mutação Simples: 2  
 Mutação Direcionada: 1  
 Mutação Gradiente: 2  
 Mutação Gradiente Direcionado: 4  
 Mutação flo d'água: 1  
 Busca Local: 2

Propriedades de Cruzamento:  
 Taxa de Seleção: 0.1  
 Taxa de Cruzamento: 0.1  
 Taxa de Cruzamento Médio: 0.1

Propriedades de Mutação:  
 Pontos de Mutação Simples: 10  
 Pontos de Mutação Direcionada: 10  
 Pontos de Mutação Gradiente: 5  
 Pontos de Mutação Gradiente Direcionado: 5  
 Pontos de Mutação flo d'água: 1  
 Taxa de Mutação: 0.1

Técnica de comparação:  
 Sem comparação  
 Com comparação  
 Nome da técnica de comparação: PNL  
 Endereço do arquivo TXT de técnica de comparação: C:\Saida\_PNL

Custo da Técnica: 0.21  
 Taxa de Desconto: 0.1

Tipo de Mercado:  
 Constante  Sazonal

Gráfico de Volume:  
 Todos as Usinas / Somente 1 gráfico  
 Somente as Usinas com Reservatório / Somente 1 Gráfico  
 Todas as Usinas / Gráficos Divididos por Rio  
 Somente as Usinas com Reservatório / Gráficos Divididos por Rio

Executar

Figura 5. Tela para configuração dos parâmetros utilizados pelos AGs.

Note que os dados contidos na Figura 5 não são fixos. Estes foram utilizados para resolver o problema de uma usina. Cada problema exige uma determinação de parâmetros que podem ou não ser iguais a de um caso já determinado. Isso foi definido nos testes de ponderação dos operadores apresentados na próxima seção desse capítulo.

Observe que os valores de taxas de seleção, taxas de cruzamentos, taxas de mutações e quantos indivíduos sofrerão aplicação de cada tipo de operador genético, estão definidos em um arquivo de entrada, conforme Figura 5. As taxas serão utilizadas para direcionar o sorteio, podendo privilegiar os indivíduos que ocupam as primeiras posições dentro da população.

O método de seleção mais simples e também mais conhecido é o método da Roleta. Nele os indivíduos de uma população, serão selecionados para formarem uma parte da próxima geração, utilizando o mesmo processo da roleta utilizada em jogos. Neste método cada indivíduo é representado na roleta por uma fatia proporcional ao seu índice de aptidão, para que, os indivíduos com maior aptidão tenham maiores possibilidades de serem escolhidos, ou seja, uma área maior na roleta, o contrário ocorrendo com os

menos aptos. Se a roleta simples fosse baseada na posição do pai, o ponto de corte é obtido da seguinte maneira:

1. Corte = posição do pai 2;
2. Sorteia-se um número  $n$ , entre 0 e posição do pai 1 + posição do pai 2:  $n = \text{random}(\text{pai 1} + \text{pai 2})$ ;
3. Se  $n < \text{corte}$ , escolhe pai 1;
4. Senão, escolhe pai 2.

Utilizando a roleta simples, o grau que privilegia os indivíduos nas melhores posições é de taxa fixa. Dependendo do problema, haverá melhor resultado se existirem graus diferentes de privilégios. Portanto, o método de seleção adotado neste trabalho é o método da roleta ponderada, baseado na posição dos pais.

Então pode-se pensar, para pai 1 e pai 2, da seguinte maneira: atribui-se mais oportunidade a um ou ao outro indivíduo, dando às suas posições graus de privilégios diferentes. Adota-se um valor entre essas duas posições que é calculado pela média ponderada pelos graus de privilégios. Então trabalha-se com uma roleta ponderada que é mais flexível que a roleta simples.

Na roleta pela posição ponderada:

1. Corte = taxa de seleção \* (pai 1) + (1 - taxa de seleção) \* (pai 2);
2. Se taxa de seleção = 0,8, definida pelo usuário no intervalo [0 a 1]:  
Corte =  $0,8 + 0,2 * 4 = 1,6$ ;
3. Se taxa de seleção = 0,1: Corte =  $0,1 + 0,9 * 4 = 3,7$ ;
4. Sorteia-se um número  $n$  entre 0 e posição do (pai 1 + pai 2):  $n = \text{random}(\text{pai 1} + \text{pai 2}) = 5$ ;
5. Se  $n < \text{Corte}$ , escolhe pai 1,;
6. Senão, escolhe pai 2.

Ou seja, quanto menor a taxa de seleção, maior a chance de se escolher o melhor indivíduo. Pois com uma taxa de seleção = 0,1, o valor de corte é igual a 3,7, como o valor  $n$  sorteado está no intervalo [0 a 5] a chance de sortear um número menor do que 3,7 é grande.

#### 4.6 Aplicação dos operadores genéticos

A seguir serão apresentados os operadores genéticos utilizados nesse trabalho. Como mencionado anteriormente, alguns foram alterados e outros acrescentados para se adaptarem às condições do problema de POSHP.

**Elitismo:** para garantir que os melhores indivíduos não sejam perdidos de uma geração para a outra, aplica-se o elitismo. Isto garante que os melhores indivíduos passem para a próxima geração preservando suas

características originais.

**Cruzamento uniforme:** quando aplica-se o cruzamento uniforme há a necessidade de se determinar de qual dos pais serão copiadas as características para criação do filho.

A maneira utilizada neste trabalho foi a seguinte: pode-se optar por três tipos de taxas de cruzamento diferentes, segundo a Tabela 3. Conforme a taxa adotada, utilizando uma função de ajuste (Schlick, 1994), o filho terá mais características do melhor pai. Para permitir que isso ocorra de forma bem aleatória, nada impede que o novo descendente tenha características dos dois pais na mesma proporção. Apenas é dado, através do sorteio, um peso maior aos melhores indivíduos, ou seja, os que estiverem nas primeiras posições poderão ser beneficiados.

Tabela 3. Tipos de taxas adotadas na seleção, cruzamento ou mutação.

Valores das taxas	Probabilidade de sorteio
taxa < 0.5	aumenta a probabilidade de sortear os indivíduos nas melhores posições.
taxa = 0.5	probabilidade de sortear os indivíduos de forma uniforme na população.
taxa > 0.5	aumenta a probabilidade de sortear os indivíduos nas piores posições.

Vale lembrar que, quanto menor for a taxa adotada, maior será a chance de selecionar o melhor indivíduo, devido à função de ajuste (Schlick, 1994). Exemplo: supondo que sejam sorteados dois indivíduos para sofrer cruzamento uniforme: pai 1 na posição 4 e pai 2 na posição 7.

Cria-se um indivíduo chamado máscara onde o valor de cada gene será aleatório. Este valor será baseado no somatório das posições escolhidas para sofrerem cruzamento uniforme, conforme Figura 6. Exemplo: pai 1, na posição 4; pai 2, na posição 7 e privilegiando o melhor indivíduo, adotando taxa = 0,1.

onde:

$$\begin{aligned} \sum_{posições} &= pai1 + pai2 + 2 \\ \sum_{posições} &= 4 + 7 + 2 \\ \sum_{posições} &= 13 \end{aligned}$$

Portanto, o sorteio será no intervalo de 0 a 13.

Baseado na taxa de cruzamento adotada, calcula-se o valor de  $\alpha$ , ou seja o corte, que será utilizado para comparar com cada gene da máscara. Se o gene tiver valor menor que  $\alpha$  copia-se característica do pai1, caso

máscara											
3,0	9,0	2,0	1,0	5,0	8,0	0	10	7,0	12	4,0	11
pai1											
1,00	0,92	0,85	0,74	0,60	0,50	0,50	0,60	0,74	0,85	0,92	1,00
Filho											
1,00	0,90	0,85	0,74	0,60	0,49	0,50	0,59	0,74	0,82	0,92	1,00
pai2											
1,00	0,90	0,82	0,71	0,59	0,49	0,49	0,59	0,71	0,82	0,90	1,00

Figura 6. Cruzamento uniforme proposto.

contrário copia-se característica do pai 2, conforme Figura 6.

Cálculo da variável  $\alpha$  (corte):

privilegiar o melhor  $\Rightarrow taxa = 0,1$

$$\alpha = taxa * (pai1 + 1) + (1 - taxa) * (pai2 + 1)$$

$$\alpha = 0,1 * (4 + 1) + (1 - 0,1) * (7 + 1)$$

$$\alpha = 0,1 * 5 + 0,9 * 8$$

$$\alpha = 0,5 + 7,2$$

$$\alpha = 7,7$$

indiferente  $\Rightarrow taxa = 0,5$

$$\alpha = taxa * (pai1 + 1) + (1 - taxa) * (pai2 + 1)$$

$$\alpha = 0,5 * (4 + 1) + (1 - 0,5) * (7 + 1)$$

$$\alpha = 0,5 * 5 + 0,5 * 8$$

$$\alpha = 2,5 + 4,0$$

$$\alpha = 6,5$$

privilegiar o pior  $\Rightarrow taxa = 0,9$

$$\alpha = taxa * (pai1 + 1) + (1 - taxa) * (pai2 + 1)$$

$$\alpha = 0,9 * (4 + 1) + (1 - 0,9) * (7 + 1)$$

$$\alpha = 0,9 * 5 + 0,1 * 8$$

$$\alpha = 4,5 + 0,8$$

$$\alpha = 5,3$$

Pode-se observar no exemplo, que utilizando uma taxa pequena (0,1) o indivíduo recebeu mais características do pai 1, como mostrado na Figura 6, pois o valor do corte é igual 7,7, ou seja o intervalo de sorteio que beneficia o melhor pai é maior. Mas pode acontecer, como já citado anteriormente, que o indivíduo tenha a mesma proporção, quando a taxa for igual a 0,5,

ou até mais características do pai 2, quando a taxa adotada for maior que 0,5, pois trata-se de um algoritmo aleatório.

**Cruzamento médio:** a idéia de se aplicar um cruzamento baseado na média surgiu com o objetivo de obter um indivíduo que tivesse características dos dois pais e não apenas de um.

Por exemplo, no caso de cruzamento racial, entre um branco e um negro, o descendente deste cruzamento poderá ser um branco, um negro ou um mulato. Em outras palavras, o descendente poderá se situar em qualquer posição entre os extremos, não havendo uma raça ou característica contínua, em que não há interrupção, o filho poderá ter qualquer característica entre os dois valores dos pais.

Logo, no caso em que utiliza-se gene real, codificando características contínuas como um gene, o cruzamento onde o filho herda características intermediárias dos pais, será bem sucedido.

Quando aplica-se o cruzamento médio há a necessidade de se determinar o valor obtido da soma das características (genes) do casal escolhido para criar um descendente, como já exposto acima. Pode-se aplicar a média ponderada entre as características dos dois pais.

Portanto, este trabalho propõe uma maneira de calcular a média de forma a beneficiar os indivíduos que ocupam as melhores posições, ou seja, a melhor característica de um indivíduo pode estar próxima a um valor do melhor pai.

Como no cruzamento uniforme, pode-se optar por tipos de taxas de cruzamento diferentes. Conforme a taxa adotada, tem-se uma distribuição (Schlick, 1994), que proporciona ao filho obter características próximas do melhor indivíduo. Esse tipo de distribuição, tanto no cruzamento uniforme como no cruzamento médio, dá maiores chances aos indivíduos nas primeiras posições de serem sorteados, que no nosso caso são os melhores, semelhante ao método da roleta.

Seja o seguinte exemplo. Supondo que sejam sorteados dois indivíduos para sofrerem cruzamento médio, pai 1 na posição 19 e pai 2 na posição 22.

O  $gene_{novo}$  do filho será obtido através da média ponderada entre as características dos dois pais da seguinte forma:  $gene_{novo} = T * gene_{pai1} + (1 - T) * gene_{pai2}$ , onde:  $T$  será calculado pela seguinte função:  $T = BiasRand(a)$ .  $BiasRand(a)$  devolve um valor para  $T$  entre 0 e 1, segundo o valor de “a”. Se o valor de “a” for menor que 0,5, a probabilidade de se sortear um número próximo de zero é maior. Se for igual a 0,5, a probabilidade de sortear um número entre 0 e 1 é uniforme. Finalmente, se o valor for maior que 0,5, a probabilidade de sortear um valor próximo de 1 é maior.

Diante da necessidade de obter um valor para “a” que pudesse ser usado como uma taxa, ou seja  $\in [0, 1]$ , calcula-se o valor de “a” através

da média ponderada entre as posições que os pais ocupam na população,  $pa_{i1}+1$  e  $pa_{i2}+1$ , dividida pelo somatório dessas duas posições. Assim, pretende-se relacionar a participação do gene de acordo com a sua chance na roleta, tomando “a” como a probabilidade de ser sorteado na roleta ponderada, seja o exemplo a seguir:

privilegiar o melhor  $\Rightarrow taxa = 0,1$

$$a = (taxa * (pa_{i1} + 1) + (1 - taxa) * (pa_{i2} + 1)) / (pa_{i1} + pa_{i2} + 2)$$

$$a = (0,1 * (19 + 1) + (1 - 0,1) * (22 + 1)) / (19 + 22 + 2)$$

$$a = (2 + ((0,9 * 23))) / 43$$

$$a = (2 + 20,7) / 43$$

$$a = 0,53$$

indiferente  $\Rightarrow taxa = 0,5$

$$a = (taxa * (pa_{i1} + 1) + (1 - taxa) * (pa_{i2} + 1)) / (pa_{i1} + pa_{i2} + 2)$$

$$a = (0,5 * (19 + 1) + (1 - 0,5) * (22 + 1)) / (19 + 22 + 2)$$

$$a = (10 + ((0,5 * 23))) / 43$$

$$a = (10 + 11,5) / 43$$

$$a = 0,5$$

privilegiar o pior  $\Rightarrow taxa = 0,9$

$$a = (taxa * (pa_{i1} + 1) + (1 - taxa) * (pa_{i2} + 1)) / (pa_{i1} + pa_{i2} + 2)$$

$$a = (0,9 * (19 + 1) + (1 - 0,9) * (22 + 1)) / (19 + 22 + 2)$$

$$a = (18 + ((0,1 * 23))) / 43$$

$$a = (18 + 23) / 43$$

$$a = 0,47$$

De acordo com a taxa que privilegia o melhor indivíduo e os valores de  $a$  e  $T$ , pode-se obter o novo valor do gene 1, conforme Figura 7, da seguinte maneira:

Para  $a = 0,57$  e  $T = 0,8$

$$gene\ 1 = T * gene\ 1_{p1} + (1 - T) * gene\ 1_{p2}$$

$$gene\ 1 = 0,8 * 0,90 + (1 - 0,8) * 0,75$$

$$gene\ 1 = 0,72 + 0,2 * 0,75$$

$$gene\ 1 = 0,72 + 0,15$$

$$gene\ 1 = 0,87$$

Caso fosse adotada taxa = 0,5 ou taxa que privilegie o pior indivíduo, o valor do gene 1 seria:

Para  $a = 0,5$  e  $T = 0,5$

$$gene\ 1 = T * gene\ 1_{p1} + (1 - T) * gene\ 1_{p2}$$

$$gene\ 1 = 0,5 * 0,90 + (1 - 0,5) * 0,75$$

$$gene\ 1 = 0,45 + 0,5 * 0,75$$

$$gene\ 1 = 0,45 + 0,375$$

$$gene\ 1 = 0,825$$

Para  $a = 0,47$  e  $T = 0,2$

$$\begin{aligned}
 gene\ 1 &= T * gene\ 1_{p1} + (1 - T) * gene\ 1_{p2} \\
 gene\ 1 &= 0,2 * 0,90 + (1 - 0,2) * 0,75 \\
 gene\ 1 &= 0,18 + 0,5 * 0,75 \\
 gene\ 1 &= 0,18 + 0,60 \\
 gene\ 1 &= 0,78
 \end{aligned}$$

pai1											
1,00	<b>0,90</b>	0,85	0,74	0,60	0,49	0,50	0,59	0,74	0,82	0,91	1,00
pai2											
1,00	<b>0,75</b>	0,65	0,64	0,50	0,49	0,49	0,50	0,64	0,65	0,75	1,00
Filho											
1,00	<b>0,87</b>	0,74	0,68	0,60	0,49	0,49	0,54	0,68	0,73	0,82	1,00

Figura 7. Cruzamento médio proposto.

**Mutação:** com o objetivo de manter a diversidade da população, o operador genético de mutação será aplicado em alguns descendentes criados através de cruzamento uniforme, cruzamento médio e em alguns que não sofreram nenhum tipo de cruzamento.

Trabalhando com características contínuas, há a necessidade de uma adaptação do operador de mutação. O binário muda de 0 para 1 ou vice-versa. No problema proposto, isso não será viável. Portanto, adota-se uma maneira de mutar os indivíduos de uma forma não brusca, ou seja, não é coerente mudar o volume do reservatório de 90% para 45%. Porém, pode-se pensar em uma maneira de conseguir que o novo valor do gene mutado esteja próximo do gene original.

Primeiramente, o número de pontos de mutação será considerado como sendo um dado de entrada, conforme Figura 5. Entretanto, quando da aplicação deste operador genético, é feito um sorteio no intervalo de 0 (zero) ao número de pontos de mutação determinado, ou seja, não necessariamente será modificada a quantidade de pontos pré-determinados.

Após sorteada a quantidade de pontos que serão modificados, faz-se uma lista com cada posição sorteada para sofrer mutação, evitando que um mesmo ponto seja escolhido mais de uma vez.

Exemplo: supondo que o indivíduo na 25ª colocação foi escolhido para sofrer mutação. Determina-se, por sorteio, a quantos pontos de um indivíduo será aplicada a mutação. Supondo que o número de pontos determinados é igual a 5, conforme mostrado na Figura 8, faz-se:

1. Sortear um número que está no intervalo de [0 a 5]. Supondo que o número sorteado seja igual a 4;
2. Montar uma lista com pontos a serem mutados. Sabendo que esta lista deverá conter a informação das posições dos genes a serem alteradas.

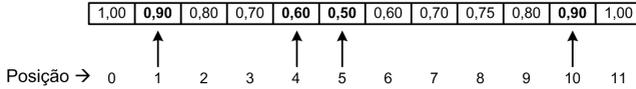


Figura 8. Mutação proposta.

- Sorteia-se qualquer posição do indivíduo: A posição 1 foi sorteada. Esta posição não entra mais no sorteio das próximas posições
  - Sorteia-se qualquer posição do indivíduo: A posição 4 foi sorteada. Esta posição corresponde à posição 5 no indivíduo, pois a posição sorteada não conta mais.
  - Sorteia-se qualquer posição do indivíduo: A posição 3 foi sorteada. Esta posição corresponde à posição 4 no indivíduo.
  - Sorteia-se qualquer posição do indivíduo: A posição 7 foi sorteada. Esta posição corresponde à posição 10 no indivíduo.
3. Obter os valores novos para cada posição sorteada.
- Para a primeira posição sorteia-se um valor próximo do valor original 0,9, com uma taxa de 0,1. Com esta taxa de mutação o valor sorteado pode ser qualquer número próximo do valor original do gene, segundo a função *Gain* (Schlick, 1994), exemplificada a seguir. Aplicada a função, o valor do novo gene passa a ser 0,92. Repete-se o procedimento para as demais posições, usando:  $gene_{novo} = CGainRand(gene, taxa)$ , onde:  $CGainRand((gene), taxa)$  devolve um valor entre 0 e 1, segundo a taxa escolhida. Se o valor da taxa for menor que 0,5, a probabilidade de se sortear um número próximo do valor do gene escolhido para ser mutado é maior. Se for igual a 0,5, a probabilidade é igual para um valor entre 0 e 1. Finalmente se o valor for maior que 0,5, a probabilidade de sortear um valor próximo do valor do gene escolhido para ser mutado é pequena. Portanto:
  - Para a posição 4 o novo valor será de 0,65.
  - Para a posição 5 o novo valor será de 0,49.
  - Para a posição 9 o novo valor será de 0,88.
4. Finalmente, têm-se o novo indivíduo, apresentado na Figura 9.

**Mutação direcionada:** como se trata de um problema onde, quanto mais uniforme for a geração térmica menor será o custo da operação, faz-se

1,00	<b>0,92</b>	0,80	0,70	<b>0,65</b>	<b>0,49</b>	0,60	0,70	0,75	0,80	<b>0,88</b>	1,00
------	-------------	------	------	-------------	-------------	------	------	------	------	-------------	------

Figura 9. Novo indivíduo após mutação proposta.

um rastreamento de todos os valores de maior e menor geração (picos), ocorridos durante o período de planejamento, e aplica-se mutação nestes pontos, de forma a manter o valor de geração o mais uniforme possível.

**Mutação induzida:** através de vários estudos é suposto que se conhece o comportamento ótimo esperado das usinas em cascata, quando em condições ideais. Como as usinas ocupam posições diferentes no conjunto, as características de operação são diferenciadas. As usinas à montante assumem a função de regularização do sistema, o que significa que oscilam no volume de seus reservatórios, enquanto as usinas à jusante mantêm os seus reservatórios cheios durante o período de planejamento. Baseado na operação ótima de reservatórios, estudada pela técnica tradicional de otimização, o operador de mutação induzida daria maiores chances aos genes correspondentes às usinas à montante de variarem os seus volumes, enquanto os genes correspondentes às usinas a jusante permaneceriam inalterados, ou seja, com 100% do volume útil, o que reduziria o tempo computacional, pois fazem com que as usinas à reservatório de fim de cascata tenham características de operação a fio-d'água, ou seja, não esvaziem seus reservatórios.

**Mutação local:** como a aplicação foi feita em um problema complexo e de grande porte, para evitar que o algoritmo demore a convergir, a partir do momento em que se tem um indivíduo que atenda a todas as restrições é feita uma verificação a cada  $n$  iterações, pré-definida pelo operador do sistema. Nesta etapa, para os indivíduos com os mesmos valores de genes, isto é, indivíduos idênticos, somente um é mantido na população, enquanto os outros sofrerão alterações nas suas características com objetivo de explorar a vizinhança, ou seja, obter um outro indivíduo com características próximas do que foi descartado e que possa se aproximar da solução ótima, conforme mostrado na Figura 10. Essas alterações seguem os mesmos mecanismos utilizados na mutação, respeitando uma taxa pré-determinada pelo operador, só que, neste caso, todas as características serão alteradas.

Indivíduo antes de aplicar busca local

1,00	0,91	0,87	0,75	0,63	0,55	0,56	0,63	0,75	0,87	0,91	1,00
------	------	------	------	------	------	------	------	------	------	------	------

Depois da busca local

1,00	0,90	0,85	0,74	0,65	0,56	0,57	0,65	0,74	0,85	0,90	1,00
------	------	------	------	------	------	------	------	------	------	------	------

Figura 10. Mutação local proposta.

A literatura internacional tem mostrado a importância e o sucesso dos algoritmos híbridos, onde uma técnica tradicional de otimização trabalha em conjunto com os AGs (Gen & Cheng, 1997; Corne et al., 1999). Portanto, optou-se por implementar o método do gradiente (Salomon, 1998; Bazaraa et al., 1993), dando origem a dois operadores genéticos de mutação, que foram definidos como mutação gradiente e mutação gradiente direcionada. Estes serão descritos abaixo.

**Mutação gradiente:** para melhor compreensão dos dois operadores genéticos baseados no método do gradiente, será mostrado brevemente a forma de calculá-lo para uma função  $f(x)$ .

O gradiente, ou velocidade de variação, de  $f(x)$  em um ponto  $x$ , pode ser aproximado por  $\Delta f/\Delta x$ , e pode-se escrevê-lo exatamente como na Equação 21, que é conhecida como a derivada parcial de  $f(x)$  em relação a  $x$ .

$$\frac{\partial f(x)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (21)$$

Suponha que se deseja minimizar a função  $f(x)$  e efetuar uma variação em  $x$  para minimizar  $f(x)$ . Precisa-se para isso conhecer a derivada de  $f(x)$ . Há três casos a considerar:

- Se  $\partial f/\partial x > 0$ , então  $f(x)$  aumenta quando  $x$  aumenta, por isso se deve diminuir  $x$ .
- Se  $\partial f/\partial x < 0$ , então  $f(x)$  diminui quando  $x$  aumenta, por isso se deve aumentar  $x$ .
- Se  $\partial f/\partial x = 0$ , então  $f(x)$  é um máximo ou um mínimo, ou seja um ponto de inflexão ou ponto de sela, por isso não se deve variar  $x$ .

Portanto,

$$x(k) = x(k-1) - \eta \frac{\partial f}{\partial x} \quad (22)$$

em que  $\eta$  é uma constante positiva, pequena, que especifica quanto se deve variar  $x$  (na iteração  $k$ ) e a derivada  $\partial f/\partial x$  indica a direção dessa variação. Se, repetidamente, esta equação for usada,  $f(x)$  continuará a descer em relação ao seu valor mínimo (desde que  $\eta$  seja suficientemente pequeno).

No caso do problema do POSHP, escolhe-se um indivíduo da população, alguns pontos (gene) são sorteados aleatoriamente, e calcula-se o gradiente nestes pontos visando a minimização da função objetivo.

Por exemplo: o indivíduo  $i$  foi sorteado na população e nos pontos 2, 7 e 10 serão calculados os gradientes. Calculando o custo, conforme definido na formulação matemática, para o indivíduo  $i$ , tem-se:  $custo(i) = 550219,3222$ , custo do indivíduo antes da aplicação da mutação gradiente.

A seguir acrescenta-se um valor  $\Delta x = 0,00001$  em cada ponto escolhido e calcula-se novamente o custo do indivíduo  $i$ .

$$\text{custo}(x_2) = 550217,7717, \text{ quando } x_2 = 0,96651;$$

$$\text{custo}(x_7) = 550216,9448, \text{ quando } x_7 = 0,73541;$$

$$\text{custo}(x_{10}) = 550221,1841, \text{ quando } x_{10} = 0,91181;$$

O próximo passo consiste em calcular a derivada de  $f(x)$  em cada ponto:

$$\frac{\partial f(x)}{\partial x} = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

$$\frac{\partial f(x)}{\partial x_2} = \frac{f(x_2 + \Delta x) - f(x)}{\Delta x} = \frac{\text{custo}(i_2) - \text{custo}(i)}{0,00001} = -155048,653$$

$$\frac{\partial f(x)}{\partial x_7} = \frac{f(x_7 + \Delta x) - f(x)}{\Delta x} = \frac{\text{custo}(i_7) - \text{custo}(i)}{0,00001} = -237734,200$$

$$\frac{\partial f(x)}{\partial x_{10}} = \frac{f(x_{10} + \Delta x) - f(x)}{\Delta x} = \frac{\text{custo}(i_{10}) - \text{custo}(i)}{0,00001} = 186190,928$$

Conforme mencionado anteriormente,  $\eta$  tem que ser pequeno. Então  $\eta$  será obtido através de um número sorteado dividido por  $\alpha$ . Sendo  $\alpha$  dado pela normalização da derivada da função  $f$  em relação aos pontos escolhidos para aplicação do operador. Portanto, normalizando:

$$\alpha^2 = \left( \frac{\delta f}{\delta x_2}, \frac{\delta f}{\delta x_7}, \frac{\delta f}{\delta x_{10}} \right) = (-155048,653, -237734,200, 186190,928) = \sqrt{(-155048,653)^2 + (-237734,200)^2 + (186190,928)^2} = 339447,635$$

Calculando  $\eta$ : onde um número sorteado pseudo-aleatoriamente  $\in [0,1]$  é dividido por  $\alpha^2$ .

$$\eta_2 = \frac{\text{número sorteado}}{\alpha} = \frac{0,00051}{339447,635} = 1,50244E - 09$$

$$\eta_7 = \frac{\text{número sorteado}}{\alpha} = \frac{0,00071}{339447,635} = 2,09163E - 09$$

$$\eta_{10} = \frac{\text{número sorteado}}{\alpha} = \frac{0,00031}{339447,635} = 9,13248E - 10$$

finalizando, tem-se os novos valores para  $x_2$ ,  $x_7$  e  $x_{10}$ .

$$x_i \text{ novo} = x_i - \eta_i \frac{\partial f}{\partial x_i}$$

$$x_2 \text{ novo} = 0,9665 - \eta_2 \frac{\partial f}{\partial x_2} = 0,9667$$

$$x_7 \text{ novo} = 0,7354 - \eta_7 \frac{\partial f}{\partial x_7} = 0,7359$$

$$x_{10} \text{ novo} = 0,9118 - \eta_{10} \frac{\partial f}{\partial x_{10}} = 0,9116$$

Portanto, o  $\text{custo}(i_{\text{novo}}) = 550037,1841$ , ou seja, a função foi minimizada, e as novas características do indivíduo  $i$  são apresentadas na Figura 11.

Indivíduo antes de aplicar mutação gradiente

0,9929	0,9665	0,9251	0,8726	0,8147	0,7612	0,7354	0,7573	0,8345	0,9118	0,9746	1,0000
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Depois da mutação gradiente

0,9929	0,9667	0,9251	0,8726	0,8147	0,7612	0,7359	0,7573	0,8345	0,9116	0,9746	1,0000
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Figura 11. Mutação gradiente.



Adotou-se como início de todos os períodos o mês de maio, por ser o início do período seco na bacia hidrográfica em questão. É de se esperar que, em uma operação otimizada, os reservatórios estejam cheios no final do período chuvoso, justamente para superar o período seco que se inicia. Consequentemente, como final, adotou-se o mês de abril, final do período de afluições mais elevadas. As afluições naturais aos reservatórios em toda pesquisa foram retiradas do arquivo de vazões do Sistema Brasileiro.

Este estudo teve como objetivo verificar o comportamento na prática do método desenvolvido, apresentando uma aplicação do algoritmo proposto em um sistema teste composto por 35 usinas hidroelétricas, abrangendo grande parte do Sistema Sudeste Brasileiro, sendo utilizadas usinas a reservatório e a fio d'água, conforme apresentado na Figura 12. Com uma potência instalada total de 36271 MW.

Um subsistema incluindo somente as usinas a reservatório foi testado e apresentado em Alencar et al. (2009), Alencar & Leite (2010a) e Alencar & Leite (2010b). Neste capítulo optou-se por apresentar com as usinas a fio d'água incluídas no subsistema, com objetivo de se verificar as interferências que poderiam surgir para o algoritmo proposto, principalmente quanto ao crescimento do número de usinas, o que atestou a viabilidade do AGs para operar com várias usinas.

A apresentação deste estudo, ao mesmo tempo em que mostra didaticamente a aplicação dos AGs no POSHP, comprova a aplicabilidade e eficiência do algoritmo em sistemas grandes e complexos, com muitas usinas de grande porte, interligadas em paralelo e em cascata, com dados reais e características típicas das usinas brasileiras.

Observa-se nas Figuras 13 a 18 que os AGs conseguiram definir um cronograma de operação que se repetiu no segundo ano, devido aos dados sazonais da vazão afluente ao longo do intervalo de planejamento. Diferentemente da ferramenta que utiliza programação não linear – PNL (Cicogna, 2003) que em alguns casos não repetiu a mesma trajetória no segundo ano.

Através da Figura 19 pode-se avaliar a geração hidráulica e a complementação térmica definidas pelas duas técnicas, quando se adota afluições iguais à MLT. Vale destacar que, a complementação térmica é bastante uniforme ao longo do período de planejamento, indicando um custo de operação menor, pois este custo é dado por uma função com crescimento quadrático, conforme demonstrado na formulação matemática do problema. Portanto, quanto mais uniforme menor será o custo operativo ao longo do intervalo de planejamento.

O Estudo realizado neste capítulo comprovou, que as técnicas de Inteligência Artificial, baseada em Algoritmos Genéticos, é eficiente na resolução de problemas de Planejamento da Operação de Sistemas Hidrotérmicos de Potência. De acordo com as trajetórias de volumes e gerações apresentadas pode-se concluir que nesta situação de complexidade da cascata, os

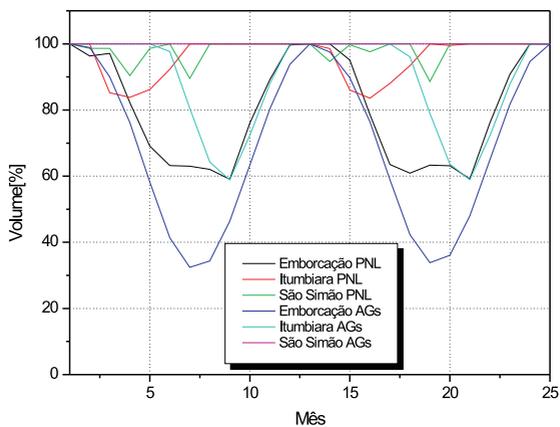


Figura 13. Estudo V – Trajetória ótima de volume da cascata do rio Paranaíba.

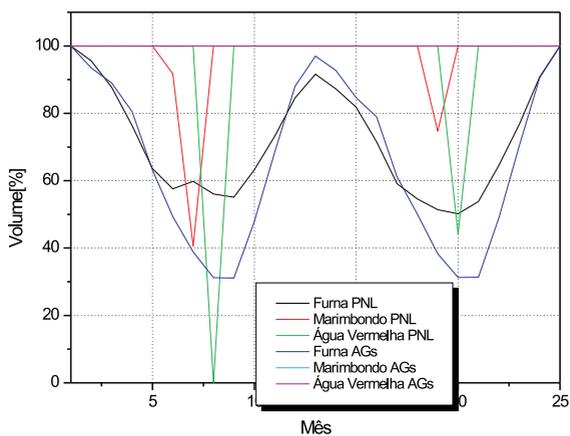


Figura 14. Estudo V – Trajetória ótima de volume da cascata do rio Grande.

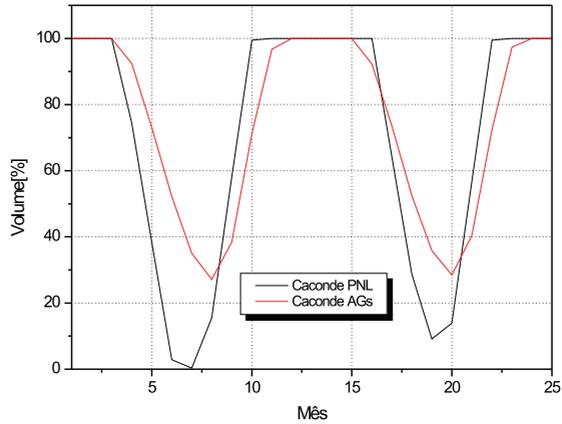


Figura 15. Estudo V – Trajetória ótima de volume da cascata do rio Pardo.

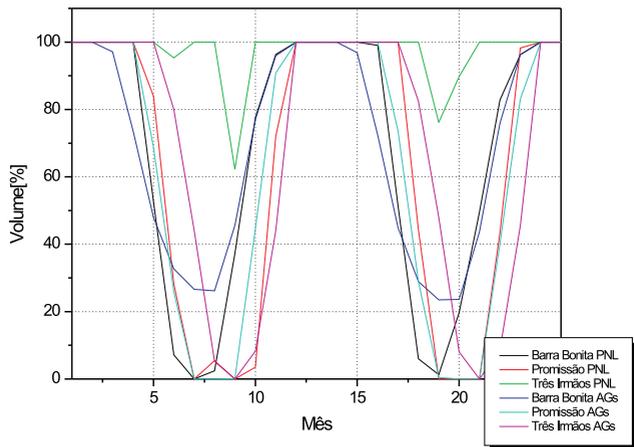


Figura 16. Estudo V – Trajetória ótima de volume da cascata do rio Tietê.

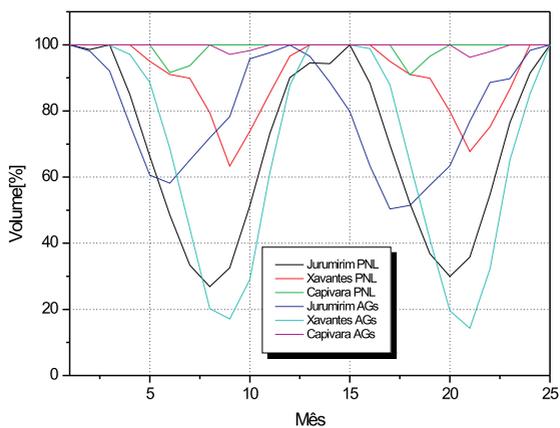


Figura 17. Estudo V – Trajetória ótima de volume da cascata do rio Parapanema.

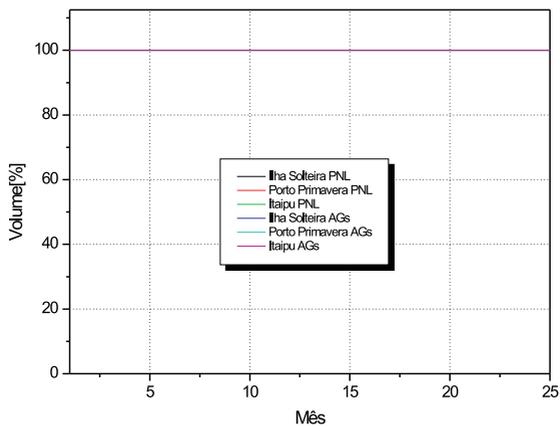


Figura 18. Estudo V – Trajetória ótima de volume da cascata do rio Paraná.

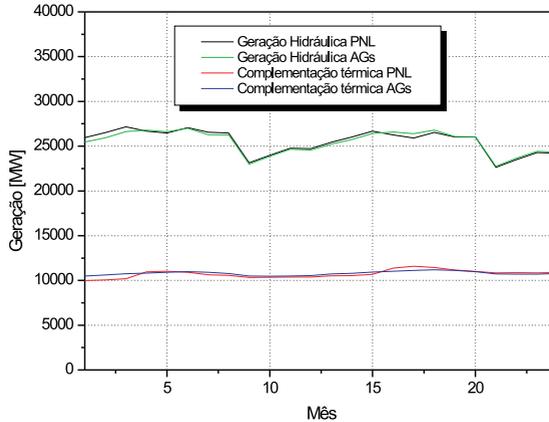


Figura 19. Estudo V – Geração hidráulica e complementação térmica das usinas do sistema sudeste.

AGs conseguiram apresentar uma boa solução. Portanto, acredita-se fortemente que este algoritmo proposto poderá auxiliar nos estudos e ações do planejamento da operação resolvendo ou mesmo auxiliando as análises dos gerenciadores do setor elétrico Brasileiro.

## 6. Conclusão

Conforme apresentado neste documento, o referido trabalho mostra a aplicabilidade e potencialidade dos Algoritmos Genéticos no Planejamento da Operação de Sistemas Hidrotérmicos de Potência.

Com relação à técnica adotada, vários testes no que diz respeito a definição do tamanho de uma população eficiente, busca de uma solução inicial mais adequada ao problema e a ponderação dos operadores genéticos utilizados foram considerados nesta fase de desenvolvimento da ferramenta computacional. Tais análises buscam uma maior segurança na avaliação quanto aos tipos de operadores que melhor se adaptam ao problema e ao emprego de técnicas especiais, como os algoritmos híbridos, que contribuíram para um melhor desempenho da ferramenta, conforme evidenciado.

Cabe lembrar que o objetivo principal da ferramenta, investigar a aplicação da técnica citada no Planejamento Eletro-Energético, voltado principalmente para o Planejamento Energético de curto prazo, foi exaustivamente avaliado através de testes com usinas pertencentes ao Sistema Sudeste Brasileiro. Estes testes abrangeram desde usina isolada, sistemas com usinas em cascata simples, sistema com várias usinas com cascatas em paralelo, até um complexo sistema de grande porte (35 usinas interli-

gadas hidráulicamente), adotando-se distintas vazões afluentes, o que vem a mostrar a pertinência e a consistência da abordagem.

Os bons resultados obtidos na aplicação dos AGs mostraram o grande potencial desta ferramenta, que conseguiu captar as diferentes características de operação das usinas, sem a necessidade de simplificar a formulação original do problema. Observou-se que em todas as situações os AGs apresentaram um desempenho satisfatório na determinação de uma estratégia de operação que venha a atender as restrições do Planejamento da Operação de Sistemas Hidrotérmicos de Potência.

## Agradecimentos

Os autores agradecem à Fundação Estadual de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro ao projeto.

## Referências

- Alencar, T.R. & Leite, P.T., Development of an intelligent tool for planning the operation. *World Academy of Science, Engineering and Technology*, 63:519, 2010a.
- Alencar, T.R. & Leite, P.T., Development of a computational tool for application in the operation of hydrothermal power systems. In: *Proceedings of IEE Transmission and Distribution Conference and Exposition*. New Orleans, USA, p. 1–6, 2010b.
- Alencar, T.R.; Leite, P.T.; Carneiro, A.A.F.M. & de Carvalho, A.C.P.L.F., Desenvolvimento de uma ferramenta inteligente para o planejamento da operação de sistemas hidrotérmicos de potência. In: *Proceedings of the 8<sup>th</sup> Latin American Congress on Electricity Generation and Transmission*. Ubatuba, SP, 2009.
- Bazaraa, M.S.; Sherali, H. & Shetty, C., *Nonlinear Programming - Theory and Algorithms*. 2a edição. New York, USA: John Wiley & Sons, 1993. 656 p.
- Beasley, D.; Bull, D.R. & Martin, R.R., An overview of genetic algorithms: Part 2, research topics. *University Computing*, 15(4):170–181, 1993.
- Carneiro, A.A.F.M., *Contribuição ao Planejamento da Operação Energética de Sistemas Hidrotérmicos de Potência*. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, 1991.
- Carneiro, A.A.F.M.; Soares, S. & Bond, P.S., Modelling and optimization of hydrothermal generation scheduling. Anais do VII Congresso Brasileiro da Operação de Sistemas Hidrotérmicos de Potência, 2:823–829, 1988.

- Carneiro, A.A.F.M.; Soares, S. & Bond, P.S., A large scale application of an optimal deterministic hydrothermal scheduling algorithm. *IEEE Transactions on Power Systems*, 5(1):204–211, 1990.
- Carvalho, M.F.H., *Modelos de Fluxo em Redes Aplicados a Sistemas de Energia Elétrica*. Tese de doutorado, Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, SP, 1986.
- Chang, H.C. & Chen, P.H., Hydrothermal generation scheduling package: A genetic based approach. *IEEE Proceedings Generation, Transmission, and Distribution*, 145(4):451–457, 1998.
- Chen, P.H. & Chang, H.C., Genetic aided scheduling of hydraulically coupled plants in hydro-thermal coordination. *IEEE Transactions on Power Systems*, 11(2):975–981, 1996.
- Cicogna, M.A., *Modelo de Planejamento da Operação Energética de Sistemas Hidrotérmicos a Usina Individualizada Orientado por Objetos*. Dissertação de mestrado, Faculdade de Engenharia Elétrica e de Computação, Universidade de Campinas, Campinas, SP, 1999.
- Cicogna, M.A., *Sistema de Suporte a Decisão para o Planejamento e a Programação da Operação de Sistemas de Energia Elétrica*. Tese de doutorado, Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, SP, 2003.
- Cicogna, M.A. & Soares, S., Um sistema de suporte à decisão para o planejamento e a programação da operação de sistemas hidrotérmicos de potência. In: *Anais do XVII SNPTEE – Seminário Nacional de Produção e Transmissão de Energia Elétrica*. Uberlândia, MG, 2003.
- Corne, D.; Dorigo, M. & Glover, F. (Eds.), *New Ideas in Optimization*. New York, USA: McGraw-Hill, 1999. 450 p.
- EPE, , BEN - Balanço Energético - Relatório Final 2010. Dados técnicos, Empresa de Pesquisa Energética, <http://www.epe.gov.br>, 2010.
- Fortunato, L.A.M.; Neto, A.; Barreto, L.A.L. & Ferreira, C., Operation planning studies of the brazilian generation system. In: *Proceedings of IFAC Symposium on Planning and Operation of Electric Energy Systems*. Rio de Janeiro, RJ, p. 193–200, 1985.
- Gen, M. & Cheng, R., *Genetic Algorithms and Engineering Design*. Reading, EUA: Addison-Wesley, 1997. 432 p.
- Goldberg, D.E., *Genetic Algorithms in Search Optimization and Machine Learning*. Reading, EUA: Addison-Wesley, 1989. 432 p.
- Leite, P.T., *Aplicação de Técnicas de Inteligência Artificial no Planejamento da Operação de sistemas Hidrotérmicos de Potência*. Tese de doutorado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, 2003.

- Leite, P.T.; Carneiro, A.A.F.M. & Carvalho, A.C.P.L.F., Energetic operation planning using genetic algorithms. *IEEE Transactions on Power Systems*, 17(1):173–179, 2002.
- Leite, P.T.; Carneiro, A.A.F.M. & Carvalho, A.C.P.L.F., Aplicação de algoritmos genéticos na determinação da operação Ótima de sistemas hidrotérmicos de potência. *Revista Controle & Automação*, 17(1):81–88, 2006.
- Macieira, M.E.P.; Marcato, R.M. & Marcato, A.L.M., Comparação entre abordagem estocástica e determinística no planejamento da operação de médio prazo de sistemas hidrotérmicos interligados. In: *Anais do XVII SNPTEE – Seminário Nacional de Produção e Transmissão de Energia Elétrica*. Uberlândia, MG, 2003.
- MAE, , Processo de Utilização dos Modelos de Otimização na Formação do Preço do MAE. Apêndice D, Mercado Atacadista de Energia Elétrica, <http://www.mae.org.br>, 2002.
- Martinez, L., *Política de Controle Malha Fechada e Malha Aberta no Planejamento da Operação Energética de Sistemas Hidrotérmicos*. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, 2001.
- Matsumoto, M. & Nishimura, T., Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, 1998.
- Miranda, V.; Srinivasan, D. & Proenca, L.M., Evolutionary computation in power systems – example 2: Genetic algorithms in hydrothermal operation planning. *International Journal of Electrical Power & Energy Systems*, 20(2):89–98, 1998.
- Pereira, M.V., Optimal scheduling of hydrothermal systems - an overview. In: *Proceedings of IFAC Symposium on Planning and Operation of Electric Energy Systems*. Rio de Janeiro, RJ, p. 1–9, 1985.
- Pereira, M.V.; Cunha, S.H.F.; Terry, L.A. & Mosse, A., Modelos computacionais para planejamento e operação de sistemas hidrotérmicos de grande porte. *SBA: Controle de Automação*, 1(1):31–41, 1987.
- Rabêlo, R., *Componentes de Software no Planejamento da Operação Energética de Sistemas Hidrotérmicos*. Tese de doutorado, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, SP, 2010.
- Rezende, S.O. (Ed.), *Sistemas Inteligentes – Fundamentos e Aplicações*. São Paulo, SP: Manole, 2003. 525 p.
- Rosenthal, R.E., *Status of Optimization Models for the Operation of Multi-reservoir Systems with Stochastic Inflows and Nonseparable Benefits*. Research Report n. 75, Water Resources Research, The University of Tennessee, Water Resources Research Center, 1980.

- Salomon, R., Evolutionary algorithms and gradient search: Similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.
- Schlick, C., Fast alternatives to Perlin's bias and gain functions. In: Heckbert, P.S. (Ed.), *Graphics Gems IV*. San Diego, USA: Academic Press Professional, p. 401–403, 1994.
- Silva, E.L., *Formação de Preço em Mercados de Energia Elétrica*. Porto Alegre, RS: Editora Sagra Luzzatto, 2001.
- Soares, S. & Carneiro, A.A.F.M., Optimal operation of reservoirs for electric generation. *IEEE Transactions on Power Delivery*, 6(3):1101–1107, 1991.
- Tavares, H.; Soares, S. & Lyra, C., Modelling and optimization of hydrothermal generation scheduling. *IEEE Transactions on Power Apparatus and Systems*, PAS-103(8):2126–2133, 1984.
- Yang, P.C.; Yang, H.T. & Huang, C.L., Scheduling short-term hydrothermal generation using evolutionary programming techniques. *IEE Proceedings Generation, Transmission, and Distribution*, 143(4):371–376, 1996.

## Notas Biográficas

**Patricia Teixeira Leite Asano** possui graduação em Engenharia Elétrica pela Universidade Federal de Mato Grosso (1995), mestrado em Engenharia Elétrica pela Universidade de São Paulo (1999) e doutorado em Engenharia Elétrica São Carlos pela Universidade de São Paulo (2003). Atualmente é professora adjunto da Universidade Federal do ABC, atuando principalmente nos seguintes temas: algoritmos genéticos, sistemas hidrotérmicos, planejamento da operação, otimização e inteligência artificial.

**Thiago Ribeiro de Alencar** é bacharel em Ciência e Tecnologia pela Universidade Federal do ABC (2009). Atualmente é pesquisador da Universidade Federal do ABC e presidente do Capítulo de Potência (PES) da IEEE (*Institute of Electrical and Electronic Engineers*) – seção sul Brasil. Tem experiência na área de Engenharia Aeroespacial, com ênfase em Propulsão Aeroespacial. Atua principalmente nos seguintes temas: energia, inteligência artificial, algoritmos genéticos, planejamento da operação, sistemas hidrotérmicos e colônia de formigas.

**Adriano Alber de França Mendes Carneiro** possui graduação em Engenharia Elétrica pela Pontifícia Universidade Católica de São Paulo (1970), especialização Engenharia de Manutenção pelo Instituto Mauá de Tecnologia (1978), especialização em Engenharia de Sistemas Elétricos pela Universidade Federal de Itajubá (1979), mestrado em Engenharia Elétrica pela Universidade Federal de Itajubá (1980) e doutorado em Engenharia Elétrica pela Universidade Estadual de Campinas (1991). Atualmente é professor da Universidade de São Paulo, atuando principalmente nos seguintes temas: sistemas elétricos de potência, planejamento da operação, geração hidrotérmica, otimização da geração, planejamento energético e programação não-linear.

**André Carlos Ponce de Leon Ferreira de Carvalho** possui graduação em Ciência da Computação pela Universidade Federal de Pernambuco (1987), mestrado em Ciências da Computação pela Universidade Federal de Pernambuco (1990) doutorado em Electronic Engineering pela University of Kent at Canterbury (1994) e pós-doutorado pela Universidade do Porto (2006). Atualmente é professor titular da Universidade de São Paulo, atuando principalmente nos seguintes temas: redes neurais, aprendizado de máquina, bioinformática, algoritmos genéticos, mineração de dados e sistemas híbridos.

**Sadao Massago** possui graduação em Licenciatura em Matemática pela Universidade Estadual de Maringá (1992), mestrado em Ciências Matemáticas pelo Instituto de Ciências Matemáticas de São Carlos (1995) e doutorado em Ciências Matemáticas pelo Instituto de Ciências Matemáticas e de Computação (2000). Atualmente é o professor adjunto da Universidade Federal de São Carlos, atuando principalmente nos seguintes temas: topologia algébrica, teoria de nós, teoria da fibração e matemática aplicada na área de pesquisa operacional.

# Algoritmos Genéticos Aplicados ao Planejamento da Distribuição de Energia Elétrica em Curitiba e Região Metropolitana

Arleide Cristina Alves,\* Maria Teresinha Arns Steiner,  
Nelson Haj Mussi e Giovani Zanelatto

---

**Resumo:** Para o contínuo crescimento dos municípios torna-se necessário a expansão otimizada das redes de distribuição de energia elétrica. A localização e o balanceamento otimizados de SE's são questões de grande importância para um planejamento adequado. Neste capítulo é abordado o problema de localização de instalações (*facilities location*), sendo que a região de estudo é o município de Curitiba e adjacências. Foram aplicados os Algoritmos Genéticos, sendo que a melhor solução obtida foi refinada através do algoritmo Teitz e Bart. O algoritmo de Gillett e Johnson é usado para a definição da região de atendimento de cada SE. A metodologia proposta serve como auxílio ao engenheiro tomador de decisões.

**Palavras-chave:** Planejamento de distribuição, Localização de instalações, Algoritmos genéticos.

**Abstract:** *The continuous development of the cities requires to an optimized expansion of energy distribution networks. The location and optimized load distribution of substations are subjects of great importance to an adequate planning. This chapter focuses the facilities location problem, and the area of study is the city of Curitiba and surroundings. Genetic algorithms were used and the best solution found was later refined through the Teitz and Bart algorithm. The Gillett and Johnson algorithm is used to define the service area for each substation. The proposed methodology serves as a tool to the decision maker engineer.*

**Keywords:** *Distribution planning, Facilities location, Genetic algorithms.*

---

\* Autor para contato: aalves@up.com.br

## 1. Introdução

O crescimento de um país afeta diretamente o crescimento e o aumento do mercado consumidor de energia elétrica. Por outro lado, dificuldades econômicas podem provocar restrições de verbas para os investimentos. Tais fatores geram a necessidade de se elaborar estudos mais complexos, para o suprimento de energia elétrica a uma determinada área, que podem envolver uma massa de dados com muitas informações, exigindo análise detalhada de um número maior de alternativas para os mais diversos problemas. Além disto, existe a necessidade de se ter respostas rápidas para o consumidor que tem se tornado cada vez mais exigente, tanto com relação ao tempo em que fica privado do fornecimento de energia elétrica, quanto em relação ao número de vezes que isto ocorre.

Todos estes fatores, por si só, já apontam para a necessidade de ajustes das concessionárias de energia elétrica com relação aos serviços prestados. Estes ajustes são comumente tratados através do planejamento de distribuição de energia elétrica em redes, que é o processo de estudos e análises que visa garantir a eficiência de uma rede elétrica. Um planejamento de distribuição satisfatório inclui tamanho, locação, rotas, interconexões e previsão do momento (“quando”) e locais adequados (“onde”) para futuras linhas de transmissão, subestações, alimentadores e equipamentos relacionados à distribuição de energia elétrica.

No que diz respeito à localização e dimensionamento de subestações, torna-se evidente a necessidade de adotar estudos de técnicas que permitam a otimização do processo, que pode ser obtida com do uso de modelos matemáticos e algoritmos, assim como proposto neste trabalho.

Tal otimização faz uso dos mapas de carga que são representações gráficas da distribuição de carga em uma localidade ou região, convenientemente subdividida em áreas elementares (quadrículas), que são as menores unidades de área a serem consideradas na distribuição geográfica de carga. Sua definição depende do tipo de estudo a ser realizado. Por exemplo, para o planejamento de subestações em Curitiba, Região Metropolitana (RMC) e adjacências, são utilizadas quadrículas de área de  $1 \text{ km}^2$ . Os dados utilizados neste estudo foram fornecidos pela concessionária de energia elétrica, através de uma planilha eletrônica, em ambiente Microsoft Excel, contendo as suas coordenadas no sistema Universal Transversa de Mercator (UTM), as demandas de cargas em cada uma delas, bem como a localização e a capacidade de atendimento das subestações existentes.

Problemas de localização de instalações (ou medianas) aparecem frequentemente na prática em uma variedade de aplicações como, por exemplo, a localização do centro de comutação numa rede telefônica, armazéns de suprimentos numa rede de distribuição e a localização de centros de triagens de cartas, dentre outras.

São muitos os trabalhos apresentados na literatura que tratam de problemas reais, cuja solução recai ou faz uso dos problemas de localização de instalações. Dentre estes trabalhos pode-se citar o de [Corrêa \(2000\)](#) onde são aplicados os algoritmos metaheurísticos algoritmos genéticos e busca tabu, comparativamente para o problema das  $p$ -medianas capacitado. Neste problema o objetivo principal era determinar cerca de 20 instalações, dentre cerca de 40 possibilidades, que melhor acomodassem, em termos de distância, os cerca de 50 mil candidatos ao vestibular da Universidade Federal do Paraná (UFPR).

Em [Detofeno & Steiner \(2010\)](#) é apresentada uma metodologia para a obtenção de rotas otimizadas para a coleta de resíduos urbanos. Para tanto, foi utilizada primeiramente a heurística de Teitz e Bart para a obtenção de  $p$ -medianas. A partir da definição destas, foram definidos grupos (*clusters*) de pontos de demanda através da designação destes às medianas, fazendo uso do algoritmo de Gillett e Johnson. Finalmente, a partir da definição dos grupos de pontos, foi utilizado o algoritmo do Carteiro Chinês, obtendo-se o roteamento (sequenciamento dos pontos de demanda a serem atendidos) em cada um dos grupos de atendimento.

Em [Scarpin et al. \(2008\)](#) é apresentada uma proposta para a otimização no serviço de saúde no estado do Paraná com relação ao fluxo de pacientes dentro do estado e a regionalização (divisão) do estado, obtendo novas configurações hierárquicas para o mesmo. Quanto à regionalização, a proposta consiste em dividir o estado em regiões menores, formadas por várias cidades, vinculadas a uma cidade-sede. Para isto, foi feito uso do algoritmo *branch-and-price*, que utiliza o algoritmo de geração de colunas em cada nó de uma árvore *branch-and-bound*. Este algoritmo se mostrou eficiente melhorando a divisão hierárquica do estado, levando em consideração o “número de habitantes” e o “número de procedimentos médicos” de cada município do estado. Os resultados obtidos têm atendido as expectativas da Secretaria de Saúde do estado do Paraná (SESA-PR).

Em [Smiderle et al. \(2004\)](#) é proposta uma metodologia para a obtenção de uma solução otimizada dos leituristas dos medidores das contas de água, da Companhia de Saneamento do Paraná (SANEPAR) no município de Pato Branco. Inicialmente foi aplicado um algoritmo genético para se determinar as  $p$ -medianas ( $p = 12 =$  número de leituristas), cuja resposta pode ser melhorada com a heurística clássica de Teitz e Bart. A partir daí, foram definidos os grupos (*clusters*) de pontos a serem atendidos por cada funcionário através do algoritmo de designação adaptado de Gillett e Johnson. A partir da definição dos *clusters*, foi realizado o roteamento para ter-se a sequência de pontos a serem visitados, utilizando o modelo matemático do problema do Carteiro Chinês.

O objetivo deste trabalho é apresentar técnicas que visam otimizar a localização e o balanceamento de subestações de distribuição de energia elétrica na cidade de Curitiba, RMC e adjacências, para o horizonte de

um ano, usando técnicas da área da Pesquisa Operacional. O problema apresentado se enquadra como um problema de localização de facilidades (*facilities location*). Para a localização “ótima” de uma nova subestação foram utilizados os Algoritmos Genéticos, cuja melhor solução obtida ao final do procedimento foi melhorada através do algoritmo de Teitz e Bart. O Algoritmo de Gillett e Johnson é usado para definir a região de atendimento, isto é, fazer um “balanceamento” de cada subestação pertencente à área de estudos, levando-se em conta as coordenadas e as capacidades de atendimento reais de cada uma. O algoritmo de Gillett Johnson cumpre este objetivo por agrupar quadrículas (com suas respectivas demandas de carga), à subestação que as atenda tornando mínimo o percurso das cargas, ou seja, minimizando o momento elétrico (produto da carga pela distância por ela percorrida).

A metodologia proposta se mostrou bastante adequada ao problema, lembrando que um atendimento satisfatório em relação à demanda de energia elétrica para uma determinada região é, na verdade, um problema de transporte de cargas através de uma rede.

A seção 2 apresenta brevemente o contexto do problema de localização de instalações, assim como os algoritmos matemáticos da área de Pesquisa Operacional utilizados à solução do problema real aqui abordado: os algoritmos genéticos (AG's), o algoritmo de Teitz e Bart (TB) e o algoritmo de Gillett e Johnson (GJ). Na seção 3 é apresentada a metodologia para a solução do problema em si, que consta do levantamento dos dados e a forma de aplicação dos algoritmos mencionados. Na seção 4 são apresentados os resultados de algumas das numerosas simulações realizadas e, finalmente, na seção 5, as conclusões do trabalho.

## 2. O Problema de Localização de Instalações

A localização “ótima” de uma possível nova subestação deve levar em conta fatores tais como:

- custo e facilidade de conexão com os alimentadores de transmissão e distribuição já existentes;
- limites de tensão e corrente, que podem afetar o número e os custos dos alimentadores necessários para a alimentação de uma determinada área;
- possibilidade de transferência de carga de uma subestação para outras em condições de emergência, ou quando do aumento na demanda de carga;
- custo e disponibilidade de terrenos próximos ao local desejado;
- restrições devido a eventuais leis de zoneamento.

Tais fatores implicam em investimentos na rede elétrica. Para que a nova subestação assuma parte da demanda da rede, é necessário planejamento adequado dos investimentos envolvidos nesta empresa concomitantemente à construção da mesma. Este fato evidencia a necessidade de subentender-se também a restauração de uma rede com planejamento, levando-se em conta que a localização de uma subestação é fator importante no que diz respeito à manutenção de serviços adequados e continuidade do fornecimento de energia aos consumidores (Gruppelli Jr. et al., 2002).

No caso destes problemas de localização serem de pequeno porte, a solução pode ser encontrada de forma exata, através de um modelo de programação inteira ou enumeração exaustiva (ou busca em árvore). Este último método foi usado por Revelle & Swain (1970) para encontrar as 3-medianas de um grafo com 10 vértices, dando início ao estudo deste tipo de problema. Existem, também, os métodos aproximados (ou heurísticos), usados por vários pesquisadores como, por exemplo, Teitz & Bart (1968), os quais se aplicam a problemas de médio e grande porte. Para estes casos pode-se utilizar, também, técnicas metaheurísticas, dentre as quais destacam-se os algoritmos genéticos (AG's) (Corrêa et al., 2004).

Considerando-se todos os vértices de um grafo dado como potenciais medianas, o problema de  $p$ -medianas pode ser definido como segue: seja  $G(V, A)$  um grafo não direcionado onde  $V$  é o conjunto de vértices e  $A$  o conjunto de arestas. Deve-se encontrar um conjunto de vértices  $V_p \subset V$  (conjunto de medianas) com cardinalidade  $p$ , tal que a soma das distâncias de cada um dos vértices restantes em  $\{V - V_p\}$ , considerando-se as demandas de cada vértice até seu vértice mais próximo em  $V_p$  seja a mínima possível.

Uma formulação do problema de  $p$ -medianas como um modelo matemático de programação inteira binária, desenvolvida por Revelle & Swain (1970), é apresentada a seguir. Este modelo permite que cada vértice de um grafo seja considerado, ao mesmo tempo, como demanda e instalação (potencial mediana), embora em muitos casos demandas e instalações pertençam a conjuntos disjuntos.

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^n a_i d_{ij} x_{ij} \quad (1)$$

sujeito a:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (2)$$

$$x_{i,j} \leq y_j, \quad i, j = 1, 2, \dots, n \quad (3)$$

$$\sum_{j=1}^n y_j = p \quad (4)$$

$$x_{ij}, y_j \in \{0, 1\}, \quad i, j = 1, 2, \dots, n \quad (5)$$

onde:

$n$  = número total de vértices do grafo;

$a_i$  = demanda do vértice  $i$ ;

$d_{ij}$  = distância do vértice  $i$  ao vértice  $j$ ;

$p$  = número de instalações utilizadas como medianas;

$$x_{ij} = \begin{cases} 1, & \text{se o vértice } j \text{ for designado para a instalação } i; \\ 0, & \text{caso contrário} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{se o vértice } j \text{ for uma instalação utilizada como mediana;} \\ 0, & \text{caso contrário} \end{cases}$$

A função objetivo na Equação 1 minimiza a soma das distâncias ponderadas dos vértices de demanda até o conjunto de medianas. O conjunto de restrições em (2) garante que todos os vértices de demanda serão designados para exatamente uma única mediana. O conjunto de restrições em (3) proíbe que um vértice de demanda seja designado para uma instalação que não esteja selecionada como mediana. O número total de vértices medianas (instalações selecionadas) é definido pela restrição em (4) como sendo igual a  $p$ . Por fim, a restrição em (5) garante que os valores das variáveis de decisão  $x_{ij}$  e  $y_j$  sejam binários (assumindo os valores “0” ou “1”).

A utilização deste modelo matemático fica restrita a problemas de pequeno porte pelo fato das variáveis serem binárias. Além disso, há a falta de flexibilidade do mesmo para se considerar no problema, por exemplo, as 30 subestações já existentes, conforme será visto mais adiante. Por este motivo, fez-se uso aqui da metaheurística AG seguida da heurística TB para a definição da localização das medianas (subestações) e, em seguida, da heurística de GJ para o balanceamento (distribuição das cargas) às medianas. Todos estes três algoritmos estão descritos a seguir.

## 2.1 Breve descrição dos algoritmos genéticos (AG's)

Os AG's são sistemas inspirados na natureza, simulando os processos naturais, utilizados para a resolução de problemas reais. São métodos ge-

neralizados de busca e otimização que simulam os processos naturais de evolução, aplicando a idéia Darwiniana de seleção. Os AG's diferem dos métodos exatos de busca e otimização, principalmente em três aspectos: trabalham a partir de uma população de soluções e não com uma única solução, utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar, utilizam regras de transição probabilísticas e não determinísticas.

Para serem utilizados, os AG's exigem que inicialmente seja gerada uma população formada por um conjunto aleatório de indivíduos que podem ser vistos como possíveis soluções do problema. A população inicial, assim como as demais que são geradas durante o processo evolutivo são avaliadas: para cada indivíduo é atribuído um valor (*fitness*), ou índice, refletindo a sua habilidade de adaptação a determinado ambiente. A probabilidade dos indivíduos mais adaptados serem mantidos é maior do que a dos indivíduos menos adaptados (Darwinismo).

Os indivíduos mantidos pela seleção podem sofrer modificações em suas características fundamentais através de mutações e cruzamentos (*crossover* ou recombinação genética), gerando descendentes para a próxima geração. Este processo é repetido até que algum critério de parada (número de iterações, convergência das soluções/genes, dentre outros) seja atendido.

Alguns critérios básicos devem ser levados em conta quando da aplicação de AG's à solução de um problema. Tais critérios abrangem aspectos como codificação, função de avaliação (*fitness*), operadores genéticos (*crossover*, mutação), taxas probabilísticas de realização dos operadores genéticos, critérios de seleção, reprodução, critério de parada e convergência prematura, dentre outros. A Seção 3 deste trabalho mostra os critérios utilizados à simulação numérica do algoritmo para o problema abordado. Os demais critérios são amplamente discutidos em Alves (2002).

O operador genético *crossover* toma dois indivíduos previamente selecionados através de algum critério e realiza o seu "cruzamento", ou seja, a troca de seus materiais genéticos. Este cruzamento pode ocorrer de diversas formas, sendo que três delas são apresentadas nas figuras 1, 2 e 3, a seguir. Vale salientar que o operador *crossover* busca solução a partir do conhecimento dos indivíduos já existentes (*exploitation*). Novos indivíduos podem ser gerados pelo cruzamento de indivíduos diferentes.

Por outro lado, o operador de mutação busca uma solução em regiões factíveis ainda não avaliadas (*exploration*). A figura 4 ilustra um dos diversos tipos de operador genético de mutação e apresenta apenas uma das formas de se realizar a mutação. A mutação é responsável pela geração de novos pontos no espaço de soluções.

Podem ser destacadas algumas variações no operador de mutação (Sampaio, 1999). Por exemplo, a mutação por inversão é caracterizada pela retirada de uma parte do cromossomo e a sua recolocação no indivíduo na ordem inversa em que foi retirada. Já na mutação por translocação,

Figura 1. *Crossover* de uma partição.Figura 2. *Crossover* de duas partições.Figura 3. *Crossover* de três partições.

Gene sorteado aleatoriamente para mutação

Figura 4. *Crossover* de três partições.

parte do cromossomo é retirada e colocada em outra posição, guardando a ordem com a qual foi retirada. A hipermutação, mostrada por [Corrêa et al. \(2004\)](#), é uma operação onde se realiza a mutação em todos os genes possíveis, avaliando-se em qual deles é melhor que se estabeleça a inserção do material genético.

Para problemas de otimização, o ideal seria que o algoritmo finalizasse assim que a solução ótima (ponto ótimo) fosse descoberta. No caso de funções multimodais, um ponto ótimo pode ser o suficiente, mas pode haver situações onde todos ou o maior número possível sejam desejados. O uso de um algoritmo evolutivo não permite afirmar se um dado ponto ótimo corresponde a um ótimo global ou local.

Como consequência, normalmente utiliza-se como critério de parada o número máximo de gerações (iterações) ou tempo limite de processamento. Outro critério plausível, segundo [Corrêa et al. \(2004\)](#), é finalizar a execução do algoritmo usando a idéia de estagnação, ou seja, quando não se observa alterações dos desvios padrão dos valores de adaptação entre duas iterações consecutivas. Assim, se não houver a melhoria da população depois de várias gerações consecutivas, o algoritmo encerra o processamento.

## 2.2 Descrição do algoritmo de Teitz e Bart (TB) para o problema das $p$ -medianas

Uma das heurísticas mais conhecidas para solução do problema das  $p$ -medianas é a desenvolvida por [Teitz & Bart \(1968\)](#), conhecida como al-

goritmo das  $p$ -medianas de Teitz e Bart. Esta heurística é baseada na substituição de vértices e seu objetivo é, a partir de uma solução inicial, melhorar o valor da função objetivo a cada iteração.

Segundo [Corrêa et al. \(2004\)](#), esta heurística produz boas soluções, principalmente quando aplicada várias vezes ao mesmo problema com diferentes soluções iniciais. Considerando-se todos os vértices de um grafo dado como potenciais medianas, o algoritmo de Teitz e Bart para o problema das  $p$ -medianas pode ser explicado como segue.

Seja  $V$  o conjunto de todos os vértices do grafo e  $S$  o conjunto das medianas ( $S$  está contido em  $V$ ), testa-se se qualquer vértice  $\nu_i$ , com  $\nu_i \in \{V - S\}$  pode substituir um vértice  $\nu_j$ , com  $\nu_j$  pertencente a  $S$  e produzir um novo conjunto  $S'$ , onde  $S' = S \cup \{\nu_i\} - \{\nu_j\}$  para o qual temos o número de transmissão<sup>1</sup>  $NT(S') < NT(S)$ .

Se isto for possível, é feita a substituição de  $\nu_j$  por  $\nu_i$ , e  $S'$  será uma nova aproximação para o conjunto  $\nu_p$ . O processo continua até que se obtenha um conjunto  $S_{\text{médio}}$ , onde nenhuma substituição de vértice de  $S_{\text{médio}}$  por outro em  $\{V - S_{\text{médio}}\}$  produz um número de transmissão menor.

O algoritmo de Teitz e Bart é apresentado através dos passos a seguir.

1. selecione aleatoriamente um conjunto  $S$ , com a cardinalidade igual a  $p$  (ou seja,  $|S| = p$ ) para formar uma aproximação inicial para as  $p$ -medianas;
2. rotule todos os vértices pertencentes a  $\{V - S\}$  como “não analisados”;
3. enquanto existirem vértices “não analisados” em  $\{V - S\}$ , faça:
  - (a) selecione um vértice  $\nu_i$  pertencente a  $\{V - S\}$ , “não analisado”, e calcule a redução  $A$  do número de transmissão, para todo  $\nu_j$  pertencente a  $S$ :  $A_{ij} = NT(S) - NT(S \cup \{\nu_i\} - \{\nu_j\})$
  - (b) faça  $A_{ij_0} = \text{Max}[A_{ij}]$ ;
  - (c) se  $A_{ij_0} > 0$  faça  $S = S \cup \{\nu_i\} - \{\nu_{j_0}\}$  e rotule  $\nu_{j_0}$  como “analisado”;
  - (d) se  $A_{ij_0} \leq 0$ , rotule  $\nu_i$  como “analisado”.
4. caso durante a execução do passo 3 ocorram modificações no conjunto  $S$ , volte ao passo 2. Caso contrário, pare, e apresente o conjunto  $S$  como uma aproximação para a solução do problema de  $p$ -medianas.

---

<sup>1</sup> Número de Transmissão: é a soma das menores distâncias existentes entre o vértice  $\nu_j$  e todos os outros vértices do grafo.

### 2.3 O algoritmo de Gillett e Johnson

O algoritmo proposto por Gillett e Johnson – GJ (Bodin et al., 1983) faz a designação dos pontos de demanda (cargas) às medianas (neste caso, as subestações – SE's), cujas localizações já foram definidas através dos procedimentos descritos nas seções 2.1 e 2.2. Inicialmente, todas as medianas encontram-se sem designação.

Para cada carga (demanda)  $i$ , seja  $Q_1(i)$  a distância de  $i$  até o posto de atendimento (subestação) mais próximo a  $i$ , e  $Q_2(i)$ , a distância de  $i$  até o segundo posto de atendimento (subestação) mais próximo a  $i$ . Para cada demanda  $i$ , a razão:  $r(i) = \frac{|Q_1(i)|}{|Q_2(i)|}$  é calculada e todas as demandas são colocadas numa “lista de designação” em ordem crescente pelos valores de  $r(i)$ . A designação começa pelos primeiros elementos da lista (demandas com menor razão  $r(i)$ ), e é feita obedecendo-se a capacidade dos postos de atendimento.

Durante a designação, sempre que uma demanda é designada para um posto de atendimento com a capacidade esgotada (evidentemente sem sucesso), a razão  $r(i)$  é recalculada para todas as demandas que ainda não tenham sido designadas, considerando-se apenas os postos de atendimento cujas capacidades ainda não estejam esgotadas. Estas demandas são novamente colocadas na “lista de designação” em ordem crescente pelos valores de  $r(i)$  e a designação continua até que todas as demandas sejam designadas para algum posto de atendimento.

Através do cálculo da razão  $r(i)$  procura-se conhecer a “urgência” de se fazer a designação de uma demanda em relação às demais, pois quanto menor for o valor de  $r(i)$ , maior é a urgência já que o segundo posto de atendimento mais próximo ficaria em posição bem pior para a demanda a ser atendida em relação ao primeiro. A seguir é apresentado o algoritmo GJ, passo a passo.

1. Para todo ponto de demanda  $i$  não designado, faça:
  - (a) encontre  $Q_1(i)$  e  $Q_2(i)$ , respectivamente o primeiro e o segundo postos de atendimento mais próximos do ponto do ponto  $i$  cujas capacidades não estejam esgotadas.
  - (b) calcule a razão:  $r(i) = \frac{|Q_1(i)|}{|Q_2(i)|}$ .
  - (c) coloque o ponto de demanda  $i$  na “lista de designação” pela ordem crescente dos valores de  $r(i)$ .
  - (d) iniciando pelo topo da “lista de designação” designe o ponto de demanda  $i$  para o posto de atendimento mais próximo.
2. Enquanto existirem pontos de demanda sem designação, faça:
  - (a) designe o ponto de demanda atual da “lista de designação” para o posto de atendimento mais próximo.

- (b) Diminua a demanda do ponto de demanda  $i$  da capacidade do posto de atendimento para a qual a demanda  $i$  foi designada.
- (c) Se a capacidade do posto de atendimento que recebeu a demanda atual ficar esgotada, ou não puder atender a próxima demanda, então volte ao Passo 1 e recalcule a “lista de designação”, desconsiderando os postos de atendimento com capacidade esgotadas. Caso contrário, continue.

Na Seção 3, a seguir, é apresentada a metodologia para solução do problema aqui abordado.

### 3. Metodologia para a solução do problema

A metodologia é composta das seguintes etapas: levantamento dos dados; aplicação do AG para determinar a localização (quase) ótima das novas subestações; aplicação do algoritmo de TB para o refinamento da solução fornecida pelo AG e, finalmente a aplicação do algoritmo de GJ para a definição dos grupos de quadriculas que deverão ser atendidos por cada uma das subestações.

#### 3.1 Levantamento dos dados

Os dados levantados para o desenvolvimento do trabalho aqui apresentado foram concedidos pelo setor de distribuição de energia da concessionária, através de planilha Microsoft Excel<sup>®</sup>. A região estudada, ou seja, o município de Curitiba, RMC e adjacências, por ocasião do estudo, estava constituída por 2013 quadriculas, cada uma das quais com cerca de 1 km<sup>2</sup>.

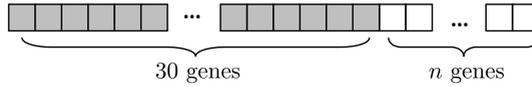
Para cada quadricula obteve-se: o seu número, a sua localização e a sua demanda (carga necessária para atender todas as cargas contidas na quadricula, em KVA). Para cada subestação obteve-se: a sua localização e o valor de sua capacidade em KVA.

#### 3.2 Aplicação do algoritmo genético – GENE

Para solução do problema da localização de subestações, a codificação escolhida para este problema é tal que cada cromossomo ficou definido como um vetor de  $(30+n)$  genes, cujos alelos são o número das quadriculas (como consequência, sabe-se a posição de cada SE), sendo que o número “30” é o número de subestações já existentes na região estudada e, portanto, fixas e  $n$  representa a quantidade de possíveis novas subestações desejadas ( $n$ , portanto, é um dado a ser fornecido pelo engenheiro usuário do programa).

Nestes cromossomos são aplicados os operadores genéticos, utilizando apenas os  $n$  alelos que representam as novas possíveis subestações. Depois disto, a melhor solução é vinculada ao cromossomo inicial que deve conter os alelos das 30 subestações fixas e aplica-se, então, o algoritmo de TB para

o refinamento da solução. A figura 5 exemplifica os cromossomos utilizados no programa.



Legenda:

- Genes correspondentes às subestações  $j$  existentes, ou seja, fixas (30);
- Genes correspondentes às subestações que serão criadas ( $n$ ).

Figura 5. Representação de um cromossomo para o problema das SE's.

Um segundo parâmetro a ser considerado é o tamanho da população. Segundo [Reeves \(2000\)](#), populações pequenas podem provocar um sério risco de não “cobrir” o espaço de busca, enquanto que populações grandes podem exigir um esforço computacional excessivo para a resolução do problema. Além disto, pode-se citar [Cunha & Pinto \(2001\)](#) que afirmam que não existem parâmetros ótimos para os operadores genéticos. Os problemas cuja solução é encontrada através de algoritmos genéticos utilizam, de uma forma geral, de 50 a 200 cromossomos para população inicial ([Tanomaru, 1995](#)). Neste trabalho, o número de cromossomos usados nas simulações obedece a este critério. No programa computacional utilizado (MATLAB), a função GENE (que executa o algoritmo genético) permite que o usuário decida quantos cromossomos terá a população inicial.

Depois de se gerar aleatoriamente a população inicial, ordenou-se os cromossomos pelo seu *fitness* que, neste caso, é o momento elétrico (medido em KVA/km), usado para avaliar o investimento em redes primárias de distribuição, como mostrado em [Savulescu \(1981\)](#). A partir disto, deve-se selecionar indivíduos (cromossomos) para recombinação.

A seleção de cromossomos pode ser aleatória ou ordenada. Porém, os melhores resultados foram obtidos utilizando-se a seleção ordenada, na qual se adota o mesmo critério proposto por [Mayerle \(1994\)](#), mostrada na Equação 6 a seguir, que segundo [Corrêa et al. \(2004\)](#) privilegia a escolha de indivíduos com melhor *fitness*:

$$Select(R) = \left\{ r_j \in R/j = p - \left\lfloor \frac{-1\sqrt{1 + 4 \cdot rnd \cdot (p^2 + p)}}{2} \right\rfloor \right\} \quad (6)$$

onde  $R$  é uma lista  $R = (R_1, R_2, R_3, \dots, R_p)$ , com  $p$  cromossomos colocados em ordem crescente pelo valor de *fitness*;  $rnd \in [0, 1)$  é um número aleatório uniformemente distribuído e o símbolo  $\lfloor b \rfloor$  significa o maior inteiro menor que  $b$ . A função  $Select(R)$  retorna um número aleatório igual à posição na lista  $R$  do cromossomo que será selecionado. Apesar de ser aleatório, o

número retornado tem maior tendência de selecionar os primeiros elementos da lista  $R$  (que são os melhores indivíduos da lista).

Realizada a seleção de dois cromossomos através da função  $Select(R)$ , inicia-se a utilização dos operadores genéticos. O primeiro deles é o *crossover*, que para esse trabalho, segue os passos:

1. Selecionados os dois cromossomos (“pais”), estes são ordenados gene a gene, de modo a permitir que as subestações próximas entre si fiquem em genes próximos. Sem esta ordenação, é possível que os cruzamentos apresentem muitos “filhos” inviáveis, uma vez que existem limitações físicas de atendimento a uma demanda de carga entre uma subestação e outra. Subestações muito distantes entre si não teriam viabilidade elétrica para “trocar” carga.
2. Selecionar o ponto de corte (ou ponto de cruzamento), ou seja, determinar a partir de qual gene o material genético entre os cromossomos será “trocado”. No programa utilizado, as opções disponíveis para o usuário são “1”, onde o ponto de corte será escolhido através de uma função, ou “0”, o ponto de corte será escolhido aleatoriamente.
3. Selecionar quantos genes a partir do ponto de corte sofrerão influência dos operadores genéticos. Este número é um inteiro pertencente ao intervalo  $[31, 30 + n]$ , onde  $n$  representa, como já mencionado, o número de novas possíveis subestações. O operador genético *crossover* é executado enquanto houver melhora no *fitness* da população.

### 3.3 Aplicação do algoritmo Teitz e Bart (TB)

O programa desenvolvido em MATLAB contém o algoritmo de TB apresentado na seção 2.2 e, para executá-lo, é preciso indicar a melhor solução obtida através do AG, ou seja, a localização das novas  $n$  ( $= 5$ ) subestações, indicadas pelas quadrículas.

O objetivo do algoritmo TB é refinar a solução fornecida pelo AG (localização das subestações), ou seja, tentar melhorar tais localizações de forma a obter o momento elétrico total mínimo.

### 3.4 Aplicação do algoritmo de Gillett e Johnson (GJ)

A função GJ do programa utiliza o algoritmo apresentado na seção 2.3. Este algoritmo objetiva distribuir (designar) as demandas de cargas de cada quadrícula para as subestações que estejam mais próximas, levando-se em conta os valores das demandas e as capacidades de atendimento das subestações.

Basicamente, a função GJ procura designar demandas de carga a subestações que as possam atender e, ao mesmo tempo, que exijam o menor momento elétrico possível. Para cumprir tal propósito, um número  $r(i)$  que

indica a urgência que existe em se designar certa demanda a uma determinada subestação é calculado, para garantir o mínimo momento elétrico, conforme já mostrado na seção 2.3.

## 4. Resultados

A forma como o programa principal foi elaborado permite a interação com os engenheiros responsáveis pelo planejamento da distribuição da concessionária (Alves, 2002). São permitidos estudos de casos diversos, sob circunstâncias diversas. Dentre as variações possíveis, podem ser destacados: o fator de multiplicação das cargas (FMC); o tipo de mutação, pois a escolha do gene que sofrerá mutação pode ser ou não aleatória; a mutação pode ser do tipo usual (um gene é trocado) ou do tipo multimutação (dois ou mais genes podem ser trocados); o tipo de cruzamento, visto que os cruzamentos podem ocorrer apenas nos  $n$  genes que estão situados à direita dos 30 primeiros genes, que representam subestações fixas já existentes. Pode-se ainda escolher quantas partições terá o cruzamento.

Alguns resultados são apresentados na tabela 1 que inclui dados como: número do estudo de caso (indicado por EC); número de possíveis novas subestações (NSE); número de cromossomos na população inicial (POP); número de iterações realizadas para obtenção da solução (NIT); cruzamentos efetivos (CE); o número de vezes em que mutações foram efetivas (ME); fator de multiplicação de cargas (FMC); tempo de processamento em segundos (TPS); carga total atendida em KVA (CTA) e momento elétrico total em KVA/km (MET).

As três primeiras linhas de cada caso referem-se a cruzamentos de uma partição e mutações usuais. A quarta linha de cada um dos casos refere-se a cruzamentos com duas (ou mais) partições e multimutações; a quinta linha de cada caso refere-se ao cruzamento de uma partição e mutação usual com aplicação do fator de multiplicação de cargas.

No caso de duas novas SE's, por exemplo, nota-se que a melhor solução em termos de momento elétrico mínimo se dá com população inicial de 70 cromossomos. Note-se, porém, que esta não é a melhor solução em tempo computacional. Isto indica que a figura humana do engenheiro responsável pelo planejamento da distribuição continua essencial no processo da decisão, uma vez que as prioridades devem ser por ele estabelecidas.

Os gráficos em MATLAB que mostram localizações ótimas de novas subestações para a região de estudos (levando-se em conta as subestações já existentes) bem como relatórios comparativos e os mapas com a localização atual das SE's já existentes na região de estudos, além de detalhes relativos a dados e implementação, podem ser encontrados em Alves (2002).

Tabela 1. Resultados obtidos em simulações.

EC	NSE	POP	NIT	CE	ME	FMC	TPS	CTA	MET
1	2	50	660	114	8	1, 0	522, 78	1074441, 4434	2410752, 9135
	2	70	880	162	5	1, 0	738, 47	1074441, 4434	<b>2404392, 7553</b>
	2	100	1210	226	4	1, 0	1114, 16	1074441, 4434	2410706, 9461
	2	50	660	99	14	1, 0	520, 42	1074441, 4434	2426580, 7237
	2	50	660	124	12	1, 2	535, 80	1289329, 7321	3102563, 5211
2	3	50	660	181	3	1, 0	545, 02	1074441, 4434	2365808, 196
	3	70	880	249	3	1, 0	803, 17	1074441, 4434	2359764, 1721
	3	100	1210	397	7	1, 0	1129, 26	1074441, 4434	<b>2353716, 7041</b>
	3	50	660	156	5	1, 0	545, 68	1074441, 4434	2367253, 8042
	3	50	660	178	5	1, 2	564, 91	1289329, 7321	3028174, 275
3	4	50	660	227	1	1, 0	584, 90	1074441, 4434	2331332, 5941
	4	70	880	260	5	1, 0	790, 76	1074441, 4434	2322294, 322
	4	100	1210	481	4	1, 0	1162, 45	1074441, 4434	<b>2309606, 8806</b>
	4	50	660	212	2	1, 0	575, 45	1074441, 4434	2310477, 186
	4	50	660	218	3	1, 2	575, 56	1289329, 7321	2972560, 9579
4	5	50	660	202	4	1, 0	588, 36	1074441, 4434	2281608, 1026
	5	70	880	292	7	1, 0	830, 63	1074441, 4434	2275504, 3639
	5	100	1210	502	6	1, 0	1208, 59	1074441, 4434	<b>2259551, 3123</b>
	5	50	660	195	3	1, 0	596, 27	1074441, 4434	2280611, 0121
	5	50	660	222	11	1, 2	609, 02	1289329, 7321	2876943, 6252

## 5. Conclusões

O estudo do problema de localização de instalações, aplicado ao problema da localização das SE's tem como objetivo a automação da localização de novas possíveis subestações. Com este trabalho, pretende-se desenvolver uma ferramenta útil aos engenheiros da área de distribuição responsáveis pela localização otimizada, de forma local ou até mesmo global das mesmas.

É certo que muitas alternativas na geração de energia são pesquisadas como, por exemplo: geração eólica, solar, pelas oscilações das ondas dos mares, etc. Mas, sem dúvida, além de avanço e pesquisa quanto à geração de energia, precisa-se ter em mente a preocupação com a sua distribuição. Para tanto, é vital um planejamento adequado, que se inicia sempre pela decisão do melhor local de construção de uma SE. Nesse trabalho, propõe-se a utilização de Algoritmos Genéticos para a concessionária.

A metodologia proposta serve como auxílio para a solução do problema, mas, a figura do tomador de decisões, neste caso, o engenheiro responsável pelo planejamento de subestações é imprescindível. Quando se procura unicamente a minimização do momento elétrico, muitas soluções satisfatórias podem ser encontradas. Entretanto, dentre estas, pode haver aquela cujo resultado não seja “o melhor” matematicamente mas, quando considerado o contexto do empreendimento (a construção de uma nova subestação), é possível que fatores como disponibilidade do terreno ideal, incluindo seu custo e outros, a torne mais atrativa do que a solução “melhor” em termos matemáticos. As opções e decisões pertencem ao cabedal de experiência do setor de planejamento.

Outros estudos podem ser realizados nesta direção e algumas sugestões para trabalhos são: explorar o assunto para outros anos do horizonte de previsão do setor estudos de mercado através da Programação Dinâmica; verificar, junto às concessionárias locais sobre quais melhorias poderiam ser realizadas no programa computacional para a sua utilização efetiva; incluir cálculos que envolvam investimentos na rede, a fim de detectar se a ampliação da capacidade de uma subestação é ou não mais apropriada do que a construção de uma nova subestação, a curto, médio ou longo prazo. Com relação ao problema real, um estudo sobre técnicas aplicáveis ao problema de cargas que deixam de ser atendidas pelas subestações pode ser alvo de pesquisa; com relação ao problema das  $p$ -medianas, é possível analisar outras heurísticas para solucioná-lo. Segundo Glover et al. (2000), o *Scatter Search* e *Path Relinking* têm sido mais eficientes do que os algoritmos genéticos e a busca tabu em alguns problemas.

## Referências

- Alves, A., *Algoritmos genéticos aplicados ao Planejamento da Distribuição de Energia Elétrica em Curitiba e Região Metropolitana – A Localização das Subestações*. Dissertação de mestrado, Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba, PR, 2002.
- Bodin, L.; Golden, B.; Assad, A. & Ball, M., Routing and scheduling of vehicles and crews – the state of the art. *Computers and Operations Research*, 10(2):63–212, 1983.
- Corrêa, E.S., *Algoritmos Genéticos e Busca Tabu Aplicados ao Problema das  $p$ -medianas*. Dissertação de mestrado, Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba, PR, 2000.
- Corrêa, E.S.; Steiner, M.T.A.; Freitas, A.A. & Carnieri, C., A genetic algorithm for solving a capacitated  $p$ -median problem. *Numerical Algorithms*, 35(2-4):373–388, 2004.
- Cunha, A.S. & Pinto, R.L.U.S., Uma técnica para ajuste dos parâmetros de um algoritmo genético. In: *Anais do XXXIII Simpósio Brasileiro de Pesquisa Operacional*. Campos do Jordão, SP, p. 1105–1116, 2001.
- Detofeno, T.C. & Steiner, M.T.A., Optimizing routes for the collection of urban solid waste: a case study for the city of Joinville, state of Santa Catarina. *IJIE - Revista Iberoamericana de Engenharia Industrial*, 2(1):124–136, 2010.
- Glover, F.; Laguna, M. & Martí, R.. *Control and Cybernetics*, 29(3):653–684, 2000.
- Gruppelli Jr., F.A.; Yuan, J.Y.; Carnieri, C.; Volpi, N.M.P.; Steiner, M.T.A.; Wilhelm, V.E.; Mussi, N.H.; Antonio, C.; Miqueles, E.; Andretta Filho, E.L.; Kalinowski, A.; Alves, A.C.; Gulín, C.; Zambenedetti, V.C. & Klimkowski, M., Algoritmos para recomposição de sis-

- temas de distribuição. In: *Anais do XV SENDI - Seminário Nacional de Distribuição de Energia Elétrica*. Salvador, BA, 2002.
- Mayerle, S.F., Um algoritmo genético para a solução do problema do caixeiro viajante. In: *Anais do 14º ENEGEP - Encontro Nacional de Engenharia de Produção*. João Pessoa, PB, 1994.
- Reeves, C.R., *Modern Heuristic Techniques for Combinatorial Problems*. New York, USA: McGraw-Hill, 2000. 320 p.
- Revelle, C.S. & Swain, R.W., Central facilities location. *Geographical Analysis*, 2(1):30–42, 1970.
- Sampaio, M.E.C.S., *Aplicação de Metaheurísticas ao problema de localização de escolas de ensino fundamental*. Dissertação de mestrado, Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba, PR, 1999.
- Savulescu, S.C., *Grafos, dígrafos e redes elétricas - aplicações na Pesquisa Operacional*. São Paulo, SP: Editora IBEC, 1981.
- Scarpin, C.T.; Steiner, M.T.A.; Dias, G.J.C. & Steiner Neto, P.J., Otimização no serviço de saúde do estado do Paraná: fluxo de pacientes e novas configurações hierárquicas. *Revista Gestão & Produção*, 15(2):275–290, 2008.
- Smiderle, A.; Steiner, M.T.A. & Wilhelm, V., Técnicas da pesquisa operacional aplicadas a um problema de cobertura de arcos. *TEMA – Tendências em Matemática Aplicada e Computacional*, 5(2):347–356, 2004.
- Tanomaru, J., Motivação, fundamentos e aplicações de algoritmos genéticos. In: *Anais do II Congresso Brasileiro de Redes Neurais*. Curitiba, PR, p. 373–403, 1995.
- Teitz, M.B. & Bart, P., Heuristic methods for estimating the generalized vertex median of weighted graph. *Operations Research*, 16(5):955–961, 1968.

## Notas Biográficas

**Arleide Cristina Alves** é Mestre em Métodos Numéricos em Engenharia e Doutora em Engenharia Mecânica, ambos pela Universidade Federal do Paraná (UFPR). Atualmente é professora na Universidade Positivo (UP) e na Universidade Tecnológica Federal do Paraná (UTFPR) nas engenharias Elétrica, Mecânica e da Computação.

**Maria Teresinha Arns Steiner** é Mestre e Doutora em Engenharia de Produção pela Universidade Federal de Santa Catarina (UFSC) e tem Pós-Doutorado em Pesquisa Operacional pelo Instituto Tecnológico da Aeronáutica (ITA). Atualmente é professora sênior dos Programas de Pós-Graduação em Métodos Numéricos em Engenharia (Mestrado e Doutorado) e de Engenharia de Produção (Mestrado), ambos da Universidade Federal do Paraná (UFPR).

**Nelson Haj Mussi** é Mestre em Métodos Numéricos em Engenharia pela Universidade Federal do Paraná (UFPR). É funcionário aposentado da COPEL (Companhia Paranaense de Energia Elétrica).

**Giovani Zanelatto** é Mestre e Doutor em Física pela Universidade Federal de São Carlos e tem Pós-Doutorado em Física pela University of California, Santa Barbara. Atualmente é professor na Universidade Positivo (UP).

### Otimização Multiobjetivo da Operação de uma Usina Hidroelétrica

Luís A. Scola\*, Oriane M. Neto,  
Ricardo H.C. Takahashi e Sérgio A.A.G. Cerqueira

---

**Resumo:** Neste trabalho, um algoritmo genético multiobjetivo é aplicado ao problema da otimização de uma única usina hidroelétrica, com os objetivos de aumentar a energia produzida durante o ano e reduzir o pico de demanda de energia complementar proveniente de fontes não renováveis de energia. Para aumentar o desempenho do algoritmo, foram propostas duas novas formulações para o problema, com diferentes maneiras de manipular as restrições operacionais. Os resultados demonstram que, em comparação com a abordagem tradicional, há ganho em eficiência computacional e expansão da frente de Pareto obtida, o que revela novas configurações possíveis para a operação do sistema.

**Palavras-chave:** Otimização multiobjetivo, Algoritmos genéticos, Operação de usinas hidroelétricas.

**Abstract:** *In this work, a multi-objective genetic algorithm is applied to problem of the optimization of a single Brazilian hydropower plant, with the objectives of increasing the net energy generation along the year and reducing the peak of demand of non-renewable energy sources. To increase the performance of the algorithm, two new formulations for the problem are proposed, with different ways of dealing with the operational constraints. In comparison with the more traditional approach, the proposed methodology presents a gain of computational efficiency and also leads to an expanded Pareto front, which reveals new possible configurations of system operation.*

**Keywords:** *Multiobjective optimization, Genetic algorithms, Hydropower operation.*

---

\* Autor para contato: antonioscola@gmail.com

## 1. Introdução

No Brasil mais de 73% da potência elétrica é produzida por usinas hidroelétricas (Tolmasquim & Guerreiro, 2009). Apesar do potencial hidráulico remanescente ser considerável, sua maior parte está localizada longe dos estados consumidores industrializados do sudeste. Em adição a este fato, a crescente oposição à construção de novos grandes reservatórios por razões ecológicas e sociais, indicam a necessidade de uma operação eficiente do sistema existente (Labadie, 2004).

O problema da otimização da operação de sistemas com único e múltiplos reservatórios é não linear, não convexo e restrito, tendo sido abordado por diversos métodos determinísticos e estocásticos (Labadie, 2004; Yeh, 1985). Na última década, métodos evolucionários mono-objetivo, em particular diversas versões de algoritmos genéticos, foram aplicados a sistemas de único reservatório (Chang et al., 2005; Jothiprakash & Shanthi, 2006; Cheng et al., 2008).

Tantos são os fatores envolvidos na operação de usinas hidroelétricas, que se torna natural que diversas abordagens para a otimização multiobjetivo do problema tenham sido propostas, para sistemas de único e múltiplos reservatórios. O modelo não linear do sistema hidroelétrico brasileiro completo foi linearizado e resolvido através de técnicas de programação linear, considerando a soma ponderada de seis objetivos (Barros et al., 2003). Um algoritmo genético foi aplicado para a otimização da operação de um reservatório para usos múltiplos, considerando dois objetivos (Reddy & Kumar, 2006). Os mesmos autores empregaram uma versão elitista-mutada do algoritmo de enxame de partículas, considerando a soma ponderada de dois objetivos (Kumar & Reddy, 2007). Uma característica comum a todos estes trabalhos é a preocupação com o tratamento da perda de diversidade da população. Por exemplo, um algoritmo genético multiobjetivo macroevolucionário foi desenvolvido para a otimização das curvas de operação de um sistema de reservatório para usos múltiplos (Chen et al., 2007).

Neste trabalho um algoritmo genético multiobjetivo é empregado para a otimização da operação de uma usina hidroelétrica brasileira ao longo de um ano típico. São propostas duas formulações matemáticas alternativas para o problema, com diferentes maneiras de tratar as restrições operacionais. As formulações propostas mostram-se mais favoráveis para a otimização multiobjetivo com algoritmos genéticos do que a abordagem tradicional.

## 2. Modelo Matemático de uma Usina Hidroelétrica

### 2.1 Geração de potência

A potência gerada numa usina hidroelétrica em um dado intervalo  $i$ , é dada pela Equação 1:

$$\dot{W}_i = \eta Q_i^t \gamma_w (h_f - h_t) \quad (1)$$

onde  $\eta$  é a eficiência combinada da turbina e gerador elétrico,  $Q_i^t$  é a vazão turbinada,  $\gamma_w$  é o peso específico da água,  $h_f$  é o nível do reservatório de captação e  $h_t$  é o nível no canal de fuga.

Como a eficiência e o peso específico são aproximadamente constantes e a diferença de níveis entre a captação e o canal de fuga é determinada principalmente pelas variações do primeiro, uma vez que o nível do canal de fuga pode ser considerado relativamente estável, a potência gerada pode ser aproximada pela seguinte expressão:

$$\dot{W}_i = \xi_i Q_i^t \quad (2)$$

onde  $\xi_i$  é a função produtividade, que é neste trabalho determinada por um polinômio de quinto grau em função do volume médio armazenado no reservatório durante o intervalo.

### 2.2 Objetivos

Na literatura são encontrados diferentes objetivos representando medidas quantitativas de diferentes aspectos específicos do sistema energético abordado. Alguns exemplos destas funções objetivo são apresentados a seguir, começando pelos objetivos considerados neste trabalho. Esses foram escolhidos porque tratam dos principais aspectos físicos da operação da usina, geração de energia e potência complementar, excluindo as volatilidades dos aspectos econômicos.

#### 2.2.1 Maximizar a geração de energia

O primeiro, a maximização da energia total gerada, é auto explicativa. Em termos econômicos, ela é linearmente correlacionada com o faturamento bruto da usina se o preço é fixo ao longo do ano, dado pela expressão:

$$\max W_T = \sum_{i=1}^N \dot{W}_i \Delta t_i \quad (3)$$

onde  $\Delta t_i$  é o intervalo de tempo considerado e  $N$  o número de intervalos.

#### 2.2.2 Maximizar a mínima potência gerada

O segundo objetivo (Equação 4) é a maximização da mínima potência gerada mensalmente, o que é equivalente à minimização do custo com instalações para geração da energia complementar.

$$\max \min_i \dot{W}_i \quad \forall i \in \{1, 2, \dots, N\} \quad (4)$$

Este objetivo torna-se particularmente importante quando o regime pluviométrico está sujeito a grandes variações que, na ausência de um grande reservatório, produziriam variações correspondentes na geração de potência. A potência complementar necessária é geralmente proveniente de fontes térmicas ou nucleares, cujos custos são mais elevados.

### 2.2.3 Minimizar a complementação da energia

Quando o método de otimização utilizado exige uma função objetivo de derivada contínua, o objetivo anterior é aproximado pelo somatório do quadrado da diferença entre a demanda e a potência gerada (Sharma et al., 2004; Barros et al., 2003):

$$\min \sum_{i=1}^N (D_T - \dot{W}_i)^2, \quad (5)$$

onde  $D_T$  é a demanda total de energia, que será sempre superior à capacidade da usina, pois inclui outras usinas hidroelétricas, além de outras fontes energéticas.

### 2.2.4 Minimizar o desvio do volume armazenado

Estabelecida uma meta para o volume mensal do reservatório ( $S_i$ ), pode-se propor como objetivo a soma no tempo do desvio quadrático entre a meta e o volume armazenado no reservatório (Barros et al., 2005):

$$\min \sum_{i=1}^N (S_i - V_i)^2, \quad (6)$$

Uma versão para este objetivo, com derivada descontínua, poderia ser proposta na forma:

$$\min \max(S_i - V_i) \quad \forall i \in \{1, 2, \dots, N\}, \quad (7)$$

com a vantagem de que somente seriam considerados os finais de período em que o volume armazenado fosse inferior à meta.

### 2.2.5 Minimizar a energia vertida

Mesmo em reservatórios isolados há momentos em que é necessário usar o vertedouro, por exemplo quando o volume do reservatório está em seu limite superior e a vazão afluente é maior que máxima vazão turbinada. O objetivo de reduzir a energia que deixa de ser produzida quando água é vertida é expresso como (Barros et al., 2005):

$$\min \sum_{i=1}^N \xi_i Q_i^s \Delta t_i, \quad (8)$$

onde  $Q_i^s$  é a vazão vertida.

### 2.2.6 Maximizar o lucro derivado da energia secundária

No sistema elétrico brasileiro, define-se a energia assegurada como a máxima produção de energia que pode ser mantida quase que continuamente pelas usinas hidrelétricas ao longo dos anos, admitindo risco de não atendimento à carga inferior a 5%. O montante produzido acima desse valor é denominado energia secundária e tem preço de venda inferior ao da energia assegurada. (Barros et al., 2003) propõe a seguinte formulação para a maximização do lucro obtido com a venda da energia secundária:

$$\max \sum_{i=1}^N p_i (\dot{W}_i - D_c) \Delta t_i, \quad (9)$$

onde  $p_i$  é o preço da energia durante o período de tempo  $i$  (\$/MWh) e  $D_c$  é a demanda contratada (energia assegurada).

### 2.3 Restrições

O problema é sujeito a restrições de natureza física e operacional.

#### 2.3.1 Conservação da massa

Para garantir a conservação da massa num dado intervalo de tempo, a variação do volume do reservatório entre o início do intervalo ( $V_i$ ) e o final ( $V_{i+1}$ ) deve ser igual ao volume de água afluente ( $V_i^a$ ), deduzidos os volumes turbinado ( $V_i^t$ ) e vertido ( $V_i^s$ ).

$$V_{i+1} - V_i = V_i^a - V_i^t - V_i^s \quad (10)$$

A conservação da massa estabelece uma relação direta entre o volume de água descarregado no canal de fuga e o volume do reservatório, uma vez que o volume afluente é conhecido. Neste trabalho, nas duas primeiras formulações apresentadas na Seção 3, o volume do reservatório é adotado como variável independente.

#### 2.3.2 Limites operacionais

O volume de água no reservatório é limitado no nível inferior pelo nível da tomada d'água e no nível superior pelo limite estrutural da barragem.

$$V^{min} \leq V_i \leq V^{max} \quad (11)$$

Os parâmetros operacionais das turbinas e dos geradores também impõem restrições ao problema, em termos da potência máxima gerada e da máxima vazão turbinada admissível.

$$\dot{W}_i \leq \dot{W}^{max} \quad (12)$$

$$Q_i^t \leq Q^{t,max} \quad (13)$$

Se a variação do volume num dado intervalo requer uma vazão turbinada que excede o máximo permitido, assume-se que essa parcela excedente da vazão foi vertida.

Uma vazão mínima de restituição ao rio deve ser garantida, por razões ecológicas, sanitárias e econômicas. Como neste trabalho considera-se que as turbinas estão sempre disponíveis, essa vazão mínima é sempre turbinada.

$$Q^{min} \leq Q_i^t \quad (14)$$

Finalmente, é necessário impor uma condição para o reservatório ao final do período considerado na otimização. Neste trabalho, assume-se que o reservatório deve retornar à condição inicial, isto é, que

$$V_0 = V_N \quad (15)$$

No algoritmo de otimização, esta restrição foi implementada reduzindo em um o número de variáveis de decisão e fixando o volume ao final do último intervalo.

### 3. Formulações Alternativas

Como será mostrado na próxima seção, as restrições não lineares representam um problema para o algoritmo genético, que tem dificuldade para encontrar indivíduos factíveis. Como resultado, a população do algoritmo genético que resolve o problema como descrito na Seção 2 apresenta perda de diversidade e a frente de Pareto produzida cobre somente um pequeno intervalo das funções objetivo.

Três abordagens foram testadas neste trabalho para enfrentar este problema e são comentadas a seguir.

#### 3.1 Redução do número de indivíduos inactíveis na população inicial

Em experimentos preliminares, verificou-se que o algoritmo de otimização não era capaz de gerar indivíduos factíveis quando a população inicial era gerada de forma aleatória sobre todo o domínio das variáveis de decisão. Por outro lado, observou-se ser possível gerar uma população inicial composta apenas por indivíduos factíveis ou, se inactíveis, próximos à região factível, com poucas restrições violadas e/ou com pequenas violações (quase factíveis). Para tanto mantiveram-se os valores das variáveis de decisão, os volumes do reservatório mês a mês, dentro da estreita faixa, entre 85% e 100% do volume máximo do reservatório. Esta abordagem será doravante denominada “formulação básica”.

### 3.2 Redução do domínio da variáveis de decisão

Observou-se que a maioria dos indivíduos ineficazes eram relacionados com a necessidade de recuperar o volume do reservatório, se ao final de qualquer intervalo o seu volume fosse significativamente menor que o inicial. Se a vazão afluente fosse insuficiente, a conservação da massa produziria uma vazão turbinada ineficaz, podendo ser até mesmo negativa. Para enfrentar tal problema, ao invés de impor um limite inferior único para o volume mínimo do reservatório, foi imposto um limite para cada intervalo.

$$V_{i-1}^{min} = V_i^{min} - (Q_i^a - Q_t^{min})\Delta t_i \quad \forall i \in \{1, 2, \dots, N-1\} \quad (16)$$

O limite inferior para o volume no início de cada período é igual ao volume mínimo ao final do mesmo período, deduzida a diferença entre o volume afluente e o volume turbinado mínimo. Como o volume final é conhecido, o conjunto de limites inferiores para o volume é determinado pela equação precedente, do último intervalo para o primeiro, antes de se iniciar a otimização.

### 3.3 Formulação irrestrita

Mesmo na formulação melhorada para o limite inferior do volume, uma parte significativa dos indivíduos gerados a cada geração era ineficaz. Este comportamento era mais pronunciado quando o reservatório enchia novamente, antes da estação mais seca. Aqui também a vazão afluente se mostrava insuficiente para permitir um aumento no volume, mantida a vazão turbinada mínima. Uma modificação para o limite superior do volume em cada intervalo foi então proposta para lidar com este problema.

$$V_{i+1}^{max} = \begin{cases} V_i + (Q_i^a - Q_t^{min})\Delta t_i, & \text{se } < V^{max} \\ V^{max}, & \text{de outra forma} \end{cases} \quad \forall i \in \{1, 2, \dots, N-1\} \quad (17)$$

Cabe ressaltar que o valor do limite superior ao final de cada intervalo é dependente do valor do volume no início daquele intervalo. Por esta razão, diferentemente ao limite inferior do volume a cada intervalo, que podia ser determinado previamente à otimização, o limite superior para o volume só pode ser determinado para cada indivíduo da população.

Para lidar com este problema foi proposta uma formulação irrestrita, mas com domínio definido, onde as variáveis de decisão são frações dos volumes disponíveis mensalmente.

$$V_i = V_i^{min} + \alpha_i (V_i^{max} - V_i^{min}), \quad \alpha \in [0, 1] \quad (18)$$

Dentro destes limites, as vazões turbinadas são sempre maiores que a mínima, embora ainda possam tanto levar a potências geradas maiores

que a máxima permitida quanto ultrapassarem o limite da vazão máxima. Para lidar com esses casos, assume-se que uma parcela da vazão foi vertida e que a vazão verdadeira através da turbina é a que satisfaz a ambas as restrições 12 e 13.

#### 4. Estudo de Caso

A usina hidroelétrica Nova Ponte, no leito rio Araguari, concluída em 1994, foi um dos últimos grandes reservatórios construídos no sudeste brasileiro. Operada pela Companhia Energética de Minas Gerais, CEMIG, seu reservatório apresenta capacidade máxima de armazenamento de 12.792 hm<sup>3</sup>, operando a partir do volume mínimo de 2.412 hm<sup>3</sup>. Com potência nominal total de 510 MW, está equipada com três turbinas hidráulicas tipo Francis, que juntas admitem um mínimo de 125 e um máximo de 510 m<sup>3</sup>/s de água

O clima na região é caracterizado por duas estações bem definidas, a primeira seca, entre os meses de abril e setembro, e a segunda chuvosa, estendendo-se de outubro a março. Ele reflete na vazão afluente ao reservatório, para a qual dados estão disponíveis desde 1931. Em termos médios 125 m<sup>3</sup>/s de água chegam ao reservatório no mês de agosto, enquanto em fevereiro a média é de 541 m<sup>3</sup>/s, com uma afluência máxima de 1243 m<sup>3</sup>/s.

A potência gerada pela usina hidroelétrica durante um intervalo é determinada pela Equação 2, onde a função produtividade é determinada pelo seguinte polinômio de quinto grau, determinado pelo ONS (Operador Nacional do Sistema) e fornecido pela operadora, CEMIG.

$$\begin{aligned} \xi_i = & 1.95365 \cdot 10^{-21} \bar{V}_i^5 - 6.49561 \cdot 10^{-17} \bar{V}_i^4 + \\ & 9.11797 \cdot 10^{-13} \bar{V}_i^3 - 7.87711 \cdot 10^{-09} \bar{V}_i^2 + \\ & 6.84174 \cdot 10^{-05} \bar{V}_i + 0.72616 \end{aligned} \quad (19)$$

onde  $\bar{V}_i = (V_i + V_{i-1})/2$ .

#### 5. Resultados

A otimização foi realizada utilizando o algoritmo NSGA-II padrão, a factibilidade foi tratada como parte do operador de seleção (Deb et al., 2000). Os experimentos foram realizados com dados de um ano típico (de maio de 1976 a abril de 1977), com os volumes do reservatório no início e no final da otimização fixados em 95 por cento do máximo permitido.

Somente alguns poucos dos experimentos realizados, empregando a formulação básica, produziram populações com indivíduos factíveis, mesmo quando a população inicial foi trabalhada para incluir somente indivíduos factíveis ou quase factíveis. As frentes de Pareto para 8 desses experimentos, com populações de 400 indivíduos e 5000 gerações são apresentadas na

Figura 1. O primeiro objetivo, a energia total gerada (MWh), é mostrada no eixo horizontal e o segundo objetivo, a mínima potência gerada (MW), está representada no eixo vertical. Em todos os testes, as frentes de Pareto apresentam-se estreitas, estendendo-se mais em direção ao primeiro objetivo.

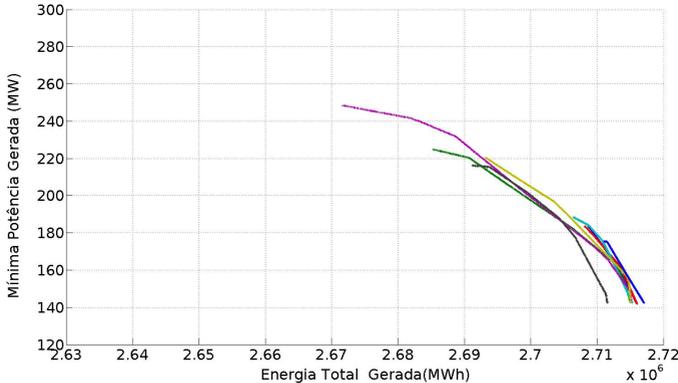


Figura 1. Conjuntos eficientes para 8 execuções da formulação básica, com 400 indivíduos e 5000 gerações.

Para cada uma das outras duas formulações, o algoritmo foi executado 21 vezes, com 400 indivíduos e 2500 gerações. A formulação da redução do domínio das variáveis de decisão obteve uma frente de Pareto completamente povoada, mas com grande dispersão nos resultados entre as execuções, conforme mostrado na Figura 2. Algumas das frentes são de fato inferiores àquelas encontradas com a formulação básica, mas neste caso a população inicial foi gerada aleatoriamente tornando imprópria a comparação.

A formulação irrestrita produziu a cada execução uma frente de Pareto ampla e bem povoada, conforme mostrado na Figura 3. Na Figura 4, são apresentadas todas as soluções eficientes e as frentes de Pareto combinadas para ambas formulações. É evidente desses experimentos que a formulação irrestrita gera melhores resultados, produzindo frentes mais amplas, que se comparam ou até mesmo superam a frente de Pareto combinada da formulação da redução do domínio das variáveis de decisão.

No que tange à operação da usina hidroelétrica, dentre os pontos eficientes gerados pela formulação irrestrita, a menor potência mínima gerada tem um valor de 141,8 MW, correspondendo a energia máxima produzida de  $2,718 \times 10^3$  GWh. No outro extremo da frente de Pareto a maior potência mínima gerada foi de 297,6 MW, com  $2,643 \times 10^3$  GWh de energia gerada.

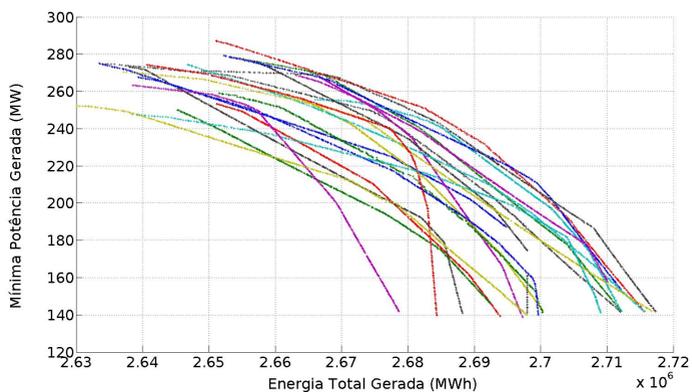


Figura 2. Conjuntos eficientes para 21 execuções da formulação de redução do domínio das variáveis de decisão, com 400 indivíduos e 2500 gerações.

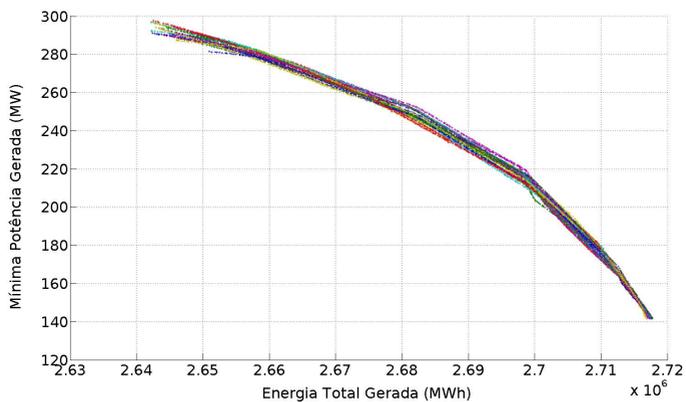


Figura 3. Conjuntos eficientes para 21 execuções da formulação irrestrita, com 400 indivíduos e 2500 gerações.

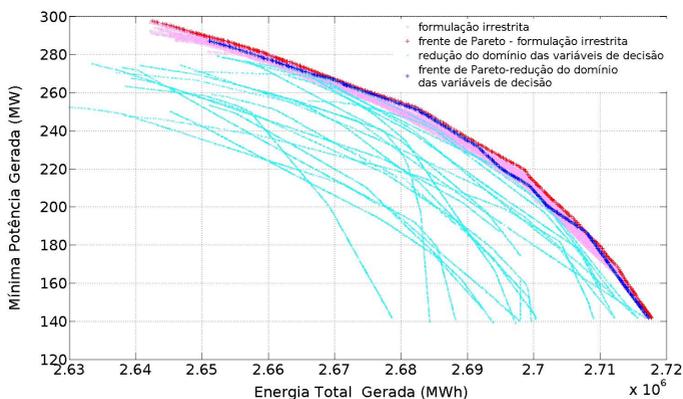


Figura 4. Conjuntos eficientes para cada execução das formulações da redução de domínio da variável de decisão e da irrestrita, com as respectivas frentes de Pareto combinadas.

As Figuras 5 e 6, apresentam as potências geradas mensalmente para estes extremos, respectivamente.

Na primeira solução extrema, durante a estação seca, a potência gerada é mantida no mínimo e o reservatório acumula água. Quando chegam as chuvas, a potência gerada é aumentada, com o elevado nível do reservatório assegurando elevada potência por unidade de massa de água turbinada.

Na segunda solução extrema, o volume do reservatório varia consideravelmente, diminuindo durante a estação seca e recuperando durante a estação chuvosa, até atingir o nível inicial. A geração é mantida estável, com uma perda de 25 GWh na comparação com o outro extremo.

## 6. Conclusões

São muitas e conflitantes as medidas da qualidade da operação de usinas hidroelétricas. Soluções de compromisso entre tais objetivos podem ser encontradas através da otimização multiobjetivo, permitindo ao operador a tomada de decisões sobre o conjunto de soluções eficientes. Neste trabalho consideraram-se como objetivos duas medidas do desempenho físico de uma usina hidroelétrica na otimização multiobjetivo de sua operação ao longo de um ano típico. A utilização de algoritmo genético, não dependente da existência ou continuidade das funções objetivo, permitiu que se empregasse uma medida mais apropriada da máxima demanda de potência complementar.

Diante da perda da diversidade observada na evolução do algoritmo genético de otimização, causada pelo grande número de soluções ineficazes,

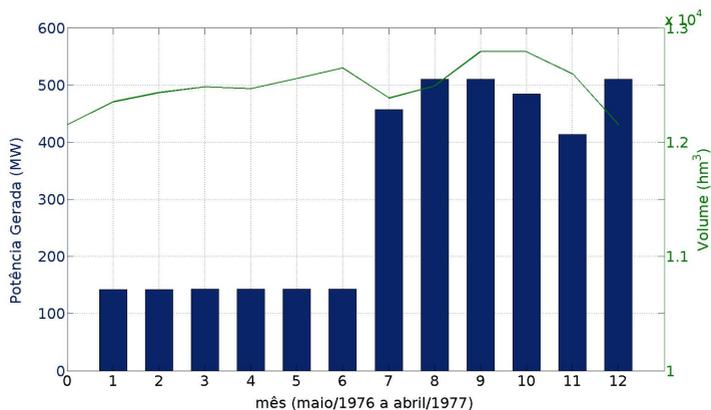


Figura 5. Potência mensal gerada e volume do reservatório para o ponto extremo com a menor potência gerada e a maior geração de energia.

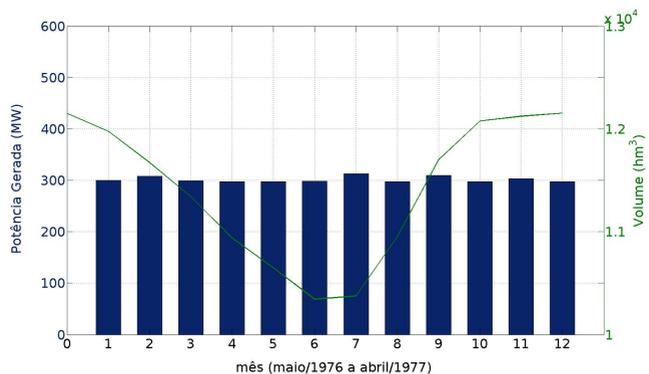


Figura 6. Potência mensal gerada e volume do reservatório para o ponto extremo com a maior potência mínima gerada e a menor geração de energia.

foram propostas duas novas formulações para o problema. Na primeira, a imposição de limites mais estreitos para as variáveis de decisão, o volume do reservatório ao final de cada período, permitiu que mais soluções factíveis fossem obtidas. Ainda assim houve grande espalhamento entre os resultados gerados, no conjunto dos experimentos.

Na segunda formulação, através de uma mudança nas variáveis de decisão, as restrições foram incorporadas ao modelo. Isso o tornou irrestrito, com as variáveis de decisão limitadas a um intervalo. Disso resultou um expressivo aumento do desempenho do algoritmo de otimização, sendo observada uma convergência nos resultados encontrados.

Diante dos resultados, um operador estaria melhor capacitado a decidir sobre a operação da usina. A vazão afluyente ao reservatório é, entretanto, dependente dos aspectos meteorológicos e portanto incerta. A base sobre a qual se toma a decisão sobre a operação estaria fragilizada se tal incerteza não fosse incorporada à tomada de decisão. O ganho de desempenho decorrente da adoção do modelo desenvolvido permite que se planeje incorporar futuramente tal incerteza já no momento da otimização.

## Agradecimentos

Os autores agradecem o apoio da Fundação Estadual de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

## Referências

- Barros, M.; Ros, D. & Lopes, J., Objective functions for hydropower system operation. In: Moglen, G. (Ed.), *Proceedings of Watershed 2005*. ASCE, p. 101–101, 2005.
- Barros, M.; Tsai, F.C.; Yang, S.L.; Lopes, J. & Yeh, J.G., Optimization of large-scale hydropower system operations. *Journal of Water Resources Planning and Management*, 129(3):178–188, 2003.
- Chang, F.J.; Chen, L. & Chang, L.C., Optimizing the reservoir operating rule curves by genetic algorithms. *Hydrological Processes*, 19(11):2277–2289, 2005.
- Chen, L.; McPhee, J. & Yeh, W., A diversified multiobjective GA for optimizing reservoir rule curves. *Advances in Water Resources*, 30(5):1082 – 1093, 2007.
- Cheng, C.T.; Wang, W.C.; Xu, D.M. & Chau, K., Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos. *Water Resources Management*, 22(7):895–909, 2008.
- Deb, K.; Agrawal, S.; Pratab, A. & Meyarivan, T., A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization:

- NSGA-II. In: *Proceedings of the Parallel Problem Solving from Nature VI Conference*. Paris, France: Springer-Verlag, p. 849–858, 2000, LNCS, v. 1917.
- Jothiprakash, V. & Shanthi, G., Single reservoir operating policies using genetic algorithm. *Water Resources Management*, 20(6):917–929, 2006.
- Kumar, D. & Reddy, M., Multipurpose reservoir operation using particle swarm optimization. *Journal of Water Resources Planning and Management*, 133(3):192–201, 2007.
- Labadie, J., Optimal operation of multireservoir systems: State-of-the-art review. *Journal of Water Resources Planning and Management*, 130(2):93–111, 2004.
- Reddy, M. & Kumar, D., Optimal reservoir operation using multi-objective evolutionary algorithm. *Water Resources Management*, 20(6):861–878, 2006.
- Sharma, V.; Jhab, R. & Naresh, R., Optimal multi-reservoir network control by two-phase neural network. *Electric Power Systems Research*, 68(3):221–228, 2004.
- Tolmasquim, M. & Guerreiro, A. (Eds.), *Balanço Energético Nacional 2009: Ano Base 2008*. Brasília, DF: Empresa de Pesquisa Energética – EPE, 2009.
- Yeh, W.G., Reservoir management and operations models: A state-of-the-art review. *Water Resources Research*, 21(12):1797–1818, 1985.

## Notas Biográficas

**Luís Antônio Scola** tem mestrado em Engenharia Mecânica (UFRJ, 1989) e é professor adjunto do Departamento de Ciências Térmicas e dos Fluidos da Universidade Federal de São João del Rei (MG). Atualmente é discente de doutorado no programa de pós-graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais.

**Oriane Magela Neto** tem mestrado em Engenharia Elétrica (UFMG, 1985), doutorado em Engenharia Elétrica (Imperial College London, 1996) e pós-doutorado (Tokyo Metropolitan University, 2002), é professor associado do Departamento de Engenharia Elétrica da Universidade Federal de Minas Gerais.

**Ricardo Hiroshi Caldeira Takahashi** tem mestrado em Engenharia Elétrica (UFMG, 1991), doutorado em Engenharia Elétrica (UNICAMP, 1998). É professor associado do Departamento de Matemática da Universidade Federal de Minas Gerais. Seu currículo no CNPq registra a autoria de mais de 60 artigos publicados ou aceitos para publicação em periódicos indexados, mais de 100 trabalhos publicados em anais de congressos, e ainda a orientação ou co-orientação de treze teses de doutorado e doze dissertações de mestrado já concluídas.

**Sérgio Augusto Araujo da Gama Cerqueira** tem mestrado em Engenharia Mecânica (UFMG, 1992), doutorado em Engenharia Mecânica (UNICAMP, 1999) e pós-doutorado (UFMG, 2006). É professor associado do Departamento de Engenharia Mecânica da Universidade Federal de São João del Rei (MG).



### Aplicação *On-line* de Algoritmo Genético na Otimização de um Regulador de Velocidade *Fuzzy* para Acionamentos Elétricos

Wander G. da Silva\*, Bernardo Alvarenga, Paul P. Acarnley e John W. Finch

---

**Resumo:** Este capítulo apresenta a aplicação da teoria de Algoritmo Genético para o ajuste automático dos parâmetros de um regulador de velocidade para acionamento elétrico baseado em Lógica Nebulosa, na presença de variação de carga. A complexidade do regulador de velocidade, aliado à característica não linear do controlador levaram ao ajuste de 16 parâmetros. Para a colocação de carga, o motor foi acoplado a um gerador alimentando um banco resistivo chaveado através de um IGBT. Como a tensão nos terminais do gerador depende da velocidade do motor, durante o processo de ajuste do regulador a carga do motor era variável, aumentando, assim, a dificuldade para o ajuste dos parâmetros do regulador de velocidade. Neste contexto, a aplicação do Algoritmo Genético foi realizada *on-line* sem qualquer intervenção humana. Os resultados obtidos mostraram que seu uso possibilitou a obtenção de um ajuste do regulador para um desempenho considerado ótimo, dentro do critério e espaço de solução previamente definidos para o problema.

**Palavras-chave:** Algoritmo genético, Otimização, Acionamento elétrico.

**Abstract:** *This chapter presents the theory of Genetic Algorithm applied to the automatic tuning of the parameters of a Fuzzy Logic speed controller for electric drives, in the presence of load variation. The complexity of the speed controller and its non-linear characteristics led to the tuning of 16 parameters. In order to apply load to the motor, it was coupled to a generator connected to a resistor bank, which was switched on and off through a single IGBT. The terminal voltage of the generator depends on the motor speed, which changed significantly during the tuning process, increasing the difficulty level of adjusting the controller parameters. Within this context, the use of Genetic Algorithm was carried out on-line without any human aid. The results showed its capability to find the optimum solution, according to the criteria and search space previously defined for the problem.*

**Keywords:** *Genetic algorithm, Optimization, Electric drives.*

---

\* Autor para contato: wander@eeec.ufg.br

## 1. Introdução

Em determinadas aplicações um controle preciso da velocidade do acionamento independente de distúrbios de carga presentes no processo é essencial para a garantia da qualidade do produto final. Em processos de laminação de chapas de aço e fabricação de filmes plásticos, por exemplo, variações na velocidade do acionamento elétrico comprometem a qualidade do produto final. Em uma planta industrial típica de fabricação de filmes plásticos existem aproximadamente 30 motores, na grande maioria de corrente contínua, trabalhando em conjunto e com velocidade constante. Assim, qualquer variação de velocidade em um dos motores é transmitida aos demais uma vez que são mecanicamente interconectados pelo processo (Ho et al., 1994). Para assegurar que a velocidade dos acionamentos seja mantida constante apesar das variações de carga é necessário que o regulador de velocidade responda rapidamente.

A grande maioria dos reguladores de velocidade encontrados no mercado é do tipo Proporcional-Integral (PI) cujos ganhos podem ser determinados com base na teoria clássica de controle. Entretanto, em função das não linearidades inerentes do sistema tais como saturação magnética, comutação, limitação de corrente e tensão das chaves eletrônicas que compõem o conversor, além do próprio distúrbio de carga, um ajuste ótimo para o regulador de velocidade nem sempre é uma tarefa fácil. Neste contexto, o conceito de “ótimo” pode variar em função dos critérios adotados, porém, em acionamento elétrico com regulação de velocidade, poderia ser, por exemplo, aquele ajuste do regulador capaz de levar o motor à velocidade de referência no menor espaço de tempo possível.

Com reguladores do tipo PI, para se limitar a ação do integrador e evitar os problemas de *windup*, por exemplo, pesquisadores propuseram diferentes circuitos *anti-windup* cujos parâmetros devem ser ajustados em conjunto com os ganhos do regulador PI (da Silva et al., 2001).

Na busca de soluções alternativas para problemas de controle de velocidade em acionamentos elétricos, diferentes técnicas foram desenvolvidas. Entre elas, encontra-se o uso da teoria de Lógica Nebulosa (*Fuzzy Logic*). Neste caso, e em função do grande número de parâmetros do controlador, um ajuste ótimo para o regulador de velocidade passou a ser uma tarefa muito difícil. Para a solução deste problema a teoria de Algoritmo Genético (AG) surgiu como uma poderosa ferramenta de busca e otimização capaz de possibilitar este ajuste. da Silva et al. (2000) propuseram o uso da teoria de AG para o ajuste *on-line* de um simples regulador de velocidade tipo PI, cujo resultado é comparado com aquele obtido utilizando-se a teoria clássica de controle linear. Naquele trabalho, o resultado encontrado através do uso de AG demonstrou melhor desempenho do que aquele obtido utilizando-se a teoria clássica de controle já consolidada pela literatura existente. Com o objetivo de se comparar o desempenho de diferentes con-

troladores do tipo PI associados a diferentes circuitos *anti-windup*, o AG foi também utilizado com sucesso (da Silva et al., 2001). Neste caso apenas três parâmetros foram ajustados para cada regulador, o que do ponto de vista da literatura, representa um desafio de pequena dificuldade para o AG. Porém, mesmo neste caso, um ajuste ótimo dos ganhos do regulador PI juntamente com os parâmetros do circuito *anti-windup* já deixa de ser uma tarefa simples, mesmo para um especialista.

Este capítulo apresenta o uso do AG para encontrar o melhor ajuste de um regulador de velocidade para acionamento elétrico baseado no princípio da lógica nebulosa, onde 16 parâmetros foram otimizados, *on-line*, sem qualquer intervenção humana. Resultados práticos são apresentados de forma a demonstrar a capacidade do AG em encontrar um ajuste ótimo para o regulador de velocidade diante das condições de partida e de variação de carga impostas.

## 2. Fundamentação Teórica

Algoritmo Genético pode ser definido como um processo estocástico de busca e otimização baseado no mecanismo de seleção natural onde o indivíduo melhor preparado ou com as melhores características genéticas possui maiores chances de sobrevivência (Gen & Cheng, 1997). O AG se inicia com um conjunto de possíveis soluções, normalmente definidas ou criadas de forma aleatória, identificado como população. Cada indivíduo da população é chamado de cromossomo, representando uma possível solução para um determinado problema. Os cromossomos são formados por um conjunto de parâmetros chamados de genes, normalmente, porém não necessariamente, codificados de forma binária (Chipperfield & Fleming, 1995). Estes cromossomos são evoluídos através de sucessivas gerações dentro das quais cada indivíduo ou cromossomo é avaliado de acordo com um determinado critério, representando assim sua capacidade para a solução do problema apresentado. Os melhores indivíduos (cromossomos) de cada população são selecionados para se cruzarem e gerar descendentes. Os menos preparados são então substituídos pelos descendentes daquela geração de forma a se manter constante o tamanho da população.

De acordo com o princípio da seleção natural, um determinado indivíduo pode sofrer mutações. Assim, em AG, mutações podem ser realizadas através de um operador de forma a produzir mudanças aleatórias nos cromossomos através da modificação de seus genes. Esta operação desempenha uma importante função no sentido de recuperar um gene perdido no processo de seleção ou simplesmente proporcionar o aparecimento de um gene que poderia não estar presente na população inicial. Dentro deste contexto, a taxa de mutação deve ser limitada em um percentual do número total de genes de uma determinada população de forma a controlar a

taxa com que novos genes serão introduzidos na população (Gen & Cheng, 1997).

Não existe na literatura um valor definido para cada problema a ser resolvido. Porém, uma elevada taxa de mutação pode fazer com que os descendentes (“filhos”) comecem a perder sua semelhança com seus ascendentes (“pais”). Desta forma, o AG poderia perder a capacidade de evolução a partir do histórico das gerações passadas. Por outro lado, se a taxa de mutação for muito baixa, muitos genes que poderiam ser importantes jamais seriam testados. Portanto, embora a taxa de mutação possa variar em função do problema ou do tamanho da população, algumas sugestões podem ser encontradas na literatura (Man et al., 1996). Do ponto de vista de convergência do processo de busca e otimização, esta operação impede que o algoritmo possa convergir para um ponto conhecido como “mínimo local” que pode não representar a melhor solução para o problema.

Um típico processo de AG como pode ser representado conforme segue (Man et al., 1996):

```
// Geração Inicial
g = 0;
// Início da população
Inicializar P(g);
// Avaliação da capacidade de cada indivíduo da população
Teste para critério de finalização do processo
(número de gerações ou outro critério qualquer)
Enquanto não concluído
// Selecionar sub-população para geração de descendentes
P'(g) = seleção de pais a partir de P(g)
// Recombinação de genes dos pais selecionados
recombinar P'(g);
// Perturbação da população cruzada estocasticamente
realizar mutação P'(g);
// Avaliar a capacidade de cada novo indivíduo
avaliar P'(g);
// Selecionar os sobreviventes
P(g+1) = sobreviventes P(g), P'(g);
// Incrementar a contagem de geração
g = g+1
fim
```

A literatura existente sobre AG é extensa e pode ser facilmente encontrada por aqueles que tiverem interesse em se aprofundar no assunto (Gen & Cheng, 1997; Man et al., 1996).

### 3. Modelo do Controlador de Velocidade do Motor CC

A forma tradicionalmente empregada para o controle de velocidade de um motor elétrico utiliza um regulador de velocidade em cascata com um regulador de corrente, conforme o diagrama de blocos ilustrado na Figura 1. Ambos os reguladores, tanto o de corrente quanto o de velocidade, são normalmente do tipo PI (da Silva et al., 2000).

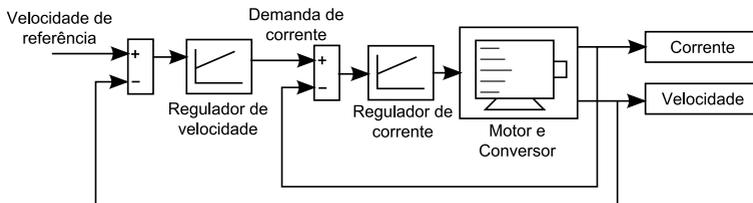


Figura 1. Representação de um acionamento elétrico clássico com controle de velocidade.

Em função das características do regulador tipo PI e das limitações de corrente e tensão das chaves do conversor, o ajuste dos ganhos proporcional e integral devem ser distintos para diferentes valores de degrau de referência de velocidade. Isto está evidenciado através de resultados obtidos utilizando-se a teoria clássica de controle e o uso do AG (da Silva et al., 2000). Além disto, um ajuste dos ganhos do regulador PI para variações de carga também não deverá ser o mesmo para qualquer condição. Quando se deseja um único ajuste do regulador PI de velocidade para diferentes valores de referência e condições de carga, um ajuste que satisfaça restrições de desempenho torna-se praticamente impossível e circuitos *anti-windup* muitas vezes são necessários para limitar a ação do integrador do regulador PI. A otimização dos ganhos do regulador PI de velocidade juntamente com seu circuitos foi investigada utilizando como ferramenta o AG e seus desempenhos comparados (da Silva et al., 2001).

Neste trabalho, o regulador PI de velocidade para um motor CC sem escovas (motor *Brushless* CC) foi substituído por outro baseado em lógica nebulosa, conforme ilustrado na Figura 2. O regulador de velocidade, neste caso, possui duas entradas e duas saídas. As entradas são o erro de velocidade e conjugado de carga, proveniente um observador de carga (Iwasaki & Matsui, 1993). As saídas são duas diferentes demandas de corrente para o regulador de corrente do tipo PI: (i) função do erro de velocidade e (ii) função da carga. É importante deixar claro neste momento que não se está analisando o desempenho do regulador, mas o emprego do AG para a determinação de parâmetros do regulador *fuzzy*.

Nesta forma de controle não linear é necessário o ajuste de diversos parâmetros. Mesmo sendo realizado por um especialista, não haveria ga-

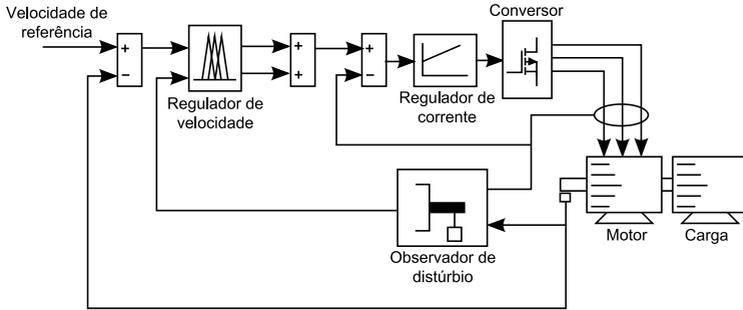


Figura 2. Diagrama de blocos de um acionamento elétrico com regulação de velocidade, baseado em lógica nebulosa.

rantia de que o melhor ajuste seria obtido, especialmente quando se deseja realizar o controle em diferentes condições de partida e distúrbios de carga em um só sistema/projeto.

Neste contexto, o AG se apresenta como uma poderosa ferramenta capaz de encontrar e otimizar os parâmetros do regulador de velocidade em questão. É importante destacar que o AG pode também ser utilizado para o ajuste automático (diretamente no sistema de acionamento) dos reguladores baseados na teoria clássica de controle (reguladores PI) onde apenas dois ou três parâmetros são ajustados (da Silva et al., 2000, 2001). Contudo, para o controlador *fuzzy* apresentado, é necessário um número muito maior de parâmetros, portanto, o desafio em se buscar um ajuste ótimo representa uma tarefa mais complexa para o algoritmo genético.

### 3.1 O controlador *fuzzy*

A teoria sobre lógica nebulosa não é objeto de discussão neste trabalho. Sugere-se consulta à literatura disponível (Bose, 1997; Ying, 2000; MathWorks, 2010). Pretende-se neste momento apresentar a construção do regulador *fuzzy* e sua característica de funcionamento.

O ponto de partida para a construção do regulador de velocidade baseado no princípio da lógica nebulosa é a definição das variáveis de entrada e saída e suas respectivas faixas de valores em termos de conjuntos nebulosos, através do uso de “funções de pertinência” (*Membership Functions* - MF) cuja quantidade e forma são definidas pelo usuário. Em seguida devem ser definidas as regras responsáveis pelo mapeamento das variáveis de entrada nas variáveis de saída. Pode-se observar então que o usuário possui um grande grau de liberdade na definição do regulador *fuzzy*. Assim, quanto maior o número de variáveis, funções de pertinência e número de regras, maior se torna o grau de complexidade e mais difícil o ajuste

deste regulador. Consequentemente, mesmo aquele considerado especialista encontra grandes dificuldades em ajustar o regulador (MathWorks, 2010), especialmente para diferentes condições de funcionamento.

Cada variável de entrada e saída do controlador *fuzzy* proposto possui duas funções de pertinência que foram definidas como sendo do tipo trapezoidal, conforme ilustrado na Figura 3, de forma normalizada, com um ponto de cruzamento entre si.

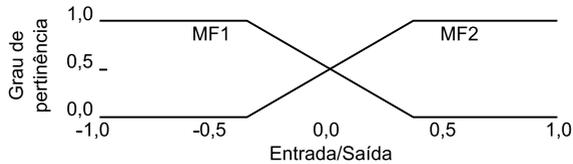


Figura 3. Funções de pertinência do regulador *fuzzy*.

As regras do regulador *fuzzy* foram definidas da seguinte maneira:

For  $i, j = 1, 2$

**IF** input( $i$ ) **is** MF( $i$ ) **then** Output( $j$ ) **is** MF ( $j$ )

O método de “*defuzzificação*” para a determinação do valor de saída de cada variável do controlador foi o cálculo do centro da área da figura (centróide) (MathWorks, 2010). Para este controlador, apenas dois pontos de cada função de pertinência precisam ser ajustados uma vez que são fundamentais na determinação do desempenho do regulador. Considerando então duas funções de pertinência para cada variável (total de 4 variáveis) com dois pontos a serem ajustados para cada função de pertinência, foram necessários o ajuste de um total de 16 parâmetros. Os pontos ajustados são ilustrados na Figura 4.

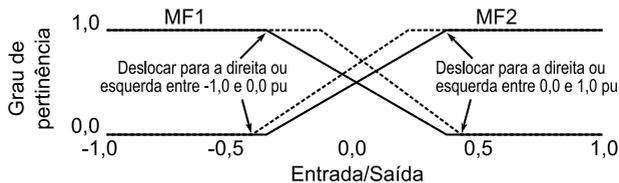


Figura 4. Ilustração de como as funções de pertinência devem ser ajustadas.

O regulador apresentado possui um número mínimo de funções de pertinência e de regras. Desta forma, o que precisou ser ajustado foram simples-

mente dois pontos de cada função de pertinência para se obter o máximo de desempenho do regulador de velocidade. Para isto foi utilizado o AG.

### 3.2 Aplicação de algoritmo genético no regulador fuzzy de velocidade

Alguns trabalhos utilizando o AG aplicado à Lógica Nebulosa já foram desenvolvidos (Meredith et al., 1993; Lee & Takagi, 1993; Homaifar & McCormick, 1995; Tang et al., 1998; da Silva et al., 2004). No trabalho aqui apresentado o AG foi utilizado para fazer o ajuste *on-line* das funções de pertinência do regulador de velocidade *fuzzy*, isto é, diretamente no sistema de acionamento, sem qualquer intervenção humana. Inicialmente foram definidas as faixas de valores admissíveis para cada função de pertinência. O sistema de controle foi desenvolvido de forma que o motor operasse entre 0 e 200 rad/s. O erro de velocidade precisou portanto, ter como limite, o módulo do valor máximo de rotação em radianos por segundo (200 rad/s). A faixa de valores definida foi, portanto, de  $-200$  a  $200$  rad/s, o que representa valores de  $-1,0$  a  $1,0$  pu. Em função da capacidade do motor, a faixa de valores para variação de carga em termos de corrente foi de  $-22$  a  $22$  A. Para a variável de saída especificada como demanda de corrente para erro de velocidade, a faixa de valores foi também de  $-22$  a  $22$  A, enquanto que para a variável demanda de corrente para variações de carga, a faixa foi de  $-15$  a  $15$  A. Desta forma, em função das características do regulador *fuzzy*, o somatório das correntes de saída não ultrapassariam 9 A de corrente nominal de armadura do motor.

Os cromossomos ou indivíduos criados no âmbito do AG devem possuir 16 genes, cada um representando um ponto a ser ajustado em cada função de pertinência definido conforme a seguir:

$$\text{Crom.} = \left[ \begin{array}{c|c|c|c} a_1 \dots a_4 & b_1 \dots b_4 & c_1 \dots c_4 & d_1 \dots d_4 \\ \hline \text{Entrada 1} & \text{Entrada 2} & \text{Saída 1} & \text{Saída 2} \end{array} \right]$$

Em seguida foi necessária a definição de como cada indivíduo deveria ser avaliado em relação à sua capacidade de proporcionar a resposta de velocidade considerada ideal. Dentro deste contexto, foi considerado como melhor preparado para a solução do problema aquele indivíduo ou cromossomo que atendesse a duas condições: (i) levar o motor à velocidade de referência a partir de um degrau de entrada no menor espaço de tempo e (ii) manter constante a rotação do motor independentemente de variação da carga conectada em seu eixo. Assim, a Integral do Valor Absoluto do Erro de Velocidade (IAE) foi utilizada como critério para a definição da capacidade do indivíduo.

## 4. Metodologia

Para a realização do presente trabalho, foi utilizado um sistema de acionamento elétrico para um motor do tipo *Brushless* CC. O sistema emprega um DSP com lógica de programação desenvolvida no ambiente MATLAB/SIMULINK. Uma interface de software possibilitou a comunicação direta entre o MATLAB e o DSP de forma a se escrever os parâmetros do controlador no DSP e obter leituras de velocidade e corrente, através de placas de aquisição de dados (French et al., 1998). O diagrama de blocos da Figura 5 ilustra o sistema utilizado.

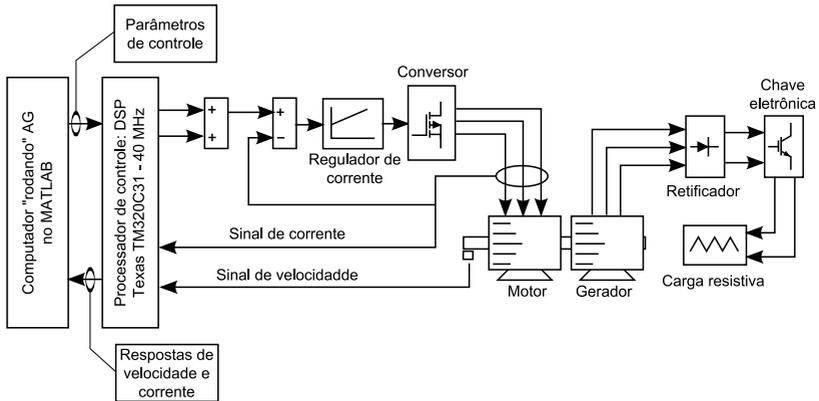


Figura 5. Diagrama de blocos do sistema realizado.

Foi inicialmente criada aleatoriamente uma população de 20 indivíduos para evoluir por 20 gerações. Cada indivíduo foi avaliado de forma a produzir uma resposta ao degrau de 1100 rpm de referência de velocidade com uma carga inicial de 11 % de seu valor nominal. Em  $t = 1$  s foi aplicado um distúrbio de carga na forma de degrau, elevando a carga aplicada para 62 % da nominal, permanecendo até o instante  $t = 4$  s. A capacidade do indivíduo de produzir uma resposta satisfatória foi analisada em função da Integral do valor Absoluto do Erro de Velocidade (IAE) em relação ao tempo. Portanto, quanto menor o valor de IAE melhor a resposta proporcionada pelo indivíduo, isto é, melhor a capacidade do indivíduo para solucionar o problema apresentado. Portanto, o AG deveria encontrar um ajuste para o regulador de velocidade *fuzzy* que levasse o motor à velocidade de referência no menor tempo possível e ali mantê-la independente do distúrbio de carga.

A bancada foi constituída de um conjunto Bosch Servodyn Servo Motor SE-B4 comercial, com estágio de entrada constituído de um retificador

trifásico totalmente controlado VM60-T, um estágio DC composto de um módulo de capacitor modelo KM1100T 1100  $\mu\text{F}$  - 840 V com saída regulada em 600V e estágio de saída composto de um inversor SM50/100 com regulador de corrente do tipo proporcional-integral. O motor possui os seguintes parâmetros:

- Corrente nominal: 9 A;
- Velocidade nominal: 3000 rpm;
- Constante de torque: 0,91 Nm/A;
- Indutância da armadura: 9,25 mH;
- Resistência da armadura: 0,74  $\Omega$ ;
- Constante de FEM: 95,7 V/1000 rpm;
- Constante de FEM do tacogerador: 2,7 V/1000 rpm  $\pm 5\%$ ; em  $t_{amb} = 20^\circ\text{C}$ ;
- Inércia motor/carga: 0,016 Nm  $\cdot$  s<sup>2</sup>/rad;
- Coeficiente de atrito viscoso: 0,0092 Nm/rad/s;
- Tempo de amostragem para sinais de entrada e saída: 1,25 ms.

Para fins de colocação de carga o motor foi acoplado a um gerador alimentando um banco resistivo. A tensão trifásica gerada foi retificada através de um retificador a diodos em ponte SEMIKRON 30/04 A1. Na saída do retificador foi colocada uma chave eletrônica SEMIKRON IGBT SKM 75GAR063D, através da qual foi possível aplicar degrau de carga no motor através do chaveamento do banco de resistores. A chave eletrônica IGBT foi ligada/desligada através de um dispositivo analógico de entrada SEMIKRON SKHI10. O sinal de entrada para o módulo SEMIKRON foi proveniente de um conversor D/A de 12 bits de resolução, utilizado como interface com o DSP.

## 5. Resultados Experimentais

Uma vez iniciado o AG, não houve mais a intervenção humana até sua conclusão, que se deu em função do número de gerações inicialmente definido. Para efeitos de verificação, a Figura 6 ilustra algumas respostas de velocidade obtidas na primeira geração. A evolução do AG medida em termos da IAE é ilustrada na Figura 7. Observa-se que a melhor resposta de velocidade obtida com um dos indivíduos da primeira geração proporcionou uma IAE de 40,6 rad e que, ao término do processo, o valor da IAE foi reduzido para 26,9 rad. A Figura 8 ilustra a resposta de velocidade e de corrente do motor obtida com o regulador *fuzzy* cujo ajuste é dado pelo AG ao final de 20 gerações, critério utilizado para a conclusão do processo.

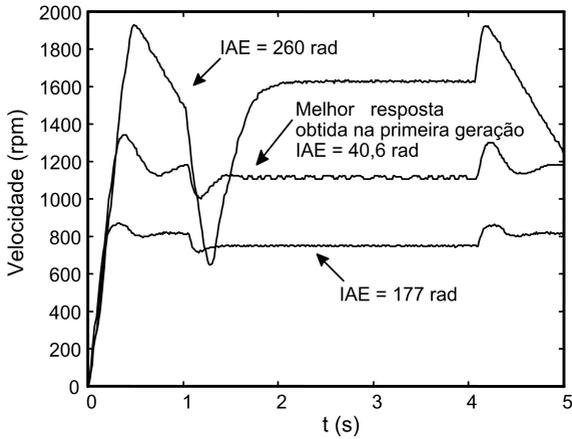


Figura 6. Algumas respostas de velocidade obtidas com o controlador de velocidade *fuzzy* dada pelo AG na primeira geração.

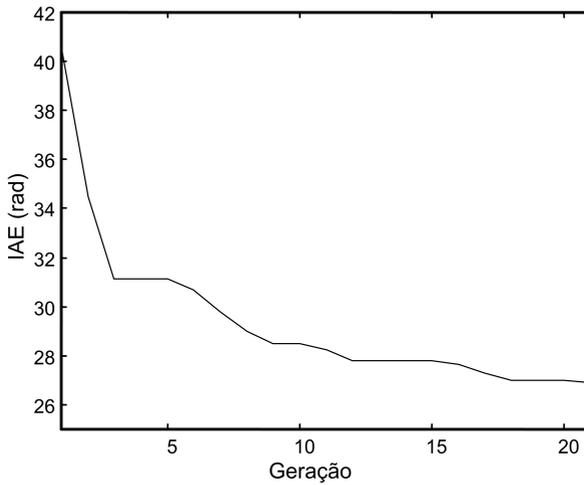


Figura 7. Evolução do processo de otimização.

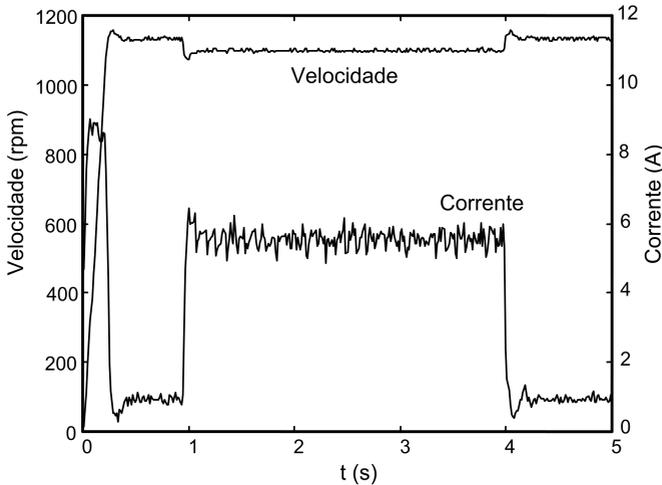


Figura 8. Melhor resposta de velocidade obtida após a evolução de uma população de 20 indivíduos por 20 gerações.

É importante registrar que a tensão aplicada no banco de resistores utilizado como carga para o gerador acoplado ao motor varia em função da rotação. Isto representa uma dificuldade a mais imposta ao AG uma vez que a carga mecânica efetivamente conectada ao motor varia para cada indivíduo ou cromossomo. Apesar desta dificuldade, o AG conseguiu encontrar um ajuste para o controlador *fuzzy* dentro das condições de velocidade de referência e carga impostas ao motor.

Como pode ser observado, tanto na Figura 7 quanto na Figura 8, em função da resposta de velocidade obtida, o AG ainda poderia evoluir se tivesse sido permitida a evolução por maior número de gerações. Em função disto, com o objetivo de se chegar mais rapidamente a uma solução ótima, o tamanho da população inicial foi aumentado para 40 indivíduos e o espaço de busca reduzido a valores próximos ao resultado obtido com uma população de 20 cromossomos: erro de velocidade reduzido em 80 % e conjugado de carga reduzida em 60 %. Permaneceram constantes a velocidade de referência e o distúrbio de carga. A Figura 9 ilustra a evolução de 40 indivíduos por 20 gerações e a Figura 10 apresenta a resposta de velocidade e corrente obtida após a conclusão do processo, isto é, a melhor resposta obtida pelo controlador *fuzzy* ajustado pelo AG. Observa-se que a IAE obtida com o melhor indivíduo da primeira geração foi de aproximadamente 21,8 rad enquanto que aquela obtida ao final do processo de

evolução foi de 16,5 rad, ilustrando que a evolução foi muito significativa apesar do tamanho da população e do espaço de procura reduzido.

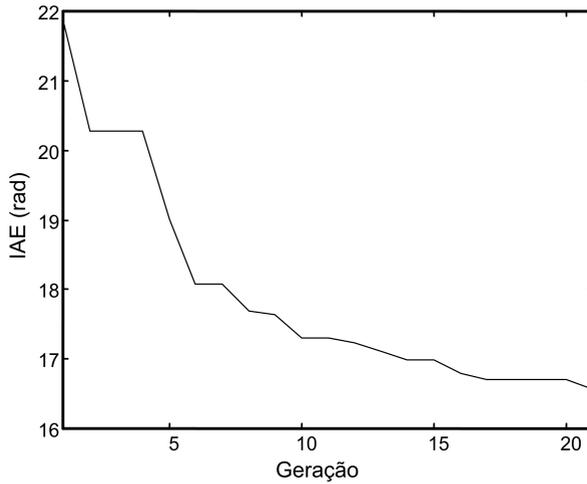


Figura 9. Evolução de 40 indivíduos por 20 gerações com espaço de busca reduzido.

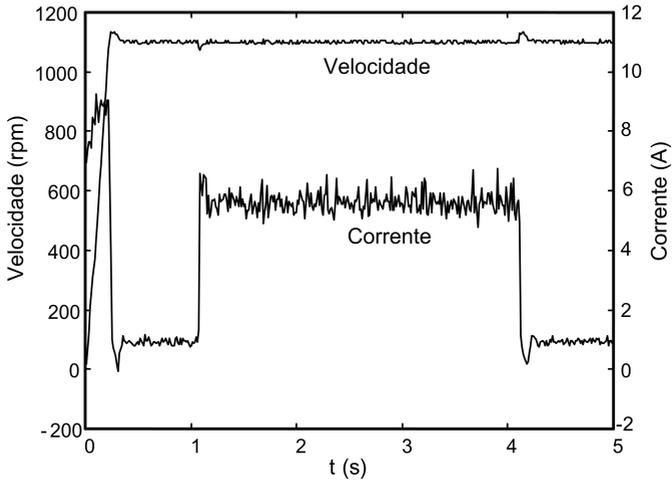


Figura 10. Melhor resposta de velocidade para o regulador *fuzzy* após evolução de 40 indivíduos por 20 gerações.

É importante registrar que no processo de busca e otimização por AG, cada indivíduo deve ser avaliado individualmente. Portanto, neste processo, cada resposta de velocidade ou seja, cada indivíduo, foi avaliado por um período de 5 s. Somado aos tempos de leitura e escrita de dados no DSP e processamento da informação, o tempo total necessário para a evolução de 40 cromossomos por 20 gerações foi de aproximadamente 3 h e 46 min. Duplicando-se o número de gerações, o tempo total necessário para a conclusão do processo poderia chegar a aproximadamente 8 h o que, em certas aplicações, pode ser considerado um tempo excessivamente longo. Para a redução do tempo necessário para a otimização, uma possível solução seria realizar o processo através de simulação e em seguida, a partir dos resultados obtidos, restringir o espaço de busca para o processo a ser realizado *on-line*.

## 6. Conclusões

Este capítulo apresentou o uso do AG aplicado à Lógica Nebulosa. Para o controlador de velocidade baseado em lógica nebulosa desenvolvido, trabalhando com número mínimo de funções de pertinência e de regras, ainda foram necessários o ajuste de 16 parâmetros relacionados às funções de pertinência. Esta tarefa não é simples, mesmo para um especialista, especialmente na presença de variações de carga. Para o ajuste do controlador foi inicialmente criada uma população de 20 indivíduos ou cromossomos, que foi evoluída por 20 gerações. O espaço de busca imposto foi relativamente grande com o objetivo de se verificar a capacidade do AG em encontrar a melhor resposta para o problema – levar o motor à sua velocidade de referência e manter sua velocidade mesmo na presença de variação de carga. Observou-se que o AG convergiu em direção à melhor resposta, porém, em função da dimensão reduzida da população inicialmente criada, bem como do número de gerações, a resposta final encontrada poderia ainda ser melhorada, seja aumentando o número de gerações, aumentando a dimensão da população, ou reduzindo o espaço de busca a partir dos resultados previamente obtidos. Aumentando-se o tamanho da população e reduzindo o espaço de busca, o AG conseguiu, após 20 gerações, um ajuste que levou o motor à velocidade de referência com a maior aceleração possível. O ajuste do regulador de velocidade *fuzzy* encontrado pelo AG fez com que o motor partisse e acelerasse com corrente praticamente nominal. Nota-se que, na presença de um degrau de carga de 62 % de seu valor nominal, praticamente não se vê variação na velocidade do motor, o que significa que o controle ajustado pelo AG conseguiu responder rapidamente à variação de carga, mantendo constante sua velocidade. Como em AG todos os cromossomos ou indivíduos devem ser avaliados, dependendo do problema a ser resolvido, da dimensão da população, do número de parâmetros, do espaço de busca e do número de gerações, o tempo necessário

para se chegar ao resultado considerado ótimo pode ser longo em algumas aplicações. Para o problema apresentado, considerando uma população de 40 cromossomos, cada um com 16 parâmetros e evoluído por 20 gerações, o tempo necessário para se chegar ao resultado final foi de 3 h e 46 min. Em certas aplicações este tempo pode ser considerado longo. Porém, uma importante observação é que o AG conseguiu encontrar o melhor ajuste do controlador de velocidade em um processo totalmente automatizado, sem qualquer intervenção humana e de certa forma, de auto-aprendizagem uma vez que, à medida que o AG evoluía, todos os cromossomos se convergiam em direção àquele que possuía as melhores características para a solução do problema.

## Referências

- Bose, B., *Power Electronics and Variable Frequency Drives: Technology and Applications*. 1a edição. New York, USA: IEEE Press, 1997.
- Chipperfield, A. & Fleming, P., The MATLAB genetic algorithm toolbox. In: *Proceedings of Colloquium on Applied Control Techniques Using MATLAB*. London, UK: IEE, p. 1-4, 1995.
- French, C.D.; Finch, J.W. & Acarnley, P.P., Rapid prototyping of a real-time DSP based motor drive controller using SIMULINK. In: *Proceedings of International Conference on Simulation*. York, UK: IEE, p. 284-291, 1998.
- Gen, M. & Cheng, R., *Genetic algorithms and engineering design*. New York, USA: John Wiley & Sons, 1997.
- Ho, S.; Jones, A. & Cox, C.S., Torque disturbances and their effect on process control. In: *Proceedings of the 5<sup>th</sup> International Conference on Power Electronics and Variable Speed Drives*. London, UK: IEE, p. 602-607, 1994.
- Homaifar, A. & McCormick, E., Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, 3(2):129-139, 1995.
- Iwasaki, M. & Matsui, N., Robust speed control of IM with torque feedforward control. *IEEE Transactions on Industrial Electronics*, 40(6):627-632, 1993.
- Lee, M. & Takagi, H., Integrating design stages of fuzzy systems using genetic algorithms. In: *Proceedings of the 2<sup>nd</sup> IEEE International Conference on Fuzzy Systems*. San Francisco, USA: IEEE, p. 612-617, 1993.
- Man, K.; Tang, K. & Kwong, S., Genetic algorithms: concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5):519-533, 1996.

- MathWorks, , *Fuzzy Logic Toolbox User's Guide*. 2a edição. Natick, USA: The MathWorks, Inc, 2010.
- Meredith, L.; Karr, C. & Kumar, K., The use of genetic algorithms in design of fuzzy logic controllers. In: *Proceedings of the 3<sup>rd</sup> Workshop on Neural Networks*. Bellingham, USA: International Society for Optical Engineering – SPIE, p. 549–555, 1993.
- da Silva, W.; Acarnley, P. & Finch, J., Anti-windup circuits with on-line optimisation by genetic algorithm. In: *Proceedings of the 9<sup>th</sup> European Conference on Power Electronics and Applications*. Graz, Austria: EPE Association, p. 1–11, 2001.
- da Silva, W.G.; Acarnley, P.P. & Finch, J.W., Application of genetic algorithms to the on-line tuning of electric drive speed controllers. *IEEE Transactions on Industry Applications*, 47(1):217–219, 2000.
- da Silva, W.G.; Acarnley, P.P. & Finch, J.W., On-line optimisation of a fuzzy drive controller using genetic algorithm. In: *Proceedings of IEEE International Symposium on Industrial Electronics - ISIE*. Ajaccio, France: IEEE Press, p. 1441–1446, 2004.
- Tang, K.; Man, K.; Liu, Z. & Kwong, S., Minimal fuzzy membership and rules using hierarchical genetic algorithm. *IEEE Transactions on Industrial Electronics*, 45(1):162–169, 1998.
- Ying, H., *Fuzzy Control and Modeling: Analytical Foundations and Applications*. 1a edição. New York, USA: Wiley-IEEE Press, 2000.

## Notas Biográficas

**Wander Gonçalves da Silva** concluiu graduação em Engenharia Elétrica pelo Instituto Superior de Ensino e Pesquisa de Ituiutaba - ISEPI em junho de 1986 e em agosto do mesmo ano tornou-se professor no curso de Engenharia Elétrica da mesma instituição. Em julho de 1989 tornou-se Mestre em Engenharia Elétrica pela Universidade Federal de Santa Catarina - UFSC. Em maio de 1999 tornou-se PhD pela Universidade de Newcastle, Inglaterra. Em 2009 juntou-se ao quadro de professores da Escola de Engenharia Elétrica e de Computação - EEEEC da Universidade Federal de Goiás - UFG. Possui publicações científicas relevantes na área de acionamento de máquinas elétricas e se interessa também pelo desenvolvimento de pesquisas relacionadas à qualidade da energia elétrica e sistemas de controle.

**Bernardo Alvarenga** graduou-se em Engenharia Elétrica pela Universidade de Brasília (1990) e recebeu os títulos de Mestre e Doutor em Engenharia Elétrica pela UFU (1993) e pela USP (2004), respectivamente. Atualmente é professor e pesquisador do Programa de Mestrado em Engenharia Elétrica e de Computação da EEEEC-UFG. Suas áreas de interesse incluem máquinas elétricas e modelagem e simulação de dispositivos eletromagnéticos.

**Paul P. Acarnley** formou-se em Engenharia Elétrica e recebeu o título de PhD pela Universidade de Leeds, Inglaterra, em 1974 e 1977, respectivamente, e o título de MA da Universidade de Cambridge, em 1978. Em 1986, juntou-se ao grupo de Eletrônica de Potência, Acionamento Elétrico e Máquinas da Universidade de Newcastle, Inglaterra. Em 2003, fundou o RESEEDS (*Research Engineering Education Services*). É também Diretor do *Electrical Energy Engineering (E3) Academy*. Tem interesse em pesquisa na área de acionamentos elétricos, incluindo estimação de estados e de parâmetros. É ainda *Fellow* do *Institution of Electrical Engineers*.

**John W. Finch** formou-se em Engenharia Elétrica pelo University College de Londres e recebeu o título de PhD pela Universidade de Leeds, Inglaterra. Foi Diretor Associado do *Resource Centre for Innovation and Design*. É autor e co-autor de mais de 100 publicações em controle aplicado, simulação, máquinas elétricas e acionamentos. É professor emérito em Engenharia de Controle na Universidade de Newcastle, Inglaterra.



### Comparação de Técnicas de Computação Evolucionária para o Projeto de Filtros Digitais de Resposta Finita ao Impulso

Daniel R. de Oliveira, Marlon H. Scalabrin, Fábio R. Teodoro,  
Rafael S. Parpinelli e Heitor S. Lopes \*

---

**Resumo:** O projeto de filtros digitais FIR é um problema clássico na engenharia. Existem diversos métodos e algoritmos para isto, mas a maior parte é limitada a poucos tipos de filtro, ou oferece pouco controle sobre as especificações. Por isso, a aplicação de algoritmos de computação evolucionária para este problema é tema recorrente na literatura. Este trabalho propõe uma nova função de *fitness* para esta classe de problema, mais adequada do que o *Parks-McClellan*. Além disto, é comparado o desempenho dos algoritmos AG, PSO e HS para o projeto de quatro tipos de filtros digitais. Experimentos mostraram bons resultados com a nova abordagem. De maneira geral, o PSO foi o algoritmo que conseguiu obter os melhores resultados.

**Palavras-chave:** Filtros digitais, Algoritmo genético, Otimização por enxame de partículas, Busca harmônica.

**Abstract:** *The project of FIR digital filters is a classical problem in Engineering. There are a number of methods and algorithms addressing this area, but most are limited to one kind of filter, or offers little control over specifications. Hence, utilization of Evolutionary Algorithms for this problem is recurrent in literature. This work proposes a new fitness function for this class of problem, better than the Parks-McClellan. Besides, the performance of GA, PSO and HS were compared for the project of four digital filters. Experiments showed good results with the new approach. In general, PSO was the better performing algorithm.*

**Keywords:** *Digital filters, Genetic algorithm, Particle swarm optimization, Harmony search.*

---

\* Autor para contato: [hslopes@utfpr.edu.br](mailto:hslopes@utfpr.edu.br)

## 1. Introdução

O projeto de filtros digitais, tanto do tipo FIR (*Finite Impulse Response*) quanto IIR (*Infinite Impulse Response*), é um problema clássico na Engenharia Elétrica (Hamming, 1997). A literatura reporta diversos métodos para o cálculo dos coeficientes que caracterizam a resposta em frequência do filtro, como amostragem em frequência e diversas abordagens por janelamento, cada um com suas características próprias, vantagens e desvantagens. (Diniz et al., 2004).

A maioria dos métodos de projeto é direcionada para filtros do tipo passa-altas ou passa-baixas, embora alguns também possam ser utilizados para filtros passa-faixas ou rejeita-faixas. Porém, poucos métodos permitem um controle maior das especificações, ou maior número de faixas. Na literatura recente, por exemplo, são raros os métodos que permitem especificar um filtro com duas bandas passantes e três de rejeição, cada uma com requisitos de *ripple* (oscilação) e ganho diferentes (Zhang et al., 2008; Ahmed, 2003). Por esta razão, métodos heurísticos de otimização, notadamente aqueles da computação evolucionária, podem ter uma grande aplicabilidade prática neste problema de engenharia.

De fato, há alguns trabalhos utilizando algoritmos genéticos (AG) (Goldberg, 1989) e otimização por enxame de partículas (*Particle Swarm Optimization* – PSO) (Kennedy & Eberhart, 2001) que mostraram bons resultados, geralmente obtendo soluções melhores do que aquelas alcançadas pelos métodos tradicionais (Barros, 2006). Porém, a grande maioria de tais trabalhos focaliza apenas o projeto de filtros passa-baixas ou passa-altas, que são os problemas mais simples de serem resolvidos. É muito raro encontrar projeto de filtros mais complexos, com especificação de várias bandas, como já mencionado. Outra questão muito importante é a forma de avaliação da qualidade dos filtros digitais, representado pela função de *fitness* num algoritmo de computação evolucionária. Todos os artigos que os autores encontraram na literatura recente minimizam o Erro Médio Quadrático (*Mean Square Error* – MSE) entre a resposta em frequência do filtro calculado e uma curva-alvo de resposta desejada (Najjarzadeh & Ayatollahi, 2008; Barros et al., 2007; Huang et al., 2004). Embora esta abordagem geralmente leve a resultados próximos do alvo desejado, ela tem algumas desvantagens.

Primeiramente, é necessário montar uma curva-alvo. Porém, o mais intuitivo e usual é o usuário fornecer as especificações do filtro sob a forma de ganho nas bandas passante e de rejeição, bem como as frequências que compõem essas faixas. Utilizar a curva ideal, expressa pelo MSE, pode levar o algoritmo a buscar um filtro muito longe do factível, tendo como resultado filtros ruins.

Outra desvantagem da abordagem tradicional de tentar igualar uma curva-alvo é a dificuldade de se obter filtros dentro das especificações, mas

com resposta diferente, na tentativa de igualar o *ripple*. Por exemplo, se a curva-alvo tiver 8 ondulações na banda de rejeição e um filtro projetado tiver 6, esse filtro terá um *fitness* mais baixo, mesmo que seja uma solução melhor do que o primeiro.

Assim, o primeiro objetivo deste trabalho é buscar uma forma mais adequada de medir o *fitness* em projetos de filtros digitais utilizando métodos de computação evolucionária. Objetiva-se encontrar uma função de *fitness* que privilegie o atendimento das especificações da melhor forma possível, e que permita encontrar filtros ainda melhores do que a especificação. Na literatura recente registra o uso de AG e PSO para projeto de filtros digitais. Assim, o segundo objetivo é comparar o desempenho destes dois algoritmos mais a busca harmônica (*harmony search* – HS) (Geem et al., 2001) para o projeto de filtros digitais complexos.

Este trabalho está dividido da seguinte maneira: na Seção 2 é apresentada a teoria básica dos filtros digitais, bem como alguns dos métodos clássicos utilizados no seu projeto. Nesta seção também são apresentados alguns aspectos sobre a medição de qualidade de filtros pelo MSE e uma visão geral sobre Computação Evolucionária e os algoritmos que serão utilizados. Na Seção 3 é discutida a metodologia do trabalho, a função de *fitness* proposta e os experimentos realizados, com seus parâmetros. Na Seção 4 são mostrados e discutidos os resultados dos experimentos realizados. Finalmente, na Seção 5 são apresentadas as conclusões do trabalho.

## 2. Fundamentação Teórica

### 2.1 Filtros digitais FIR

Um filtro é um sistema que realiza operações matemáticas em um determinado sinal, com a finalidade de reduzir ou amplificar certos aspectos deste sinal (Oppenheim & Schaffer, 1975). Um exemplo simples da aplicação de um filtro é na separação das faixas grave, média e aguda de um sinal de áudio, direcionando cada uma a um alto-falante mais adequado para sua reprodução. Um filtro também pode ser projetado com o fim de eliminar uma determinada característica, como por exemplo, o ruído decorrente da rede elétrica (de frequência constante) inserido em algum sinal que carregue outra informação.

Filtros digitais são sistemas lineares invariantes no tempo discreto (Diniz et al., 2004). Estes diferem dos filtros analógicos em vários aspectos, notadamente na forma do sinal de entrada, que passa do domínio contínuo para o domínio discreto, e na forma de implementação. Dentre os filtros digitais, há os do tipo não-recursivo, chamados assim por não terem a saída a cada instante afetada por saídas anteriores. Estes filtros são caracterizados por uma equação a diferenças que segue o formato mostrado na Equação 1.

$$y(n) = \sum_{k=0}^M \vec{b}_t x(n-k) \quad (1)$$

onde  $y(n)$  é o valor da saída do sistema no instante  $n$ ,  $x(n-k)$  é o valor de entrada  $k$  amostras atrás e  $\vec{b}$  é o vetor de coeficientes da equação a diferenças. Estes coeficientes estão diretamente ligados à resposta ao impulso unitário do sistema,  $b_t = h(t)$ , onde  $h$  é a função de transferência do sistema. Como a saída depende apenas das  $M$  entradas anteriores, a resposta ao impulso deste tipo de filtro tem duração  $M$ , que é finita. Portanto, este filtro é chamado de filtro com Resposta Finita ao Impulso (*Finite Impulse Response* – FIR). Existem diversas formas de implementação deste tipo de filtro, como a forma direta ou em cascata, no entanto este assunto está fora do escopo deste trabalho.

A resposta no domínio da frequência de um sistema linear é dada pela transformada de Fourier de sua resposta ao impulso no domínio do tempo. Visto que a resposta de um filtro FIR é igual ao vetor dos seus coeficientes, basta aplicar a transformada de Fourier discreta (DFT) neste vetor para obter a resposta do filtro em questão.

## 2.2 Métodos de cálculo de coeficientes para filtros FIR

Existem diversas abordagens para o cálculo dos coeficientes de um filtro FIR, cada uma com suas vantagens e desvantagens. O método mais simples é o da amostragem em frequência (Diniz et al., 2004) que consiste em estabelecer a resposta em frequência desejada e realizar a IDFT (transformada discreta de Fourier inversa). Este procedimento é válido pois garante que em cada ponto amostrado a resposta seja exatamente a desejada. Porém, ela não garante nada nas frequências intermediárias. Por exemplo, se o filtro desejado possuir 40 coeficientes, o método garante que nas 40 frequências uniformemente distribuídas entre 0 e  $\frac{Fs}{2}$  (onde  $F_s$  é a frequência de amostragem) a resposta seja exata. Mas entre cada uma destas 40 frequências, não há nenhum controle. Por essa razão, esta abordagem tem aplicabilidade limitada.

Os métodos de projeto mais utilizados são os que utilizam funções do tipo janela, cuja finalidade é corrigir os coeficientes obtidos pelo método da IDFT. Elas são chamadas desta forma por serem caracterizadas pelo truncamento da resposta ideal de um filtro, que é infinita, através da multiplicação dela por uma função-janela, que é projetada de forma a distorcer a resposta ideal o mínimo possível. O simples truncamento desta resposta é equivalente a multiplicar a função por uma janela retangular. A desvantagem deste método é apresentar ondulações grandes perto das extremidades de cada faixa do filtro. (Hamming, 1997).

Uma das formas de se reduzir estas ondulações é a utilização de uma janela triangular. Ou seja, uma operação que não só trunca a resposta do

filtro, mas também atenua as extremidades da mesma. Porém, em alguns casos é necessário uma redução ainda maior das ondulações, onde serão utilizadas funções-janela mais complexas, como as de Hamming, Hanning, Blackman, Kaiser e Dolph-Chebyshev, entre outras (Oppenheim & Schaffer, 1999; Barros, 2006).

Um outro método é o de Parks-McClellan (McClellan & Parks, 1973), que permite especificações de ganho por banda, enquanto outros métodos de cálculo em geral permitem apenas a escolha das frequências de corte do filtro. Uma das características dos filtros calculados por este método é um *ripple* constante tanto na banda de passagem quanto na banda de rejeição, ou seja, são filtros *equiripple*.

### 2.3 Cálculo do *fitness* de filtros FIR como procedimento de otimização

Como mencionado na Seção 1, a forma mais usual de se calcular o *fitness* de uma solução candidata no problema do cálculo de filtros FIR é a minimização do MSE. Neste método utiliza-se uma resposta-alvo para o algoritmo, que tentará calcular um filtro que tenha esta exata resposta em frequência. Porém, isto leva a dois problemas. Primeiramente, essa abordagem limita a exploração do espaço de busca. Quando um filtro é especificado, não se escolhe o exato formato de sua resposta, e sim alguns parâmetros-chave. Os mais comuns são: as frequências que compõem as bandas de passagem e rejeição, o *ripple* máximo na banda de passagem e a atenuação mínima na banda de rejeição. Para cada conjunto de especificações, pode existir um número infinito de filtros que as atendam, mas essa abordagem de curva-alvo limita a busca a apenas uma. Se, por um lado, isso conduz a evolução da resposta de forma mais rápida e/ou robusta, por outro lado são excluídas soluções que atenderiam perfeitamente as especificações, algumas de forma mais satisfatória do que a curva-alvo, por serem diferentes da mesma. Esta característica desta abordagem é exemplificada nas Figuras 1 e 2.

Na Figura 1, o filtro obtido é nitidamente melhor do que a curva-alvo. As características na banda de passagem são praticamente idênticas, mas a atenuação na banda de rejeição é cerca de 5dB maior.

Na Figura 2, o resultado obtido é ligeiramente melhor do que a curva-alvo, com uma atenuação grande nas extremidades da banda de rejeição, mas é pior do que a obtida pelo filtro da Figura 1, quando próximo da banda de passagem.

Nas Figuras 1 e 2 é possível notar também que a curva-alvo é do tipo *equiripple* (Williams & Taylors, 1995), isto é, tem uma ondulação constante na banda de rejeição. O filtro obtido na Figura 1 também obedece a esta característica, pois a resposta foi melhor do que a da Figura 2. Porém, pelo MSE, o primeiro filtro seria preterido pelo segundo. Isto ocorre porque o número de ondulações na banda de rejeição do segundo filtro foi o mesmo da curva-alvo, tornando essa resposta mais próxima se avaliada ponto a

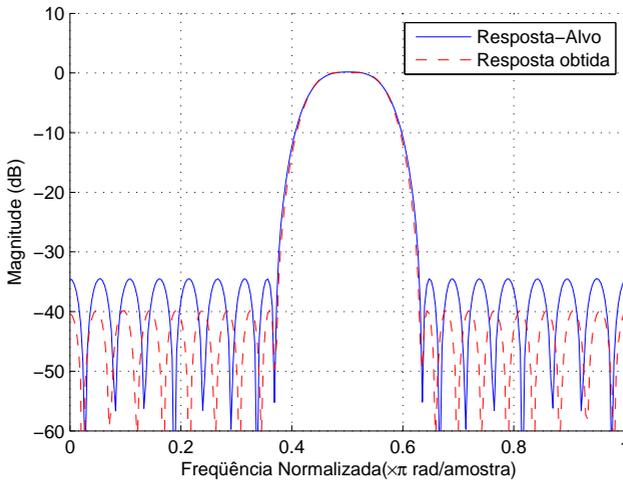


Figura 1. Exemplo 1 de curva-alvo e resposta obtida.

ponto, em vez de qualitativamente ou pelas especificações. Como esta característica não é levada em consideração no projeto de filtros digitais FIR, a abordagem MSE se prova ineficiente, pois busca os aspectos menos importantes na otimização dos filtros.

O segundo problema da abordagem MSE é a montagem da curva-alvo. A questão consiste em como sair das especificações para a resposta em frequência desejada. Um método válido é utilizar a curva ideal de um filtro. Porém, como a resposta ideal é garantidamente inalcançável, é provável que durante o procedimento de otimização vários filtros bons sejam descartados por estarem mais distantes da curva ideal do que um filtro pior, se forem analisados ponto-a-ponto.

Isto torna clara a necessidade de uma nova forma de medir a qualidade das soluções candidatas. Esta nova função de *fitness* deve avaliar a conformidade da solução às especificações e, se possível, levar o algoritmo a buscar soluções com características desejáveis, como por exemplo a máxima planicidade da banda de passagem (minimização do *ripple*). Porém, ela deve priorizar o atendimento das especificações sobre a otimização destas características, pois de nada adianta um filtro que tenha uma banda de passagem extremamente plana e baixa atenuação na banda de rejeição, por exemplo. Este balanço pode ser obtido com pesos para cada fator a ser considerado, mas é importante que tais pesos tenham valores que levem à obtenção de bons resultados com qualquer tipo de filtro. Não

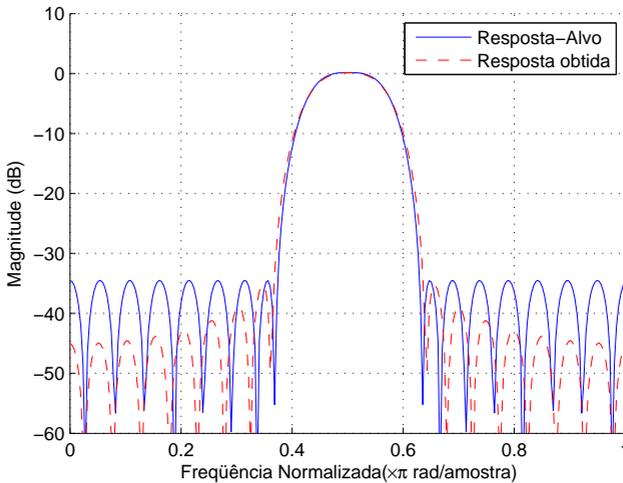


Figura 2. Exemplo 2 de curva-alvo e resposta obtida.

é desejável criar mais parâmetros para que o usuário tenha que ajustar a função de *fitness* para cada filtro que for projetar.

## 2.4 Computação evolucionária

Computação Evolucionária é uma área da Inteligência Computacional destinada à resolução de problemas de otimização através do uso de processos iterativos. A cada iteração novos indivíduos são gerados a partir de indivíduos das iterações anteriores segundo uma meta-heurística, encontrando soluções aproximadas potencialmente boas para um problema em especial. Estas meta-heurísticas são inspiradas em alguns fenômenos naturais, como por exemplo mecanismos biológicos.

Os principais algoritmos evolucionários (ou evolutivos) que se encaixam nestas meta-heurísticas são os Algoritmos Genéticos (AG), a Programação Genética (GP), a Otimização por Colônia de Formigas (ACO) e a Otimização por Enxame de Partículas (PSO), estas três últimas sub-agrupadas no que se conhece como Inteligência de Enxames (*Swarm Intelligence*). Também há outras meta-heurísticas inspiradas em outros fenômenos, como é o caso da Busca Harmônica (HS) e que são também incluídas na área de Inteligência Computacional. Particularmente, neste trabalho serão utilizadas as técnicas AG, PSO e HS para o projeto de filtros digitais FIR.

O Algoritmo Genético, proposto por [Holland \(1975\)](#) e popularizado por [Goldberg \(1989\)](#), é uma meta-heurística que combina conceitos de gené-

tica com as idéias evolutivas de Charles Darwin. Cada solução potencial representa um indivíduo de uma população, e estas soluções são capazes de evoluir através de processos de cruzamento e mutação. Estes indivíduos se recombinam gerando novos indivíduos em uma nova geração da população. Em alguns casos um ou mais dos melhores indivíduos são mantidos na geração seguinte, o que é conhecido como elitismo. Num AG canônico, a evolução é controlada pelos parâmetros *popsize* (tamanho da população), *pcross* (probabilidade de cruzamento), *pmutation* (probabilidade de mutação).

A Otimização por Enxames de Partículas (PSO) foi proposta por [Kennedy & Eberhart \(1995\)](#) e é inspirada no comportamento coletivo de bandos de pássaros e cardumes de peixes. No PSO, cada solução potencial é denominada partícula. Elas se movem no espaço de busca de acordo com equações simples, sendo que as partículas com melhores soluções guiam as demais no espaço de busca. O movimento de cada partícula é determinado pela sua posição atual, acrescido de um pequeno movimento aleatório e uma velocidade. Os parâmetros que influenciam o seu comportamento são:  $m$  (número de partículas do enxame),  $w$  (ponderação de inércia),  $c_1$  (parâmetro cognitivo) e  $c_2$  (parâmetro social).

A Busca Harmônica ([Geem et al., 2001](#)) é inspirada no processo de composição de músicos de jazz, na busca de uma harmonia esteticamente agradável. Diferentemente do AG e do PSO, a cada iteração apenas uma nova harmonia é gerada. Existe uma memória das melhores harmonias geradas, se a nova harmonia for melhor que a pior da memória, esta é substituída pela nova harmonia. A nova harmonia é composta por partes de diversas outras harmonias gravadas na memória, sendo que estas partes podem ser idênticas às das harmonias originais ou podem sofrer pequenas modificações. A forma como estas novas harmonias são compostas depende da configuração de alguns parâmetros: *HMS* (tamanho da memória harmônica), *HMCR* (taxa de consideração de um valor existente na memória harmônica), *PAR* (taxa de ajuste fino no valor recuperado da memória harmônica).

### 3. Metodologia

#### 3.1 Cálculo do *fitness*

Como explicado na Seção 2.3, o cálculo do *fitness* de um filtro pela abordagem do MSE em relação a uma curva-alvo não é satisfatório. Por isto, neste trabalho propõe-se uma nova abordagem, que avalia de forma quantitativa o atendimento ou não às especificações fornecidas pelo projetista.

Para calcular a qualidade do filtro, é necessário primeiramente calcular a resposta em frequência do mesmo. Esta resposta é obtida através da transformada discreta de Fourier (DFT) dos coeficientes do filtro. Porém,

como a transformada de um vetor de  $N$  posições é outro vetor de  $N$  posições, representando frequências igualmente distribuídas entre 0 e  $F_s$  (onde  $F_s$  é a frequência de amostragem), essa operação fornecerá uma resposta com resolução tão baixa quanto o número de coeficientes do filtro. Como explicado na Seção 2.1, os coeficientes de um filtro representam a sua resposta ao impulso. Como ela é finita, isso quer dizer que após  $N$  amostras, a saída será sempre nula. Ou seja, um vetor de  $N + Y$  posições, onde os  $N$  primeiros valores são os coeficientes do filtro, e as últimas  $Y$  posições são zeros, também representa a resposta ao impulso do filtro. Porém, a DFT deste vetor terá  $N + Y$  posições também igualmente distribuídas entre 0 e  $F_s$ . Com esta técnica, chamada de *zero padding*, é possível aumentar a resolução da resposta em frequência calculada, permitindo uma avaliação do filtro muito mais acurada.

Como apenas a magnitude da resposta em frequência é interessante para este trabalho, é calculado o módulo de cada elemento deste vetor. Além disto, metade dos elementos é descartada. Isso ocorre por que a resposta em frequência é simétrica em relação a  $\frac{F_s}{2}$ , ou seja, apenas a primeira metade da resposta é necessária para efetuar os cálculos. Não somente isto, mas o critério de Nyquist (Lathi, 1998) diz que não se pode utilizar sinais com componentes em frequência acima de  $\frac{F_s}{2}$ . Logo, apenas a metade inferior da resposta da DFT é relevante.

Portanto, o vetor de resposta utilizada para avaliar o filtro é obtido conforme a Equação 2, que é o módulo do vetor resultante da DFT dos coeficientes do filtro com  $k = N + Y$  elementos.

$$\vec{X} = |\mathcal{F}\{\vec{x} \ 0_1 \ 0_2 \ \dots \ 0_{k-N}\}| \quad (2)$$

onde  $\vec{X}$  é o vetor de resposta do filtro,  $\mathcal{F}$  é a transformada de Fourier,  $\vec{x}$  é o vetor de coeficientes do filtro,  $N$  é o número de coeficientes, e  $k$  é o número de elementos da transformada de Fourier (isto é,  $N$  adicionado de  $k - N$  zeros). Os primeiros  $\frac{k}{2}$  elementos de  $\vec{X}$  são então transformados no vetor  $\vec{r}$ , que será utilizado para os cálculos.

De posse do vetor de resposta  $\vec{r}$ , separa-se este vetor em  $r_{rej}$  e  $r_{pas}$ , respectivamente, a resposta na banda de rejeição e a resposta na banda de passagem. Desta forma é mais simples fazer a avaliação da qualidade do filtro, pois o requisito é o mesmo para a banda inteira.

Tendo em mãos as respostas nas bandas especificadas, pode-se calcular o *fitness* deste filtro, composto por quatro termos a serem otimizados, conforme a Equação 3.

$$fit(\vec{x}) = \frac{1}{1 + p1 \cdot specRej + p2 \cdot specPas + p3 \cdot idealRej + p4 \cdot idealPas} \quad (3)$$

onde  $fit(\vec{x})$  é o valor de *fitness* do vetor  $\vec{x}$  de coeficientes do filtro a ser avaliado,  $specRej$  é o termo que diz respeito à obediência da banda de rejeição às especificações,  $specPas$  se relaciona com o atendimento da banda de passagem,  $idealRej$  e  $idealPas$  dizem o quão próxima se está da resposta ideal cada banda, e  $p1$  a  $p4$  são os pesos de cada um dos termos na equação.

Para calcular o valor de  $specRej$ , é utilizada a Equação 4.

$$specRej = \frac{\sum_{i=1}^k [\max(gMaxRej, r_{rej}(i)) - gMaxRej]^2}{largRej} \quad (4)$$

onde  $k$  é o número de elementos do vetor  $r_{rej}$ ,  $gMaxRej$  é o máximo ganho permitido na banda de rejeição, e  $largRej$  é a largura em quantidade de amostras da banda de rejeição. Este modo de calcular o erro na banda de rejeição é similar ao MSE, mas apenas considera os elementos da resposta que tenham ganho acima do permitido. Ou seja, mesmo que esta resposta na banda de rejeição varie com um *ripple* muito elevado, ele terá erro zero se este *ripple* estiver sempre abaixo do máximo. Desta forma, este termo mede de forma eficaz a obediência do filtro à especificação da banda de rejeição. A divisão pela largura da banda visa uma certa normalização do erro por banda. Isso é necessário pois como as bandas de um filtro têm larguras diferentes, um erro grande na banda mais estreita geraria um erro numericamente inferior a um erro pequeno na banda mais larga.

De forma similar, o termo  $specPas$  é calculado segundo a Equação 5.

$$specPas = \frac{\sum_{i=1}^k [[\min(gMinPas, r_{pas}(i)) - gMinPas]^2 + [\max(1, r_{pas}(i)) - 1]^2]}{largPas} \quad (5)$$

onde  $k$  é o número de elementos do vetor  $r_{pas}$ ,  $largPas$  é a largura da banda de passagem em número de amostras, e  $gMinPas$  é o mínimo ganho permitido na banda de passagem, que também pode ser interpretado como o máximo *ripple* permitido, já que, por definição, o ganho máximo é 1. Este termo então calcula o erro quadrático em relação à faixa de ganho permitida na banda de passagem. Portanto, o ganho pode estar em qualquer ponto entre 1 e  $gMinPas$ , e terá erro zero. Desta forma, apenas ganhos que desobedecerem a especificação da banda de passagem gerarão erro.

Se os termos  $specPas$  e  $specRej$  avaliam a conformidade do filtro às especificações, os outros fornecem uma medida de qualidade do filtro no sentido de proximidade com a resposta ideal. Eles são desejáveis pois os termos *spec* não diferenciam entre filtros que obedeçam às especificações, sendo que uma boa função de *fitness* deve levar o algoritmo a buscar um filtro com a melhor resposta possível. As Equações 6 e 7 mostram como estes termos são calculados.

$$idealPas = \frac{\sum_{i=1}^k [1 - r_{pas}(i)]^2}{largPas} \tag{6}$$

$$idealRej = \frac{\sum_{i=1}^k [r_{rej}(i)]^2}{largRej} \tag{7}$$

Estas equações são mais simples pois um filtro ideal tem um valor fixo para o ganho em cada banda, em vez de uma faixa de valores que a especificação supõe. A consequência de ter estes dois termos a mais na equação do *fitness* é que jamais existirá um filtro com *fitness* igual a 1, pois para isso os 4 termos da equação precisariam ser iguais a zero. Enquanto os termos *spec\** podem e devem chegar a zero ao final da evolução, os termos *ideal\** jamais atingirão este valor, já que apenas um filtro ideal teria tal característica.

### 3.2 Experimentos

Para avaliar a metodologia proposta para cálculo do *fitness* e comparar os métodos de computação evolucionária, foram utilizados quatro filtros digitais diferentes: um passa-baixas (PB), um passa-altas (PA), um passa-faixa (PF) e um rejeita-faixa (RF). Os resultados foram comparados com o método tradicional de *Parks-McClellan*, que garante um *ripple* constante nas bandas (McClellan & Parks, 1973). Este método foi escolhido em vez de um que utilize janelamento, porque permite a especificação de cada banda, enquanto que por função-janela só é possível escolher a frequência de corte.

Cada filtro a ser projetado deverá ter pelo menos 40dB de atenuação na banda de rejeição, e no máximo 1dB de *ripple* na banda de passagem. As frequências que delimitam cada banda em cada filtro, assim como a ordem de cada um, estão na Tabela 1. As frequências estão normalizadas em relação a  $\frac{F_s}{2}$ , como é comum no projeto de filtros. A coluna Ordem mostra a ordem necessária para atingir as especificações com o método *Parks-McClellan*, seguida da ordem de fato utilizada nos experimentos. Ou seja, alguns destes filtros têm ordem menor do que a necessária, tornando o problema muito mais difícil, pois é matematicamente impossível atingir a especificação.

Tabela 1. Frequências características dos filtros de teste.

Filtro	Ordem	B. Passagem 1		B. Passagem 2		B. Rejeição 1		B. Rejeição 2	
		$\omega_1$	$\omega_2$	$\omega_1$	$\omega_2$	$\omega_1$	$\omega_2$	$\omega_1$	$\omega_2$
PB	71 / 53	0.00	0.80	-	-	0.84	1.00	-	-
PA	76 / 54	0.74	1.00	-	-	0.00	0.70	-	-
PF	28 / 40	0.47	0.53	-	-	0.00	0.37	0.63	1.00
RF	60 / 104	0.00	0.27	0.40	1.00	0.32	0.35	-	-

Para uma DFT de 512 amostras, como utilizado neste trabalho, um  $\omega = 1,00$  equivale à amostra de índice 256. Para acelerar os cálculos, as bandas foram delimitadas na implementação através de índices, evitando a repetição do cálculo de arredondamento. Os índices utilizados estão na Tabela 2. Eles foram calculados multiplicando-se a frequência normalizada  $w$  pelo número de amostras consideradas no cálculo (256). O arredondamento foi feito de forma que a banda especificada estivesse sempre contida na banda efetivamente considerada pelos índices. Isto foi realizado arredondando-se para baixo os índices correspondentes às frequências iniciais de cada banda, e arredondando-se para cima os índices equivalentes às frequências finais.

Tabela 2. Índices característicos dos filtros de teste.

Filtro	B. Passagem 1		B. Passagem 2		B. Rejeição 1		B. Rejeição 2	
	$i1$	$i2$	$i1$	$i2$	$i1$	$i2$	$i1$	$i2$
PB	1	205	-	-	209	256	-	-
PA	1	179	-	-	190	256	-	-
PF	119	136	-	-	1	96	160	256
RF	1	69	102	256	81	90	-	-

Cada experimento foi executado 20 vezes com cada algoritmo evolucionário objetivando avaliar o desempenho médio. Em cada algoritmo, o tamanho da população foi 200, com 1000 iterações, limitando, assim, o número de execuções da função de *fitness* em 200.000. Foram utilizados os valores padrão de parâmetros para cada um, conforme a literatura. Para o PSO,  $c_g = 1.8$ ,  $c_l = 1.8$ ,  $w = 0.6$  e  $v_{max} = 0.005$ . Para o AG, a probabilidade de mutação foi 1%, e a taxa de *crossover* 70%. E no HS, os valores foram  $HMCR = 90\%$ ,  $PAR = 5\%$  e  $HMS = 20$ .

Após a inicialização dos indivíduos em cada método, foi realizada uma normalização em função do maior ganho de cada filtro, garantindo, assim, que todos os indivíduos representassem filtros cuja resposta se situasse entre 0 e 1, ou seja, entre  $-\infty$  dB e 0 dB. Esta operação foi realizada calculando-se a resposta do filtro com a DFT, procurando o maior ganho no vetor resposta, e dividindo todos os coeficientes por esse valor. Isso é possível pois uma das propriedades da transformada de Fourier é a linearidade, ou seja, é possível multiplicar um sinal por uma constante tanto no domínio do tempo quanto no domínio da frequência, obtendo o mesmo resultado.

## 4. Resultados e Discussão

Nesta seção, os resultados obtidos pelos três algoritmos são apresentados e comparados com o filtro calculado por *Parks-McClellan*. Em cada figura a seguir, os pontos marcados são as frequências que limitam cada banda.

### 4.1 Comparação entre Parks-McClellan e PSO

Por razões que serão apresentadas na Seção 5, o algoritmo PSO alcançou os melhores resultados entre os métodos evolucionários testados. Por isso, a eficiência da função de *fitness* foi comparada ao método Parks-McClellan utilizando os resultados do PSO. Cada filtro foi calculado 5 vezes utilizando o PSO, sendo escolhido o melhor resultado para a comparação. No filtro PB, mostrado nas Figuras 3 e 4, a ondulação na banda de passagem foi similar, mas o desempenho próximo das frequências delimitadoras de banda não. O Parks-McClellan privilegiou a banda de passagem, fazendo com que a banda de rejeição começasse ainda com um ganho próximo de -3dB. O filtro obtido pelo PSO iniciou a queda um pouco antes, ficando um um ganho próximo de -7dB no final da banda de passagem, mantendo a banda de rejeição inteira abaixo de -20dB.

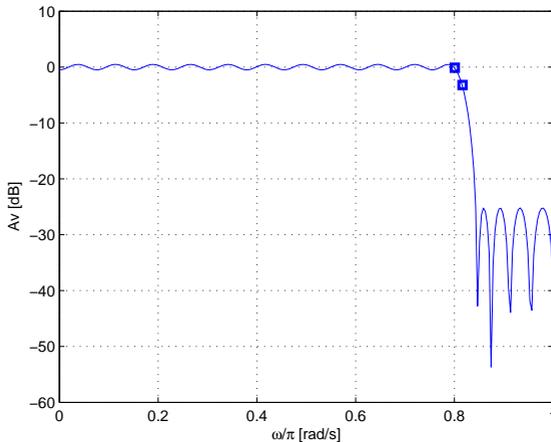


Figura 3. Resposta do filtro PB obtido pelo método Parks-McClellan.

No filtro PA, mostrado nas Figura 5 e 6, a ondulação na banda de passagem foi similar, mas o desempenho próximo das frequências delimitadoras de banda não. O Parks-McClellan privilegiou a banda de passagem, fazendo com que a banda de rejeição começasse ainda com um ganho próximo de -3dB. O filtro obtido pelo PSO iniciou a queda um pouco antes, ficando um um ganho próximo de -7dB no final da banda de passagem, mantendo a banda de rejeição inteira abaixo de -20dB.

Para o filtro PF, o resultado obtido por ambas as abordagens, PSO e Parks-McClellan, foi satisfatório, com ligeira vantagem para o método evolucionário. A Figura 7 mostra que no início da segunda banda de rejei-

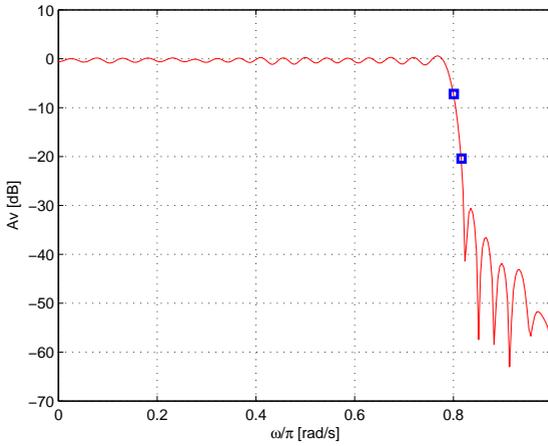


Figura 4. Resposta do filtro PB obtido pelo algoritmo PSO.

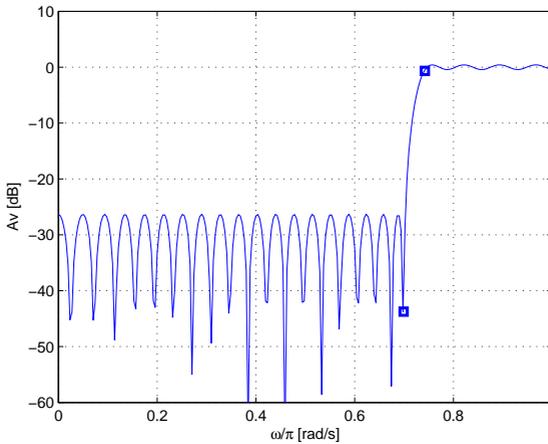


Figura 5. Resposta do filtro PA obtido pelo método Parks-McClellan.

ção, o ganho ainda é alto, enquanto o filtro obtido pelo PSO tem as duas bandas de rejeição abaixo de -40dB, como exibido na Figura 8 .

No filtro RF, a ondulação na banda de passagem foi praticamente eliminada pelo Parks-McClellan (Figura 9), enquanto o filtro obtido pelo PSO

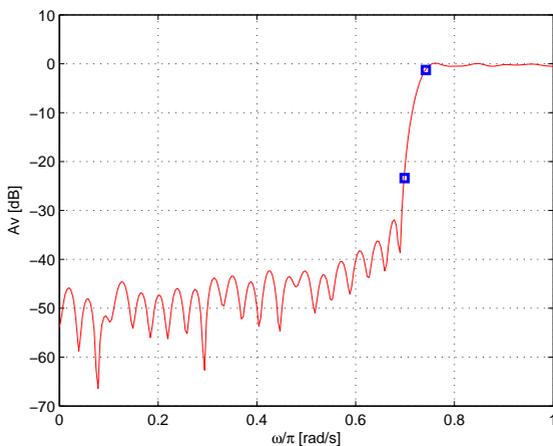


Figura 6. Resposta do filtro PA obtido pelo algoritmo PSO.

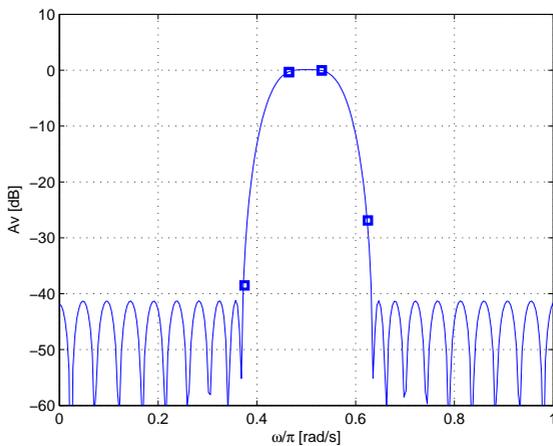


Figura 7. Resposta do filtro PF obtido pelo método Parks-McClellan.

(Figura 10) não conseguiu um bom resultado neste quesito. Em compensação, sua banda de rejeição está inteira abaixo de -40dB, enquanto que no Parks-McClellan ela inicia próxima de -23dB.

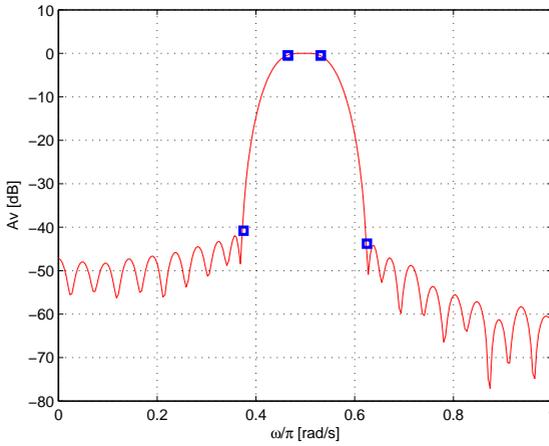


Figura 8. Resposta do filtro PF obtido pelo algoritmo PSO.

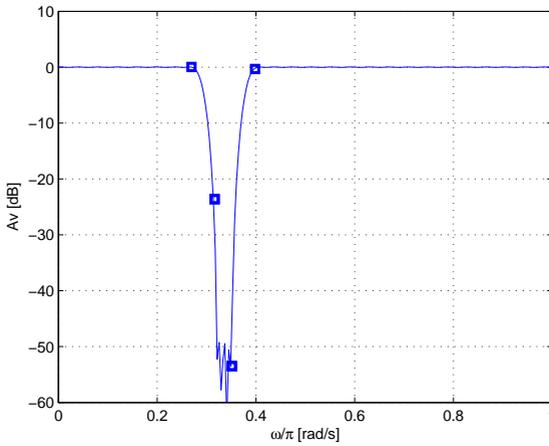


Figura 9. Resposta do filtro RF obtido pelo método Parks-McClellan.

### 4.2 Comparação entre métodos evolucionários

O objetivo desta comparação é avaliar qual dos algoritmos evolucionários obtém melhores resultados para o projeto de filtros FIR como função de *fitness* proposta.

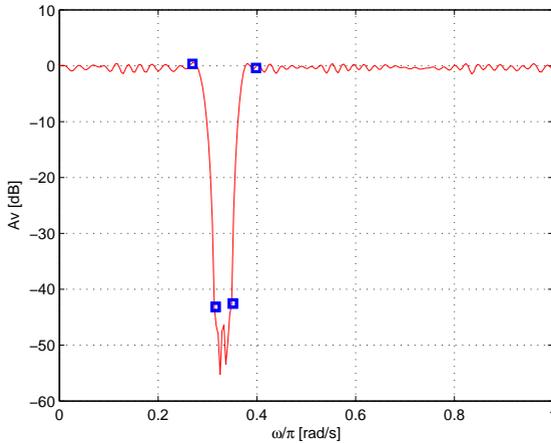


Figura 10. Resposta do filtro RF obtido pelo algoritmo PSO.

Os resultados numéricos para os três algoritmos nos quatro filtros são mostrados na Tabela 3. São mostrados os resultados médios e o melhor alcançado por cada um. A tabela mostra que o PSO foi o algoritmo que obteve os melhores resultados para todos os filtros. Assim, em complemento à tabela, as Figuras 11, 12, 13 e 14 mostram as curvas de resposta em frequência do melhor filtro obtido pelo PSO e dos correspondentes com o método Parks-McClellan.

Tabela 3. *Fitness* obtidos em 20 rodadas.

Filtro	AG		HS		PSO	
	Med	Max	Med	Max	Med	Max
PB	0.8541 ± 0.0309	0.8952	0.8705 ± 0.0504	0.9363	0.9526 ± 0.0044	0.9569
PA	0.8640 ± 0.0096	0.8324	0.8736 ± 0.0081	0.8792	0.9954 ± 0.0020	0.9981
PF	0.9537 ± 0.0270	0.9873	0.9982 ± 0.0015	0.9996	0.9990 ± 0.0005	0.9996
RF	0.8179 ± 0.0371	0.8857	0.9327 ± 0.0134	0.9546	0.9896 ± 0.0035	0.9950

## 5. Conclusões

Neste trabalho foi proposta uma nova abordagem para o cálculo de *fitness* de filtros digitais com métodos de computação evolucionária. Esta abordagem avalia a conformidade do filtro às especificações em vez de buscar igualar uma curva-alvo. Os resultados obtidos se mostraram melhores do

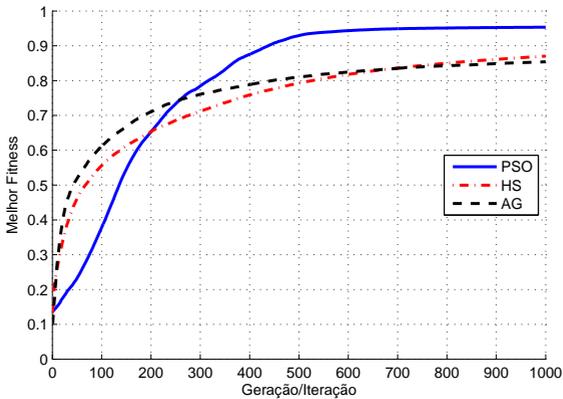


Figura 11. Evolução média do filtro PB.

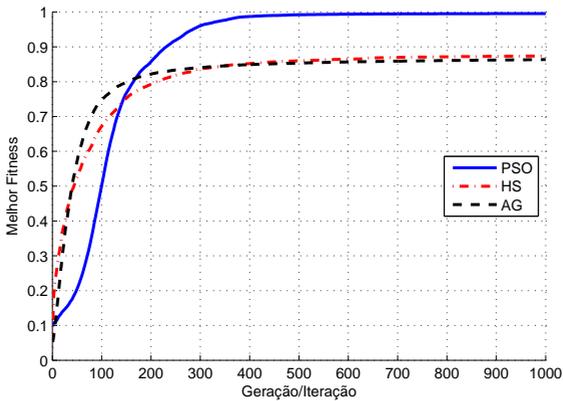


Figura 12. Evolução média do filtro PA.

que os fornecidos pelo método Parks-McClellan, embora a ondulação na banda de passagem tenha sido pior em alguns casos.

Esta nova função de *fitness* é composta por 4 termos, que neste trabalho foram combinadas através de uma soma ponderada. Futuros trabalhos tratarão do uso de técnicas multiobjetivo em vez desta soma, que exige a atribuição de pesos para cada termo.

Dentre os métodos evolucionários testados, o PSO foi o que obteve consistentemente os melhores resultados. Isto foi verificado na média, no má-

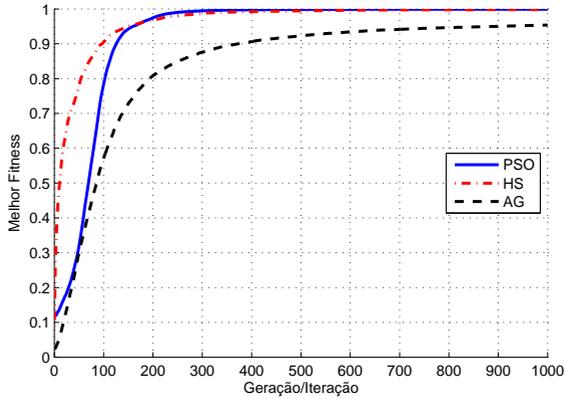


Figura 13. Evolução média do filtro PF.

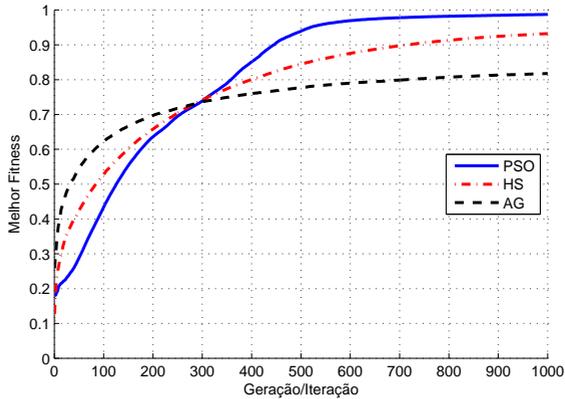


Figura 14. Evolução média do filtro RF.

ximo encontrado, e na repetibilidade do desempenho, avaliado pelo desvio-padrão. É muito importante observar que nenhum esforço foi feito para ajustar os parâmetros dos algoritmos testados e sabe-se que parâmetros diferentes podem levar a desempenhos diferentes para o mesmo problema. É mostrado na Seção 4.2 que um dos diferenciais do PSO foi a convergência mais lenta, o que costuma levar a resultados melhores em algoritmos de computação evolucionária. Como a velocidade de convergência é facilmente afetada pelos parâmetros dos algoritmos, é possível que com outras con-

figurações estes possam obter resultados mais similares entre si ou ainda melhores dos que são aqui reportados. Esta questão será estudada em trabalhos futuros.

É necessário ressaltar que a abordagem proposta não leva em conta a resposta em fase do filtro. Muitas aplicações demandam atraso de grupo constante, ou seja, fase linear na banda de passagem. É possível provar que qualquer filtro com essa característica segue um de 4 possíveis conjuntos de características, cada grupo chamado na literatura por I, II, III ou IV (Oppenheim & Schafer, 1975). Dentre estes conjuntos de características, todos eles compartilham a simetria ou anti-simetria dos coeficientes, ou seja, a segunda é metade é igual à primeira, com o mesmo sinal ou com sinal trocado. Por isto, um filtro com a restrição da fase linear a ser otimizado com  $N$  coeficientes terá apenas  $\frac{N}{2}$  variáveis passíveis de controle, pois a outra metade é diretamente relacionada à primeira. Desta forma ocorre uma redução do espaço de busca, uma vez que o número de variáveis cai à metade. Esta redução em geral deixa o problema mais fácil de ser resolvido por métodos de computação evolucionária. Isto sugere fortemente que a função de *fitness* proposta também seja capaz de obter bons resultados nesta sub-classe de filtros. Este também será assunto de futuros trabalhos.

## Agradecimentos

Os autores agradecem ao CNPq pela bolsa de produtividade em pesquisa a H.S. Lopes, à UDESC e ao programa FUMDES pela bolsa de doutorado a R.S. Parpinelli, bem como à CAPES pelas bolsas de mestrado (demanda social) a D.R. Oliveira, M.H. Scalabrin e F.R. Teodoro.

## Referências

- Ahmed, S., Design of FIR filters with arbitrary amplitude and phase specifications using genetic algorithm. In: *Proceedings of IEEE International Symposium on Micro-NanoMechatronics and Human Science*. Piscataway, USA: IEEE, v. 2, p. 648–651, 2003.
- Barros, A., *Projeto de Filtros FIR Através de Algoritmos Genéticos*. Dissertação de mestrado, Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná, 2006. 127 p.
- Barros, A.; Lopes, H. & Stelle, A., FIR filter design method and tool based on genetic algorithms. In: *Proceedings of the 1st IEEE Symposium on Computational Intelligence in Image and Signal Processing*. Piscataway, USA: IEEE Computer Society, v. 1, p. 151–156, 2007.
- Diniz, P.; da Silva, E. & Netto, S., *Processamento Digital de Sinais – Projeto e Análise de Sistemas*. Porto Alegre, RS: Artmed, 2004. 590 p.

- Geem, Z.; Kim, J. & Loganathan, G., A new heuristic optimization algorithm: Harmony search. *Simulation*, 76(2):60–68, 2001.
- Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, USA: Addison-Wesley, 1989.
- Hamming, R., *Digital Filters*. 3a edição. Mineola, NY: Dover Publications, 1997.
- Holland, J., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- Huang, W.P.; Zhou, L.F. & Qian, J.X., FIR filter design: Frequency sampling filters by particle swarm optimization algorithm. In: *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*. v. 4, p. 2322–2327, 2004.
- Kennedy, J. & Eberhart, R., Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. IEEE, v. 4, p. 1942–1948, 1995.
- Kennedy, J. & Eberhart, R., *Swarm Intelligence*. San Francisco, USA: Morgan Kaufmann, 2001.
- Lathi, B., *Modern Digital and Analog Communication*. 3a edição. New York, USA: Oxford University Press, 1998.
- McClellan, J. & Parks, T., A unified approach to the design of optimum FIR linear phase filters. *IEEE Transactions on Circuit Theory*, 1:190–196, 1973.
- Najjarzadeh, M. & Ayatollahi, A., A comparison between genetic algorithm and PSO for linear phase FIR digital filter design. In: *Proceedings of 9th International Conference on Signal Processing*. IEEE, v. 4, p. 2134–2137, 2008.
- Oppenheim, A. & Schaffer, R., *Digital Signal Processing*. Prentice-Hall, 1975. 585 p.
- Oppenheim, A. & Schaffer, R., *Discrete-Time Signal Processing*. 2a edição. Prentice-Hall, 1999. 870 p.
- Williams, A. & Taylors, F., *Electronic Filter Design Handbook*. 3a edição. New York, USA: McGraw-Hill, 1995. 800 p.
- Zhang, J.; Jia, D. & Li, K., FIR digital filters design based on chaotic mutation particle swarm optimization. In: *Proceedings of the International Conference on Audio, Language and Image Processing*. v. 1, p. 418–422, 2008.

## Notas Biográficas

**Daniel Rossato de Oliveira** é graduado em Engenharia Elétrica pela Universidade Tecnológica Federal do Paraná/Curitiba (UTFPR, 2009) e atualmente é mestrando na área de Engenharia de Computação na mesma universidade, além de ser professor substituto do Departamento de Eletrônica da UTFPR/Curitiba.

**Marlon Henrique Scalabrin** é formado em Engenharia da Computação pela Universidade Estadual de Ponta Grossa – UEPG (2008); Especialista em Tecnologia Java pela UTFPR/Pato Branco (2009) e atualmente é mestrando na área Engenharia Elétrica e Informática na UTFPR/Curitiba.

**Fábio Roberto Teodoro** é graduado em Ciência da Computação (UDESC/Joinville, 2008) e atualmente é mestrando em Informática na UTFPR/Curitiba.

**Rafael Stubs Parpinelli** é graduado em Ciência da Computação pela Universidade Estadual de Maringá (2000) e tem mestrado em Informática Industrial (UTFPR, 2001). Atualmente faz parte do Laboratório de Bioinformática da UTFPR como aluno de doutorado. Ele também é professor efetivo da UDESC/Joinville no Departamento de Ciência da Computação. Suas principais áreas de interesse são bioinformática e todas as formas de algoritmos bioinspirados para otimização.

**Heitor Silvério Lopes** é graduado em Engenharia Elétrica (UTFPR/Curitiba, 1984) e tem mestrado em Engenharia Biomédica (UTFPR/Curitiba, 1990) e doutorado em Engenharia Elétrica (UFSC/Florianópolis, 1996). Atualmente é professor associado do Departamento de Eletrônica da UTFPR e orientou cerca de 35 dissertações e teses. Suas áreas de interesse são: computação evolucionária, bioinformática, computação de alto desempenho e lógica reconfigurável.

### Caracterização e Otimização de Câmaras Reverberantes para Testes de Compatibilidade Eletromagnética

Sérgio L. Avila\*, Djonny Weinzierl,  
Mário A. Santos Jr., Luiz Lebensztajn,  
Carlos A.F. Sartori, José R. Cardoso, Laurent Krähenbühl

---

**Resumo:** A crescente preocupação com a qualidade dos equipamentos eletrônicos fomenta esforços para o desenvolvimento de metodologias e equipamentos de testes para análise de compatibilidade eletromagnética (CEM). Atualmente, câmaras reverberantes são foco de estudos como ambiente propício para estes ensaios. Este artigo propõe um processo de caracterização e otimização de uma câmara reverberante excitada a partir de um arranjo de condutores. Foram utilizados os Algoritmos Genéticos para, primeiramente, com um procedimento multiobjetivo, entender o comportamento da câmara e, em segundo lugar, com um procedimento mono-objetivo, otimizá-la. Além de resultados expressivos, é mostrada a contribuição deste trabalho na busca pelo desenvolvimento científico de câmaras reverberantes.

**Palavras-chave:** Caracterização, Otimização, Algoritmos genéticos, Compatibilidade eletromagnética.

**Abstract:** *The growing concern about the electronic equipment certification has fueled efforts to develop methodologies and test equipments for electromagnetic compatibility (EMC) analysis. Nowadays, reverberation chambers are the focus of research on the development of EMC test equipments. This paper proposes a characterization and optimization procedure to analyze a reverberation chamber excited by transmission lines. We have used multi and mono-objective evolutionary computation (namely, Genetic Algorithms) procedures in order to understand the chamber behavior and optimize it, respectively. Expressive results were obtained, showing a contribution to the analysis of reverberation chambers.*

**Keywords:** *Characterization, Optimization, Genetic algorithms, Electromagnetic compatibility.*

---

\* Autor para contato: Sergio.Avila@ifsc.edu.br

## 1. Introdução

Devido à constante preocupação em atender as exigências normativas quanto aos produtos comercializados no Brasil e no exterior (ANATEL, 2000; IEC, 2003), novas técnicas de medição relacionadas aos testes de Compatibilidade Eletromagnética (CEM) requerem a atenção de Universidades e Institutos de Pesquisa em todo o mundo. Contudo, o alto custo envolvido no desenvolvimento de tais técnicas pode se apresentar como uma barreira econômica. No intuito de se buscar alternativas menos onerosas, inúmeros estudos vêm sendo empreendidos nos meios acadêmico e industrial, dentre os quais se destacam o desenvolvimento de câmaras reverberantes de modos misturados (IEC, 2003).

O modelo canônico da câmara reverberante constitui um ambiente blindado, cujos modos de propagação transversal-elétrico (TE) e transversal-magnético (TM) são misturados em seu interior por um ou mais arranjos de pás metálicas girantes. Este processo visa obter um campo eletromagnético estatisticamente uniforme, cujo campo elétrico é avaliado com base no desvio padrão das componentes deste campo. Várias propostas são analisadas pela comunidade científica com o intuito de se resolver o problema apresentado. Dentro deste contexto, estudos vêm sendo desenvolvidos pelo grupo de pesquisadores autores do presente artigo, que busca analisar a excitação de câmaras reverberantes de modos misturados realizada por arranjos de condutores, em substituição às pás anteriormente referidas. Os condutores representam uma grande vantagem como método de excitação, pois o espaço por eles requerido é menor em relação ao modelo canônico, onde pás suspensas localizadas na parede superior da câmara ocupam um volume considerável. Assim, se comparadas às convencionais, as dimensões da câmara podem ser reduzidas, o que torna sua construção economicamente viável e melhor aplicável a equipamentos de pequeno porte.

A proposta do presente artigo diz respeito à caracterização e à otimização de uma câmara reverberante excitada por condutores, conforme ilustrado na Figura 1. O objetivo principal é conhecer o comportamento dos campos dentro da câmara, o que permite a determinação e maximização do volume de trabalho útil. Para clareza de entendimento, seguem duas definições, de acordo com as acepções utilizadas neste trabalho:

- **Caracterização:** é compreender o compromisso entre o tamanho do volume de trabalho disponível dentro da câmara e a homogeneidade do campo dentro deste volume – isto é feito com a busca pela fronteira Pareto;
- **Otimização:** é, por exemplo, encontrar as posições geométricas dos condutores, bem como a forma mais adequada de excitá-los (frequências, amplitudes, defasagens e formas de onda), com o objetivo de obter a melhor câmara reverberante possível (segundo requisitos determinados por normas).

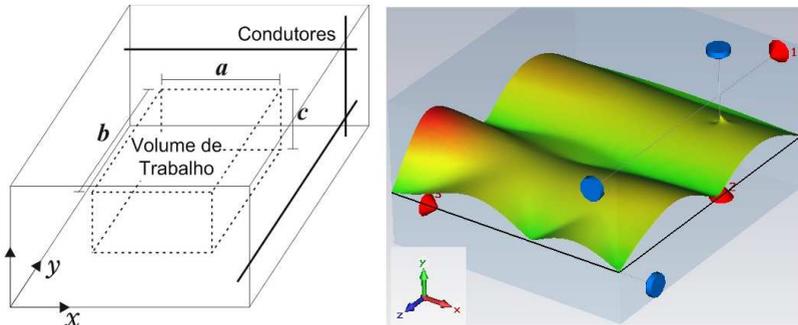


Figura 1. Câmara reverberante excitada com condutores. (esquerda) Geometria da câmara reverberante. (direita) Distribuição típica do campo elétrico.

Tanto para a caracterização quanto para a otimização, utilizaram-se ferramentas de computação evolucionária, os Algoritmos Genéticos mono-objetivo (AG) e multi-objetivo (AGMO) (Avila, 2006). A escolha destas ferramentas foi em função das características do problema a ser resolvido. Isto está descrito na Seção 2, após a apresentação do problema, sua formulação matemática e todos os seus detalhes.

Apresenta-se aqui uma compilação dos trabalhos já publicados pelo grupo de autores (Weinzierl et al., 2008; Avila et al., 2009; Santos Jr. et al., 2010) e os últimos avanços atingidos. Os resultados obtidos pelo esforço da equipe resultaram na melhoria da compreensão deste tipo de câmara para testes de compatibilidade.

## 2. Fundamentação Teórica

Esta seção apresenta a fundamentação teórica deste trabalho. São reportados o entendimento físico e a formulação matemática de câmaras reverberantes de modos misturados canônicas e do modelo não convencional. Em seguida, são apresentados os detalhes sobre a caracterização do modelo não convencional e sua necessidade de otimização. Por fim, conhecendo-se o problema, parte-se para a justificativa de utilização da computação evolucionária, em específico os algoritmos genéticos, para a sua resolução.

### 2.1 Entendimento físico e formulação matemática do problema

A câmara reverberante de modos misturados canônica se destaca por oferecer uma alternativa economicamente viável aos testes de CEM de equipamentos de médio e grande portes (Weinzierl, 2004; Santos Jr., 2009). Tais dispositivos consistem em ambientes isolados do meio exterior, destacando-se por utilizarem o princípio físico das cavidades ressonantes. Tal caracte-

terística dispensa a utilização de absorvedores de Rádio-Frequência (RF), e.g. em câmaras anecóicas, cuja técnica permite que o campo elétrico no interior da câmara se mantenha uniforme e cujos investimentos tornam o projeto menos atraente do ponto de vista econômico. A caracterização do fenômeno ondulatório no interior de câmaras reverberantes de modos misturados se dá através de uma combinação adequada das equações de Maxwell e suas condições de contorno. Para uma câmara retangular de dimensões definidas, os únicos campos que podem ser encontrados em seu interior são os modos  $TE_{mnp}$  e  $TM_{mnp}$ . Estes estabelecem a única solução às condições de contorno do problema, cujos valores do campo elétrico tangencial nas paredes da câmara são nulos. Diversos modos de propagação podem ser obtidos, compreendendo a cada modo uma frequência de ressonância determinada pelas dimensões físicas da câmara. A menor frequência de ressonância é denominada de fundamental, de maneira que se a frequência de excitação for maior que a frequência fundamental, o modo correspondente será propagado sem atenuação. Caso contrário, a energia se atenuará exponencialmente com a distância, tendo um valor extremamente baixo a uma distância muito curta, isto é, há na câmara um modo evanescente.

A excitação de uma câmara de modos misturados canônica se dá através de uma antena disposta em seu interior, resultando em um campo variável definido pelas reflexões das paredes da câmara. Desta forma, nos locais onde as reflexões se somam em fase, o campo é máximo. Em outros pontos da câmara, onde as reflexões encontram-se defasadas, valores mínimos ocorrem. De forma a uniformizar os campos, encontram-se dispostas no teto da câmara estruturas de pás metálicas girantes, que, ao mudarem de posição em relação ao seu eixo de rotação, modificam as dimensões da câmara, experimentando campos que ressoam em várias frequências. Esta técnica permite que elevadas intensidades de campo sejam geradas no interior da câmara devido ao aumento da amplitude dos campos quando propagados nas frequências de ressonância. Do ponto de vista teórico, não há um limite máximo para sua frequência de operação. Porém, a frequência mínima deve ser estabelecida levando-se em conta algumas considerações de ordem prática, como as dimensões da câmara e um número mínimo de modos ressonantes, por exemplo. A frequência mínima de operação é estabelecida a partir de um valor cujos modos possam se propagar no interior da câmara, sendo estes obtidos através da Equação 1:

$$f_{mnp} = \sqrt{\frac{1}{\mu_0 \epsilon_0}} \cdot \left[ \sqrt{\left(\frac{m}{a}\right)^2 + \left(\frac{n}{b}\right)^2 + \left(\frac{p}{c}\right)^2} \right] \quad (1)$$

onde  $f_{mnp}$  é a frequência para o modo ressonante, em Hz;  $a$ ,  $b$  e  $c$  são as dimensões da câmara, em metros, e  $m$ ,  $n$  e  $p$  são números inteiros que

definem o modo da frequência de ressonância. A menor frequência de ressonância é denominada de fundamental. Para  $a < b < c$ ,  $m = 0$ ,  $n = 1$  e  $p = 1$ , por exemplo, o modo  $TE_{011}$  corresponde à frequência fundamental. Abaixo desta os modos se tornam evanescentes, limitando a operação da câmara às frequências encontradas acima da frequência fundamental. Pesquisas vêm sendo empreendidas no intuito de se estender a operação das câmaras reverberantes de modos misturados às baixas frequências (Weinzierl, 2004; Santos Jr., 2009). Obtém-se um maior número de modos para uma determinada frequência de excitação quando as dimensões das três paredes da câmara são distintas, evitando-se, assim, que modos degenerativos possam aparecer. Tanto a densidade dos modos propagados quanto a eficiência das pás girantes determinam a menor frequência de operação da câmara, que é definida para as condições em que a câmara esteja em conformidade com as exigências estabelecidas por normas específicas. Outros fatores que influenciam na adoção da menor frequência de operação da câmara são suas dimensões, bem como seu fator de qualidade. A Equação 2 define o fator de qualidade de uma câmara:

$$Q = \frac{16\pi^2 V}{\eta_{tx}\eta_{rx}\lambda^3} \times FC \quad (2)$$

onde  $Q$  é o fator de qualidade da câmara;  $\eta_{tx}$  e  $\eta_{rx}$  são os fatores de efetividade das antenas transmissora e receptora, respectivamente;  $V$  é o volume da câmara, em  $m^3$ ;  $\lambda$  é o comprimento da onda para a frequência de excitação, em metros, e  $FC$  é o fator de calibração da câmara. De acordo com a norma IEC 61000-4-21 (IEC, 2003), o fator de qualidade descreve a capacidade de uma câmara em armazenar energia – fator que se relaciona às perdas presentes no ambiente. A condutividade das paredes da câmara torna-se, normalmente, o principal fator atribuído às perdas de energia. Materiais de boa condutividade, isto é, cobre e alumínio, oferecem uma condição economicamente viável quanto ao aspecto custo versus benefício. Perdas adicionais em suportes de antenas e estruturas metálicas também podem influenciar no fator de qualidade de uma câmara. A blindagem da câmara torna-se, ao longo de um projeto, outro fator importante a ser avaliado, devido às várias aberturas necessárias, como portas e passagens para cabos e condutores, por exemplo. A efetividade de blindagem de uma câmara reverberante de modos misturados é definida pela norma IEC 61000-4-21, com base na Equação 3:

$$EB = -10 \log \left( \frac{P_{EET}}{P_R} \right) \quad (3)$$

onde  $EB$  é a efetividade de blindagem da câmara, em dB;  $P_{EET}$  e  $P_R$  são as potências medidas sobre a face  $EET$  e pela antena de referência, respectivamente.

No que diz respeito às dificuldades tecnológicas relacionadas ao projeto de câmaras reverberantes de modos misturados canônicas destaca-se a obtenção de uma distribuição de campos uniformes para um amplo espectro de frequências, cuja limitação principal ocorre principalmente para baixas frequências. Os esforços empreendidos pelos autores do presente artigo vêm ao encontro de uma solução para esta restrição. De acordo com (?), a restrição quanto ao limite de operação de câmaras reverberantes pode ser contornada por meio da substituição das pás mecânicas por arranjos de condutores, os quais são capazes de excitar o modo transversal-eletromagnético (TEM) para todo o espectro de frequências, além dos modos TE e TM. Esta técnica permite que os campos eletromagnéticos sejam propagados para todas as frequências no interior da câmara, cujo limite de operação se restringe apenas ao espectro do gerador de campos utilizado.

Os condutores representam uma grande vantagem como método de excitação, pois o espaço por eles requerido é menor em relação ao modelo convencional, onde pás suspensas localizadas na parede superior da câmara ocupam um volume considerável no interior da câmara. Assim, se comparadas às convencionais, as dimensões da câmara podem ser reduzidas, o que torna sua construção economicamente viável e melhor aplicável a equipamentos de teste de pequeno porte.

Sobre a metodologia originalmente proposta, um ponto a ser destacado diz respeito à otimização do posicionamento do arranjo de condutores como forma de se obter uma maior uniformidade do campo no interior da câmara analisada. Ressalta-se esta condição como sendo predeterminada pela norma que define as características de operação de câmaras reverberantes de modos misturados, tratada a seguir.

## 2.2 Volume de trabalho

A análise da homogeneidade do campo elétrico em uma câmara reverberante de modos misturados pode ser realizada com base na norma IEC 61.000-4-21 (IEC, 2003), que considera como índice de mérito os valores de desvio padrão das componentes individuais de campo elétrico e de seus valores combinados. Segundo a norma, os valores individuais e combinados para o cálculo do desvio padrão devem ser calculados a partir das Equações 4 e 5, respectivamente:

$$\bar{E}_k = \frac{1}{N} \sum_{i=1}^N E_{k,i} \quad (4)$$

$$\bar{E}_{x,y,z} = \frac{1}{3N} \sum_{k=x,y,z} \sum_{i=1}^N E_{k,i} \quad (5)$$

em que  $E_k$  representa a média aritmética da componente k do valor máximo

do campo elétrico  $\overline{E}_k$ ;  $\overline{E}_{x,y,z}$  a média aritmética do valor máximo do campo elétrico considerando-se as componentes  $x$ ,  $y$  e  $z$ , e  $N$  o número de amostras obtidas para cada componente do campo. Baseando-se nas Equações 4 e 5, obtêm-se os valores do desvio padrão individual  $\sigma_k$  e combinado  $\sigma_{x,y,z}$ , conforme seguem:

$$\sigma_k = \sqrt{\frac{\sum_{i=1}^N (E_{k,i} - \overline{E}_k)^2}{N - 1}} \quad (6)$$

$$\sigma_{k,y,z} = \sqrt{\frac{\sum_{k=x,y,z} \sum_{i=1}^N (E_{k,i} - \overline{E}_{x,y,z})^2}{3N - 1}} \quad (7)$$

ou, em uma notação em decibéis:

$$\hat{\sigma}_k = 20 \log_{10} \frac{\sigma_k + \overline{E}_k}{\overline{E}_k} \quad (8)$$

$$\hat{\sigma}_{x,y,z} = 20 \log_{10} \frac{\sigma_{x,y,z} + \overline{E}_{x,y,z}}{\overline{E}_{x,y,z}} \quad (9)$$

Conforme determinado pela norma, os valores individuais e combinado do desvio padrão entre um espectro de frequências de 80 MHz a 1 GHz, considerando-se uma função Gaussiana para a distribuição do campo elétrico, devem satisfazer as seguintes condições:

- $\sigma_k < 4$  dB e  $\sigma_{x,y,z} < 4$  dB entre 80 e 100 MHz;
- Entre 100 e 400 MHz, os limites de  $\sigma_k$  e  $\sigma_{x,y,z}$  decaem linearmente de 4 para 3 dB;
- $\sigma_k < 3$  dB e  $\sigma_{x,y,z} < 3$  dB entre 400 e 1 GHz.

A uniformidade do campo produzido dentro da câmara está diretamente relacionada ao número de posições assumidas pela estrutura de pás girantes. Durante a calibração, a estrutura de pás girantes deve operar de modo que o sensor de campo, posicionado em cada um dos pontos de medição, possa adquirir um número específico de amostras, compreendendo o valor máximo das componentes do campo elétrico dentro da área de testes especificada.

O volume de testes, ou volume de trabalho, é determinado a partir das dimensões da câmara, devendo-se descartar as proximidades das paredes devido à presença de muitos modos próximos um do outro. A Figura 1 apresenta uma ilustração deste volume.

O presente trabalho, portanto, direciona especial atenção à operação da câmara em frequências abaixo da frequência fundamental, considerando a maximização do volume de trabalho e a homogeneidade do campo elétrico em seu interior. Desta forma, contorna-se a restrição anteriormente apresentada para câmaras canônicas a partir de adequada excitação dos

condutores (posicionamento, amplitude, fases dos sinais etc.), buscando-se uma solução a partir de cálculos computacionais aliados ao processo de caracterização e otimização.

### 2.3 Caracterização e otimização

O objetivo da proposta é caracterizar o volume interno disponível para teste, buscando seu máximo tamanho com a máxima homogeneidade dos campos internos a ele. Os resultados desse esforço conjunto resultaram na melhoria da compreensão dessa câmara como um todo.

Conforme mencionado anteriormente, qualquer volume interno à câmara reverberante possui três atributos distintos: o seu volume, o seu campo elétrico combinado, conforme definido pela Equação 5 e os valores máximos de desvio-padrão, definidos pelas Equações 8 e 9. Quando este espaço interno atende a norma IEC, ele é denominado “volume de trabalho”. Deseja-se que este volume seja o maior possível, com alto valor de campo elétrico combinado e desvio padrão prescrito conforme a norma (para 10 MHz assume-se um valor máximo no desvio padrão de 4 dB), agregando a vantagem de se trabalhar em frequências abaixo da frequência fundamental. Trata-se de um problema em que se percebe antagonismo entre os requisitos, ou seja, a melhora de um deles implica na deterioração dos demais. Neste contexto, o problema de caracterização de uma câmara, observando o conceito de volume de trabalho, possui dois objetivos a serem maximizados, a saber: máximo volume de trabalho e máximo campo elétrico combinado; e uma restrição imposta pelo valor máximo de desvio padrão. Para buscar tais objetivos, são ajustadas as variáveis geométricas que definem o volume de trabalho.

Na presente proposta, optou-se por trabalhar com um elipsóide como figura geométrica representando o volume de trabalho. A escolha por este sólido se deu pela diminuição do esforço computacional necessário para a detecção deste volume, que necessita de seis variáveis (três semi-eixos e três deslocamentos do ponto central da câmara) para sua correta determinação.

A sequência do processo de caracterização da câmara é descrito como:

- (a) Predição dos campos eletromagnéticos, realizado com base em cálculos computacionais utilizando-se a Técnica das Integrações Finitas (FIT) (CST, 2002);
- (b) Geração de uma grade tridimensional de pontos com valores do campo elétrico ao longo do volume total da câmara;
- (c) Busca pelo máximo volume de trabalho e máximo campo elétrico por um método de otimização, obedecendo às restrições anteriormente apresentadas.

A opção pelo uso do elipsóide torna-se mais clara ao se confrontar os itens b) e c). Todo candidato a volume de trabalho sempre é testado pela

verificação dos pontos da grade que são internos a ele. O elipsóide faz com que seja necessária apenas uma comparação para saber se o ponto é interno ou externo ao volume. Outro sólido, por exemplo, um paralelepípedo, implicaria em várias comparações (seis faces, seis confrontações) para a determinação da condição de ponto interno ou externo.

Até aqui trabalhou-se na caracterização do volume de trabalho. Isto significa que existe a necessidade da resolução de um problema com dois objetivos [ $max(volume)$ ,  $max(campo)$ ], sujeito a restrições; com seis variáveis de otimização que definem geometricamente o volume de trabalho: dois raios equatoriais, um raio polar ( $r_x, r_y, r_z$ ) e três parâmetros associados à translação do elipsóide a partir do centro da câmara ( $x, y, z$ ). Agora, esta caracterização – a busca pelo melhor volume de trabalho – pressupõe que as dimensões físicas da câmara e as grandezas eletromagnéticas de sua excitação estejam definidas. Em outras palavras, existe ainda outra procura: a busca pelo melhor posicionamento dos condutores responsáveis pela excitação, pela amplitude e fase da corrente que irá alimentar estes condutores, entre outros parâmetros. Para cada arranjo dessas variáveis, tem-se um infinito número de possíveis valores para o par “volume de trabalho e homogeneidade de campo”. Pode-se, inclusive, pensar em adicionar o próprio tamanho da câmara como parâmetro desta otimização.

Em resumo, cada possível solução possui uma fronteira Pareto própria. Se cada solução possui sua própria fronteira de objetivos, como determinar o melhor arranjo? Esta fronteira apresenta o compromisso entre os objetivos. Aqui, buscou-se uma maneira de expressar a flexibilidade deste compromisso, o índice de mérito LA. Este índice consiste na determinação da área abaixo da fronteira Pareto, conforme ilustrado na Figura 2. A maximização desta área implica em uma maior liberdade de uso da câmara.

Portanto, este trabalho trata o processo de construção de uma câmara reverberante excitada por condutores com dois estágios de otimização:

1. a caracterização do compromisso do melhor volume de trabalho;
2. a maximização da flexibilidade da câmara.

## 2.4 Porque computação evolucionária – algoritmos genéticos

Otimizar significa buscar a melhor solução para um determinado problema (Coello Coello et al., 2007). Entretanto, o processo de busca da melhor solução, se bem conduzido e se as ferramentas utilizadas permitirem, pode levar a uma melhor compreensão do problema. Neste trabalho, existe a necessidade de utilizar ferramentas de otimização que permitam melhorar o entendimento da câmara. A computação evolucionária permite, através das informações produzidas durante todo o processo de evolução das soluções, que os pesquisadores envolvidos possam compreender a relação entre o arranjo de parâmetros de otimização e os objetivos (Avila, 2006). Este é um primeiro direcionamento. Entretanto, existem muitos métodos de

otimização e cada um deles alcança melhores resultados em determinados tipos de problema. A escolha do método depende de uma série de características do problema a ser otimizado, principalmente do comportamento da função que o representa, a qual muitas vezes é de difícil determinação. Para esta escolha, faz-se necessário também um bom conhecimento das ferramentas de otimização.

De acordo com as características dos problemas, pode-se classificar as ferramentas de otimização em dois grandes grupos: programação linear e programação não-linear. O primeiro grupo trata da resolução de problemas que sejam perfeitamente representados por um sistema de equações lineares. A programação não-linear trata de problemas não-lineares. As técnicas para programação não-linear podem ser subdivididas em três sub-grupos: métodos determinísticos, estocásticos e enumerativos. Os métodos determinísticos são baseados no cálculo de derivadas do problema, ou em aproximações destas. Necessitam, portanto, de alguma informação do vetor gradiente, seja procurando o ponto onde ele se anula ou usando a direção para a qual aponta. Os métodos estocásticos utilizam um conjunto de ações probabilísticas que buscam a solução ótima de maneira “aleatória orientada”, sem necessitar de qualquer informação de derivadas ou sobre o comportamento do problema. Por outro lado, os métodos enumerativos fazem uma varredura completa (busca exaustiva) de todas as possíveis soluções, o que, na maior parte dos problemas, implica em um tempo excessivo de cálculo.

Na engenharia, particularmente em eletromagnetismo, os problemas são geralmente complexos, não-lineares, de difícil representação e necessitam de métodos numéricos para se obter a solução. Por isto, ferramentas de programação não-linear estocásticas são as mais aptas para a otimização destes problemas.

Na família dos estocásticos, pode-se encontrar a computação evolucionária. Dentre muitos métodos, existe um que vem obtendo destaque devido à sua robustez, simplicidade de implementação e por poder funcionar sem a necessidade do conhecimento do comportamento do problema: os Algoritmos Genéticos (AG), apresentado inicialmente por [Holland \(1992\)](#). Inúmeras pesquisas já foram realizadas qualificando os AGs para otimização de problemas em eletromagnetismo. Uma rápida pesquisa na “*IEEE Transactions on Magnetics*”, após o ano de 2000, aponta que foram publicados centenas de artigos nesta área da engenharia.

No que se refere à otimização, existe outra importante consideração. Em muitos problemas, a solução ótima não leva em conta uma única característica que deve ser minimizada ou maximizada (abordagem mono-objetivo), mas várias. Normalmente estas características devem ser consideradas simultaneamente na busca pela melhor solução. Por exemplo, um engenheiro não pode conceber um equipamento pensando exclusivamente

em obter o melhor desempenho. Também é necessário que o produto final obedeça ao nível máximo de ruído permitido e que o consumo, assim como o custo deste equipamento, sejam os menores possíveis. Neste caso, uma abordagem multiobjetivo do problema faz-se necessária (Coelho Coelho et al., 2007; Avila, 2006). A principal diferença deste tipo de abordagem é a maneira de apresentar os resultados. Como os objetivos são frequentemente conflitantes ou antagonistas, ou seja, a melhora de um acarreta na deterioração de outro(s), a resposta de um problema multiobjetivo não corresponde a uma única solução ótima, mas sim a um grupo de soluções que caracteriza o comprometimento entre os diversos objetivos (soluções não-dominadas), conforme ilustra a Figura 2. Este conjunto configura uma fronteira de soluções, se elas forem as melhores – não as melhores encontradas, mas as melhores soluções possíveis para o problema – neste caso elas são conhecidas na literatura como soluções Pareto.

Entender este comprometimento leva o projetista/engenheiro a compreender melhor seu problema, o que lhe permite obter um produto melhor. O AG se adapta bem à resolução de problemas multiobjetivos, pois trabalha simultaneamente com uma população de soluções possíveis, o que facilita a obtenção das soluções não-dominadas.

Por todo o contexto apresentado, a escolha pela computação evolucionária com algoritmos genéticos é natural para problemas do eletromagnetismo. Acrescenta-se ainda, mais apropriadamente ao problema da câmara apresentado, que a análise eletromagnética é realizada numericamente, sem acesso a qualquer informação sobre as derivadas do problema. Ainda, por se tratar de um problema com campos reverberantes, é grande a probabilidade de múltiplas soluções diferentes com valores de objetivos semelhantes – conceito de nichos (Avila et al., 2006a). Assim, a ferramenta de otimização escolhida deve contemplar todas estas características. Optou-se pelos métodos desenvolvidos pelos autores deste trabalho (Avila et al., 2006a,b).

### 3. Metodologia

De forma breve, conforme ilustrado na Figura 2, o procedimento adotado segue os seguintes passos:

1. Inicia-se o processo de otimização mono-objetivo (AG) com a definição das variáveis de projeto: limites para posicionamento dos condutores, amplitudes e defasagens das correntes, etc. (Assegurando o correto funcionamento e a adequação aos testes de compatibilidade pretendidos);
2. Para cada possível solução, realizam-se os cálculos de propagação de onda e a determinação do campo elétrico no interior da câmara. Isto é feito através do software Microwave Studio (CST, 2002), utilizando-se a Técnica de Interações Finitas;

3. Possuindo-se o comportamento do campo elétrico, utiliza-se a otimização multiobjetivo (AGMO) para se conhecer o compromisso entre o volume de trabalho e a homogeneidade do campo - uma fronteira não-dominada para cada solução possível. O “volume de trabalho” é o espaço interno da câmara que atende as normas, conforme comentado anteriormente;
4. Com base na fronteira encontrada, calcula-se o índice de mérito LA, que é igual à área sob a fronteira não-dominada, restrita a um volume e a um campo mínimos, conforme ilustrado na Figura 2. O índice LA representa a “flexibilidade” ou “estabilidade” do arranjo construtivo em análise, conforme também já explicado; e
5. Por fim, a decisão da melhor solução é feita com base na maximização do índice de mérito LA, realizada por AG mono-objetivo.

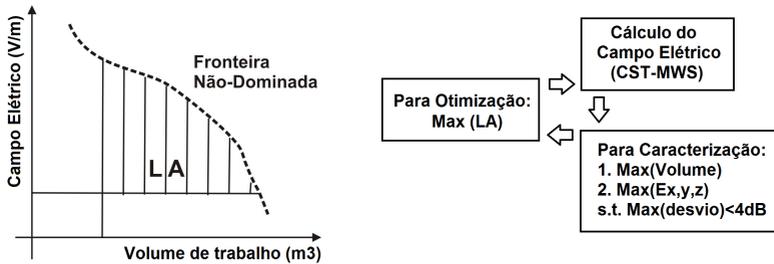


Figura 2. Procedimento híbrido para caracterização e otimização. (esquerda) Maximização de LA. (direita) Caracterização da câmara.

## 4. Resultados

Como exemplo de resolução do processo de construção proposto, apresentam-se os últimos resultados obtidos. O problema foi inicialmente parametrizado da seguinte forma, para otimização:

- A dimensão geométrica da câmara foi fixada em 5.20m x 4.55m x 2.70m;
- As variáveis de otimização são os posicionamentos dos condutores de excitação ( $x_1, y_1; x_2, z_2; y_3, z_3$ ), os quais possuem como limites toda a extensão da câmara, mas limitado a um afastamento de no mínimo 0,05m das paredes;
- O objetivo da otimização é a maximização do índice LA;
- Como restrição, trabalhou-se com valores mínimos de campo e volume para o cálculo do índice LA ( $1V/m$  e  $3m^3$ , respectivamente);

- Foram estipulados 50 indivíduos e 20 gerações, tendo como critério de parada um índice de diversidade mínima da população; e
- Probabilidade de cruzamento igual a 80% e probabilidade de mutação igual a 15%.

Para caracterização:

- As variáveis de otimização são as dimensões do elipsóide “volume de trabalho”  $[r_x, r_y, r_z, x, y, z]$ , tendo como limites mínimos  $[0, 0, 0, -2,6, -2,3, -1,35]$  e máximos de  $[2,6, 2,3, 1,35, 2,6, 2,3, 1,35]$ ;
- Os objetivos da caracterização é a maximização do “volume de trabalho” e a maximização da homogeneidade do campo dentro deste volume;
- Foram estipulados 100 indivíduos e 30 gerações, tendo como critérios de parada a qualidade da fronteira Pareto, medida pela evolução das suas soluções não-dominadas e o espaçamento entre elas;
- Probabilidade de cruzamento igual a 90%, probabilidade de mutação igual a dez por cento. Valores maiores que para a otimização, em virtude da dificuldade do problema. A intenção é permitir uma varredura mais ampla do espaço de busca.

Cabe informar que, em ambos os processos, foram utilizadas ferramentas de elitismo (permanência no processo evolutivo das melhores soluções sempre) e de nichos (cuidado especial para soluções bem diferentes que possuem valores de objetivos semelhantes, assim como também soluções bem semelhantes que possuem valores de objetivos diferentes). Os artigos [Avila et al. \(2006a\)](#) e [Avila et al. \(2006b\)](#) apresentam em detalhes estes cuidados.

O melhor indivíduo obtido (com índice de mérito LA igual a 2,25) tem como posicionamento dos seus condutores de excitação as seguintes coordenadas:  $x_1=3,72\text{m}$ ;  $y_2=2,46\text{ m}$ ;  $x_2=0,62\text{ m}$ ;  $z_2=4,12\text{ m}$ ;  $y_3=0,62\text{ m}$ ;  $z_3=4,23\text{m}$ . Como já explicado, para esta configuração dos condutores existem vários compromissos possíveis entre a dimensão “volume de trabalho” e o valor de campo elétrico mínimo em seu interior – vários elipsóides possíveis – isto é demonstrado pela fronteira não-dominada apresentada na Figura 3. Os valores mínimos para desta solução são  $1,007\text{V/m}$  e  $3,11\text{m}^3$  – para o cálculo do índice LA, os valores mínimos considerados foram  $1\text{V/m}$  e  $3\text{m}^3$ . Além dos extremos estarem próximos dos mínimos desejados, esta solução foi quem atingiu o maior índice LA em função também de sua estabilidade, o que pode ser observado pela pequena inclinação da fronteira não-dominada. Obviamente, esta solução obedece às restrições de campo impostas pelas normas, sendo que os valores encontrados estão limítrofes ao permitido, como também mostra a Figura 3.

A título de curiosidade: o volume em destaque (circulado), aquele que se encontra mais distante dos 4 dB, portanto, aquele talvez menos sujeito a

desrespeitar a norma em caso de pequenas alterações construtivas ou erro no posicionamento do equipamento em teste dentro do volume de trabalho. Ressalta-se também que o posicionamento dos volumes está entre o centro e o canto oposto do tripé formado pelos condutores. Isto é coerente, uma vez que longe das excitações o comportamento do campo elétrico tende a ser mais homogêneo. Importante dizer que o valor do campo elétrico no interior da câmara poderia, ainda, ser amplificado a partir da elevação da amplitude do sinal de entrada, que depende, basicamente, da potência do amplificador de RF utilizado.

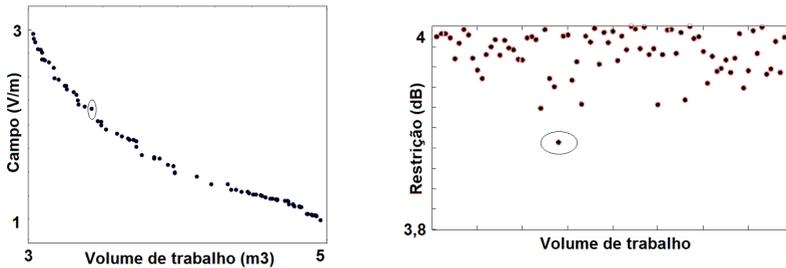


Figura 3. Caracterização da solução com maior índice LA encontrado. (esquerda) Caracterização da melhor configuração encontrada. (direita) Comportamento dos volumes de trabalho da melhor solução encontrada em função da restrição de 4dB.

## 5. Discussão e conclusões

O presente trabalho apresenta uma contribuição para uma compreensão adequada do funcionamento de câmaras reverberantes de modos misturados não convencionais, cuja excitação é realizada a partir de um arranjo de condutores. Algoritmos Genéticos mono e multiobjetivo foram as ferramentas utilizadas, as quais permitiram a análise de um grande número situações e suas possíveis soluções. Isto possibilitou a equipe fomentar discussões a respeito de câmaras reverberantes aplicadas a testes de compatibilidade eletromagnética.

Cabe destacar, ainda, que o mais interessante neste trabalho não é só obter a melhor solução. A intenção é proporcionar ao engenheiro não apenas uma vasta gama de possíveis soluções (o que facilita negociações – testes para vários tipos de equipamentos, com funcionalidades e tamanhos físicos diferentes, por exemplo), mas também a possibilidade de conhecer melhor o próprio problema.

Com relação a trabalhos futuros, pensa-se inserir no processo novas variáveis de otimização, como, por exemplo, a amplitude e a fase das exci-

tações. Também se pode buscar uma melhor configuração das excitações no que diz respeito à disposição dos condutores não alinhados com os eixos, ou ainda seguindo curvas, e até mesmo envolvendo um número maior de condutores. A própria dimensão física da câmara pode ser variável de otimização com o objetivo de se ter um equipamento com tamanho reduzido, mas com um volume de trabalho adequado. Estudos de sensibilidade quanto ao posicionamento exato do equipamento em teste dentro do volume de trabalho também são um caminho interessante a ser trilhado.

## Agradecimentos

Os autores transmitem seus sinceros agradecimentos a todos os colegas e colaboradores da EPUSP, particularmente do LMAG. Em especial, pela confiança depositada neste trabalho e pela oportunidade de conduzir as atividades de pesquisa em um centro de excelência. Os autores também deixam impresso o reconhecimento ao PROCAD/CAPES, ao CAPES/COFECUB e à FAPESP pelo suporte financeiro indispensável para o desenvolvimento destas pesquisas. Por fim, os autores agradecem às respectivas instituições de origem por permitirem a alocação de tempo para pesquisa.

## Referências

- ANATEL, , *Resolução Nº 237 – Regulamento para a certificação de equipamentos de telecomunicações quanto aos aspectos de compatibilidade eletromagnética*. Brasília, DF: Agência Nacional de Telecomunicações, 2000.
- Avila, S., *Otimização Multiobjetivo e Análise de Sensibilidade para a Concepção de Dispositivos e Sistemas*. Tese de doutorado, Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina (e co-tutela com École Centrale de Lyon), Florianópolis, SC, 2006.
- Avila, S.; Krähenbühl, L. & Sareni, B., A multi-niching multi-objective genetic algorithm for solving complex multimodal problems. In: *Proceedings of the 9<sup>th</sup> International Workshop on Optimization and Inverse Problems in Electromagnetism*. Sorrento, Italy, 2006a.
- Avila, S.; Lisboa, A.; Krähenbühl, L.; Carpes Jr., W.; Vasconcelos, J.; Saldanha, R. & Takahashi, R., Sensitivity analysis applied to decision making in multi-objective evolutionary optimization. *IEEE Transactions on Magnetics*, 42(4):1103–1106, 2006b.
- Avila, S.; Santos Jr., M.; Weinzierl, D.; Krähenbühl, L.; Perrussel, R.; Vollaire, C.; Sartori, C.; Lebensztajn, L. & Cardozo, J., Maximum working volume evaluation in a non-canonical reverberation chamber. *IEEE Transactions on Magnetics*, 45(3):1646–1649, 2009.

- Coello Coello, C.; Lamont, G. & Van Veldhuizen, D., *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2a edição. New York, EUA: Springer Science, 2007.
- CST, , *CST Microwave Studio, Advanced Topics*. Framingham, EUA: Computer Simulation Technology, 2002. [Http://www.cst.com](http://www.cst.com).
- Holland, J., *Adaptation in Natural and Artificial Systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Cambridge, USA: MIT Press, 1992.
- IEC, , *P-IEC 61000-4-21 – Electromagnetic compatibility (EMC) – Part 4-21: Testing and measurement techniques - Reverberation chamber test methods*. 1a edição. Geneva, Switzerland: International Electrotechnical Commission, 2003. 224 p.
- Santos Jr., M., *Avaliação de desempenho de câmaras excitadas via linha de transmissão aplicáveis à compatibilidade eletromagnética*. Tese de doutorado, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, 2009.
- Santos Jr., M.; Avila, S.; Sartori, C.; Weinzierl, D.; Krähenbühl, L.; Lebensztajn, L. & Cardozo, J., Conductor position optimization of a transmission line excitation chamber. *IEEE Transactions on Magnetics*, 46(8):3261–3264, 2010.
- Weinzierl, D., *Estudo de Câmaras de Teste de Compatibilidade Eletromagnética (CEM) através do Método da Modelagem por Linhas de Transmissão (TLM)*. Tese de doutorado, Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, SC, 2004.
- Weinzierl, D.; Sartori, C.; Perotoni, M.; Cardoso, J.; Kost, A. & Heleno, E., Numerical evaluation of noncanonical reverberation chamber configurations. *IEEE Transactions on Magnetics*, 44(6):1458–1461, 2008.

## Notas Biográficas

**Sérgio L. Avila** é professor Doutor do Departamento de Eletrotécnica do Instituto Federal de Santa Catarina e atuou neste trabalho como bolsista de pós-doutoramento no Laboratório de Eletromagnetismo Aplicado (LMAG) da Universidade de São Paulo (USP). Email: sergio.avila@ifsc.edu.br

**Mário A. Santos Jr.** é gerente de Pesquisa e Desenvolvimento do Centro Tecnológico da Marinha em São Paulo (CTMSP/MB) e finalizou seu doutoramento no LMAG/USP em 2010. Email: mario.alves.jr@ctmsp.mar.mil.br

**Djonny Weinzierl** é professor Doutor do Departamento de Engenharia Elétrica do Centro Universitário – Católica de Santa Catarina e atuou neste trabalho como bolsista de pós-doutoramento no LMAG/USP. Email: weinzierl@unerj.br

**Luiz Lebensztajn** é professor Associado da Escola Politécnica da Universidade de São Paulo e atuou neste trabalho como orientador no LMAG. Email: luiz.lebensztajn@poli.usp.br

**Carlos A. F. Sartori** é professor Doutor da Escola Politécnica da Universidade de São Paulo e atuou neste trabalho como orientador no LMAG. Email: sartori@pea.usp.br

**José R. Cardoso** é professor Titular da Escola Politécnica da Universidade de São Paulo e atuou neste trabalho como orientador no LMAG. Email: jose.cardoso@poli.usp.br

**Laurent Krähenbühl** é directeur de Recherche do Grupo Ampère CNRS da Ecole Centrale de Lyon e atuou nesse trabalho como orientador em missão de trabalho no LMAG. Email: laurent.krahenbuhl@ec-lyon.fr



# Computação Evolucionária Aplicada ao Problema da Recarga de Reatores Nucleares

Alan M.M. de Lima,\* Andressa dos Santos Nicolau,  
Ioná Maghali Santos de Oliveira, Jose Antonio Carlos Canedo Medeiros,  
Márcio Henrique da Silva e Roberto Schirru

---

**Resumo:** O objetivo deste capítulo é apresentar o Problema da Recarga de Reatores Nucleares (PRRN), mostrando a viabilidade da utilização de alguns métodos da Inteligência Computacional na solução deste problema. Os métodos aqui abordados correspondem às estratégias de otimização por colônia de formigas, colônia de abelhas, enxames de partículas, algoritmos genéticos, além de uma abordagem com inspiração quântica para o algoritmo de colônia de formigas. São introduzidos os principais aspectos do PRRN, bem como as principais características de cada método utilizado. Os resultados apresentados comprovam a viabilidade e eficiência destes métodos como técnicas de solução de problemas da área nuclear.

**Palavras-chave:** Recarga de reator nuclear, Otimização, Algoritmos Evolucionários, Algoritmos de Inspiração Quântica.

**Abstract:** *The objective of this chapter is to present the Nuclear Reactor Reload Problem (NRRP), demonstrating the feasibility of using some Computational Intelligence methods to solve this problem. The methods discussed here correspond to optimization strategies by ant colony, colony of bees, particle swarm, genetic algorithms, and a quantum-inspired approach for the ant colony algorithm. Here we introduce the main aspects of NRRP as well as the main features of each method. The results presented here demonstrate the feasibility and efficiency of these methods as techniques for problem solving in the nuclear area.*

**Keywords:** *Nuclear reactor reload, Optimization, Evolutionary algorithms, Quantum inspired algorithm.*

---

\* Autor para contato: alan@lmp.ufrj.br

## 1. Introdução

O Problema da Recarga do Reator Nuclear (PRRN) também conhecido como Otimização do Projeto de Padrões de Carregamento são denominações para o problema de otimização associado à operação de recarga do combustível em um reator nuclear. Trata-se de um problema clássico e uma questão de grande interesse econômico e extrema relevância para a Engenharia Nuclear, que tem sido estudado por mais de 40 anos.

A recarga do combustível nuclear ocorre periodicamente em usinas nucleares e é realizada toda vez que a queima dos elementos combustíveis (EC) no núcleo do reator atinge um determinado valor em que não é mais possível manter o reator crítico, produzindo energia à potência nominal. Desta forma, os EC são descarregados do núcleo e armazenados em uma piscina de combustíveis usados, onde os EC com baixa concentração de  $^{92}\text{U}_{235}$  são mantidos definitivamente nesta piscina, enquanto os EC com maiores concentrações de  $^{92}\text{U}_{235}$ , remanescentes do ciclo anterior e, EC novos comporão o núcleo do ciclo seguinte. Por exemplo, na operação de recarga de um Reator a Água Pressurizada (*Pressurized Water Reactor* – PWR), ao final de um ciclo de operação, aproximadamente 1/3 dos elementos combustíveis são substituídos por elementos novos, que juntamente com os 2/3 restantes vão compor um novo padrão de carregamento.

A solução mais comum para a este problema de otimização consiste na determinação de um padrão de carregamento que maximize a vida útil do combustível nuclear, permitindo o aumento do ciclo de queima do combustível, trazendo ganhos operacionais e econômicos relacionados ao funcionamento da usina, sujeito a restrições de segurança. Porém, a dificuldade de encontrar esta solução relacionada com o grande número de possíveis combinações de elementos combustíveis que podem compor o núcleo do próximo ciclo, define o PRRN como um problema combinatório altamente complexo, pertencente à classe dos problemas não polinomiais (NP) completos, onde a dificuldade cresce exponencialmente de acordo com o número de elementos combustíveis considerados.

Além da otimização manual, realizada por especialista em padrões de carregamento, várias técnicas de otimização têm sido aplicadas ao PRRN com o objetivo de automatizar este processo. Neste capítulo são apresentados os resultados da aplicação das técnicas de otimização por colônia de formigas, colônia de abelhas, enxames de partículas, algoritmos genéticos, além de uma abordagem com inspiração quântica para o algoritmo de colônia de formigas, para a solução deste problema.

O objetivo deste capítulo não é fazer um estudo comparativo entre os diferentes algoritmos mas, sim, mostrar a viabilidade do uso dos mesmos no desenvolvimento efetivo de trabalhos em uma área da engenharia, além de destacar as principais diferenças entre eles na otimização do problema da recarga nuclear.

## 2. O Problema da Recarga Nuclear

### 2.1 Aspectos básicos do problema

Os reatores nucleares do tipo PWR geram energia através de reações de fissão nuclear em condições controladas. Eles são projetados para conseguir uma reação em cadeia auto-sustentada, utilizando o processo de fissão dos átomos de urânio ( $^{92}\text{U}_{235}$  enriquecido a cerca 3%). A principal finalidade é a produção de energia utilizável na forma de eletricidade.

Enquanto for possível sustentar a reação em cadeia mantendo o reator crítico, pode-se dizer que a usina está operando a sua plena potência. Caso contrário, torna-se necessário o seu desligamento para que o núcleo seja reabastecido. Este procedimento é denominado recarga nuclear. A recarga do combustível nuclear é um processo que ocorre sempre que a energia produzida pela queima do combustível no núcleo do reator atinja certos níveis, de modo a não ser mais possível manter a central nuclear operando à plena potência. Neste momento, parte dos elementos combustíveis é substituída.

O intervalo entre as recargas é conhecido como ciclo de operação. O processo de recarga do novo ciclo irá envolver ECs novos e reutilizados, distribuídos no núcleo do reator de acordo com critérios operacionais e de segurança. O combustível reutilizado corresponde ao que foi descarregado de ciclos anteriores, que foram parcialmente queimados, mas que ainda podem ser reaproveitados.

O problema relacionado com a operação de recarga consiste, portanto, na determinação de como os combustíveis novos e reutilizados deverão ser combinados e redistribuídos no núcleo do reator, de modo a otimizar o próximo ciclo de operação. Isto se deve ao fato de que a queima de ECs não ocorre de maneira homogênea, sendo necessário encontrar uma configuração ótima de embaralhamento dos mesmos, conhecida como padrão de recarga.

Do ponto de vista econômico é importante ressaltar que padrões de recarga que maximizam a concentração de boro permitem um ciclo de operação mais longo. Tendo em vista que o comprimento de cada ciclo é medido em Dias Efetivos à Plena Potência (DEPP), do inglês, *Effective Fuel Power Days* (EFPD) e considerando que, para Angra 1, a quantidade correspondente a 4ppm de Boro solúvel é consumido durante 1 DEPP (Chapot et al., 1999), configurações que maximizem a concentração de Boro são financeiramente desejáveis.

No caso particular da Usina Nuclear Angra 1, localizada no Sudeste do Brasil, o núcleo do reator é composto por 121 elementos combustíveis, resultando em aproximadamente  $10^{200}$  possíveis combinações de padrões de carregamento, que devido à existência de restrições de posicionamento dos elementos combustíveis, relacionadas com a geometria do núcleo, pode ser reduzido para um total de aproximadamente  $10^{25}$  possíveis arranjos ou

padrões de carregamento. Com o uso de computadores e códigos de física de reatores atuais seriam necessários cerca de  $10^{19}$  anos para que todas estas combinações fossem testadas, tornando o problema atraente no sentido do desafio de desenvolver novos algoritmos para encontrar soluções próximas a ótima com o menor número possível de avaliações.

Diversos métodos de otimização têm sido aplicados ao PRRN, ao longo dos anos e observa-se que a aplicação de técnicas de otimização, em especial, as chamadas metaheurísticas de otimização, tais como técnicas de Arrefecimento Simulado (Kropaczek & Turinsky, 1991), Busca Tabu (Lin et al., 1998), Algoritmos Genéticos (Chapot et al., 1999), Aprendizado Incremental Baseado em Populações (Schirru et al., 2006), Otimização com Colônia de Formigas (de Lima et al., 2008), Otimização com Colônia de Abelhas (Oliveira et al., 2009), Otimização com Enxame de Partículas (Meneses et al., 2009) e Otimização Quântica de Colônia de Formigas (da Silva et al., 2011), têm apresentado melhoras significativas nos resultados obtidos na otimização deste problema.

### 3. Algoritmos de Inteligência Computacional Baseados em Computação Evolucionária e Inteligência de Enxames

A Inteligência Computacional é uma linha de pesquisa situada no contexto da Ciência da Computação e tem como objetivo desenvolver, avaliar e aplicar técnicas na criação de algoritmos inteligentes. Pode-se dizer que algoritmo é todo e qualquer procedimento capaz de solucionar um determinado problema, através de passos previamente elaborados e, que otimizar significa aperfeiçoar algum processo. Desta forma, algoritmos de otimização podem ser entendidos como procedimentos previamente elaborados para resolução de um determinado problema. Dentre os algoritmos de Inteligência Computacional, podemos destacar os algoritmos baseados em processos naturais, como é o caso dos algoritmos de computação evolucionária e, os baseados em interação entre grupos sociais (enxames) como é o caso dos algoritmos da Inteligência de Enxames.

Esta seção apresenta as características do principal algoritmo de Computação Evolucionária – Algoritmos Genéticos (Holland, 1975), bem como os de Inteligência Enxames – Enxame de Partículas (Kennedy & Eberhart, 1995), Algoritmo de Abelhas (Karaboga, 2005), Colônia de Formigas (Dorigo & Gambardella, 1996), além de uma versão quântica do algoritmo de Colônia de Formigas (Wang et al., 2008), que foram aplicados ao PRRN abordado neste capítulo.

#### 3.1 Algoritmo de inteligência computacional baseado em computação evolucionária – algoritmos genéticos (AG)

Os algoritmos genéticos são uma família de modelos computacionais, inspirados no mecanismo de seleção natural e no princípio da seleção natural de

Darwin. Desenvolvidos por [Holland \(1975\)](#) e popularizados por [Goldberg \(1989\)](#), são implementados como uma simulação computacional, que consiste em gerar através de regras de transições probabilísticas, um grande número de indivíduos (população), de forma a promover uma varredura tão extensa quanto necessária do espaço de soluções. Normalmente incorporam uma solução potencial numa estrutura semelhante à de um cromossomo, onde cada cromossomo representa um ponto no espaço de soluções do problema de otimização ([Holland, 1975](#)).

Os algoritmos genéticos diferem basicamente dos algoritmos tradicionais de otimização em quatro aspectos:

- (a) baseiam-se em uma codificação do conjunto de soluções possíveis, e não nos parâmetros da otimização em si;
- (b) os resultados são apresentados como uma população de soluções e não como uma única solução;
- (c) não necessitam de nenhum conhecimento derivado do problema, apenas de uma forma de avaliação do resultado (função objetivo);
- (d) usam transições probabilísticas e não regras determinísticas.

Os principais conceitos dos algoritmos genéticos são: Cromossomo (genótipo) – possível solução para o problema e pode ser representado de forma: binária, inteira ou real. É composto de  $l$  genes específicos, onde cada  $i$ -ésimo gene tem um conjunto de  $k_i$  alelos ([Holland, 1975](#)). Gene – representação codificada de cada parâmetro de acordo com o alfabeto utilizado (binário inteiro ou real); Fenótipo – cromossomo decodificado; População – conjunto de pontos (cromossomos) no espaço de busca; Geração – iteração completa do algoritmo genético que gera uma nova população; Aptidão bruta – saída gerada pela função objetivo para um cromossomo da população; Aptidão normalizada – aptidão bruta normalizada, entrada para o algoritmo de seleção; Aptidão máxima – melhor cromossomo da população corrente e Aptidão média – aptidão média da população corrente.

O procedimento de evolução do algoritmo genético a cada iteração corresponde à aplicação de um conjunto de quatro operações básicas, descritas a seguir.

### 3.1.1 Cálculo da função objetivo

A função objetivo incorpora os parâmetros envolvidos no problema a ser otimizado. Com o cálculo da função objetivo tem-se uma medida de aptidão de cada cromossomo, para a solução do problema e, este valor é usado na fase de seleção dos cromossomos para a reprodução.

### 3.1.2 Seleção

O processo de seleção consiste em selecionar cromossomos de melhores aptidões para gerar, através do processo de cruzamento, novos cromossomos

que compõem a próxima geração. São muitos os métodos para selecionar o melhor cromossomo, pode-se citar, por exemplo, a seleção de Boltzman, a seleção pelo método da roleta, a seleção por campeonato, a seleção por classificação, a seleção por estado estacionário e, outras (Goldberg, 1989). Um dos métodos mais usados nos algoritmos genéticos simples, para a implementação do método de seleção é o método de seleção por roleta. Neste método cada cromossomo ocupa, em uma roleta, uma área proporcional ao valor de sua aptidão. Assim obtém-se uma aptidão relativa que pode ser interpretada como a probabilidade de seleção de um cromossomo.

### 3.1.3 Cruzamento

O processo de cruzamento tem como principal objetivo perpetuar o material genético (bits) dos cromossomos mais aptos e consiste na recombinação de alelos através da troca de segmentos entre pares de cromossomos. O processo de cruzamento é aplicado aos cromossomos com uma probabilidade determinada pela taxa de cruzamento  $p_c$ . Este processo pode ser implementado de vários modos, sendo os mais comuns o cruzamento de um ponto e o cruzamento de dois pontos. A diferença entre o cruzamento de dois pontos e o cruzamento de um ponto é que ao invés de escolher apenas um ponto de cruzamento escolhem-se randomicamente dois. O cruzamento de um ponto é realizado da seguinte forma: primeiramente dois cromossomos da população atual são selecionados; a seguir seleciona-se (de forma aleatória) um ponto de cruzamento sobre a sequência de bits (cromossomo) e, em seguida a troca de segmentos é realizada a direita do ponto de cruzamento escolhido. O resultado deste processo é a criação de dois novos cromossomos que apresentam características dos cromossomos que os geraram.

### 3.1.4 Mutação

O processo de mutação tem o papel de introduzir e manter a diversidade genética da população. Desta forma, pode ser definido como um distúrbio introduzido no gene, que altera a direção de busca do algoritmo. Tem por objetivo assegurar uma maior varredura do espaço de busca e prevenir o aprisionamento da população em falsos ótimos locais. O operador de mutação é aplicado aos cromossomos com uma probabilidade dada pela taxa de mutação  $p_{mr}$ , geralmente se utiliza uma taxa de mutação pequena, para que o algoritmo não seja transformado em um processo de busca aleatória. O processo de mutação é realizado da seguinte forma: primeiramente seleciona-se aleatoriamente um cromossomo da população atual; em seguida seleciona-se (também de forma aleatória) um alelo deste cromossomo e, altera-se o valor do alelo selecionado. O resultado deste processo é a criação de indivíduos formados por alelos não previamente testados.

Ao fim destas quatro operações cria-se uma nova população, chamada de geração. Espera-se que esta nova população criada, represente uma melhor solução para o problema que a população anterior.

Ao longo da evolução do algoritmo genético percebe-se que certos padrões (bits) que possuem aptidões mais altas tendem a aumentar sua representatividade de geração para geração, seguindo um teorema que se baseia nas probabilidades de “sobreviverem” ao cruzamento e mutação. Um grupo de padrões que representa determinada classe de cromossomos é chamado de esquemas ou modelo de similaridade. Tais padrões (esquemas) são explicados pela Teoria dos Esquemas e dão origem ao Teorema Fundamental dos Algoritmos Genéticos (Equação 1), que rege a evolução dos esquemas considerando os processos de seleção, cruzamento e mutação.

$$m(H, t + 1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \cdot \left[ 1 - p_c \cdot \frac{d(H)}{l - 1} - o(H) \cdot p_m \right] \quad (1)$$

onde,  $m(H, t)$  é o número  $m$  de indivíduos pertencentes ao esquema  $H$ , no tempo  $t$ ,  $f(H)$  é a aptidão bruta do cromossomo,  $\bar{f}$  é a aptidão média da população,  $p_c$  é a taxa de cruzamento,  $l$  é o comprimento dos cromossomos e  $l - 1$  é o número máximo de pontos de corte de um cromossomo de comprimento  $l$ ,  $d(H)$  é o comprimento definido do esquema,  $o(H)$  é a ordem do esquema e  $p_m$  é a probabilidade de mutação considerando um ponto de corte.

## 3.2 Algoritmos de inteligência computacional baseados em inteligência de enxames

### 3.2.1 Enxame de partículas – PSO

A técnica de otimização por Enxame de partículas (*Particle Swarm Optimization* – PSO) é uma técnica de computação estocástica baseada em Inteligência de Enxames. Foi desenvolvida por Kennedy & Eberhart (1995) e implementa a metáfora do comportamento social, da interação entre bando de pássaros ou cardume de peixes quando em busca de alimento em uma determinada região. Verifica-se que o comportamento do grupo é influenciado pela experiência acumulada por cada indivíduo e pela experiência acumulada pelo grupo.

No PSO, cada partícula corresponde a um ponto do espaço de soluções do problema, que é avaliada a cada iteração de acordo com uma função objetivo. Uma partícula é a representação de um objeto que se move ao longo do espaço de busca, impulsionado por comparações entre sua posição atual, sua melhor posição atingida até o momento e a posição obtida pela melhor partícula do enxame. Estes termos, devidamente equacionados, permitem o cálculo de suas novas posições e velocidades a cada iteração, direcionando o movimento das partículas. Cada uma das partículas realiza

um balanço entre a capacidade de exploração e a de tirar proveito da exploração realizada pelas outras. O balanceamento destas características pode fazer grande diferença em um processo de otimização. Pouca capacidade de exploração pode significar convergência prematura para um resultado não necessariamente ótimo, e pouca capacidade de obter vantagem do sucesso alheio pode fazer com que o enxame não convirja para nenhum bom resultado. Desta forma, o sucesso da busca realizada pelo enxame está condicionado ao balanço entre individualidade e coletividade (Meneses et al., 2009).

Cada partícula é representada por um vetor de dimensão igual à dimensão do espaço de busca, que é inicializado com velocidade e posição aleatórias. Ela carrega consigo informação sobre a melhor posição por ele já visitada e a informação da melhor posição já alcançada pelo enxame, e a cada iteração os indivíduos da população são avaliados, e as melhores posições de cada indivíduo e do enxame são armazenadas.

Assim como no AG o PSO é inicializado por uma população de soluções randômicas que buscam o ótimo através da atualização das iterações, bastando para sua aplicação à existência de uma função objetivo. Porém, ao contrário do AG onde o aprendizado é feito através dos operadores de cruzamento e mutação, no PSO o aprendizado é realizado através do ajuste das equações de velocidade e posição.

Seja um enxame com  $P$  partículas em um espaço de busca  $n$ -dimensional. Cada partícula  $i$  tem uma posição  $X_i(t) = [x_{i1}, x_{i2}, \dots, x_{in}]$  e uma velocidade  $V_i(t) = [v_{i1}, v_{i2}, \dots, v_{in}]$  em uma iteração  $t$ , através da dimensão  $j$ , atualizadas de acordo com as Equações 2 e 3.

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_1(pbest(t)_{ij} - x_{ij}(t)) + c_2r_2(gbest(t)_j - x_{ij}(t)) \quad (2)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \text{ onde } i = 1, 2, \dots, P_{ij} = 1, 2, \dots, n \quad (3)$$

No segundo membro da Equação 2, o primeiro termo representa a influência do movimento da partícula, agindo como uma memória do comportamento anterior da partícula, o segundo termo representa a cognição individual, onde a partícula  $i$  compara sua posição atual com a sua melhor posição  $pbest_i(t)$ , e o terceiro termo representa o aspecto social do aprendizado, baseado em uma comparação entre a posição da partícula e o melhor resultado obtido pelo enxame ( $gbest$ ). A Equação 3 descreve como as posições são atualizadas. Ambos,  $c_1$  e  $c_2$  são constantes de aceleração:  $c_1$  está relacionado com a cognição individual, enquanto que  $c_2$  está relacionado ao aprendizado social,  $r_1$  e  $r_2$  são números aleatórios distribuídos no intervalo  $[0,1]$  e  $w$  é conhecido como fator de inércia e estabelece o compromisso entre diversidade e refinamento de busca. Seu valor pode ser fixo ou cair linearmente.

As posições  $x_{ij}(t)$ , atualizadas pelas e Equações 2 e 3, são avaliadas por uma função objetivo do problema  $f(x_{ij}(t))$ , que no caso dos estudos aqui apresentados, corresponde à Equação 17. Os vetores de posição  $gbest = [gbest_1 \ gbest_2 \dots \ gbest_n]$  e  $pbest_i = [pbest_{i1} \ pbest_{i2} \dots \ pbest_{in}]$  são atualizados dependendo da informação adquirida pelo enxame, construindo seu conhecimento no espaço de busca ao longo das iterações.

### 3.2.2 Sistema de colônia de formigas – SCF

O SCF (Dorigo & Gambardella, 1996) foi desenvolvido para aplicação em problemas de otimização combinatória do tipo NP-difícil, ou seja, a dificuldade deste problema cresce exponencialmente com o número de elementos, como é o caso do Problema do Caixeiro Viajante (PCV). O PCV é um dos mais importantes e estudados problemas de otimização. A importância de encontrar um algoritmo para este problema reside no fato de muitos problemas que ocorrem na ciência da computação, e em muitas outras áreas, poderem ser modelados a partir do mesmo, como é o caso do problema da recarga nuclear.

Quando se aplica o SCF ao PCV, o agente  $k$  é uma formiga artificial que se move de cidade em cidade, até retornar à cidade inicial, construindo um mapa de percurso. A este agente  $k$  está associada uma lista de cidades a visitar  $J_k(r)$ , onde  $r$  é a cidade atual do agente  $k$ . A palavra agente está diretamente relacionada com formiga.

O agente  $k$  escolhe uma cidade para deslocar-se através de uma regra de transição de estado descrita pela Equação 4 a seguir:

$$s = \begin{cases} \max \{ [FE(r, s)]^\delta \times [HE(r, s)]^\beta \} & se \ q \leq q_0 \\ Roleta & se \ q > q_0 \end{cases} \quad (4)$$

onde *Roleta* é dada pela distribuição de probabilidades pseudo-randômica-proporcional, mostrada na Equação 5.

$$Roleta = \begin{cases} \frac{[FE(r, s)]^\delta \times [HE(r, s)]^\beta}{\sum_{z \in J_k(r)} [FE(r, z)]^\delta \times [HE(r, z)]^\beta} & se \ s \in J_k(r) \\ 0 & se \ s \notin J_k(r) \end{cases} \quad (5)$$

onde  $FE(r, s)$  é um valor real e positivo associado ao arco  $(r, s)$  e  $FE$  é uma matriz quadrada cuja ordem é definida pelo número de cidades. Os valores  $FE(r, s)$  desta matriz são modificados em tempo de execução do algoritmo e indicam quanto é vantajoso o movimento para a cidade  $s$  quando se está na cidade  $r$ . Estes valores representam o rastro de feromônio artificial depositado pelas formigas na construção de seus caminhos.

$HE(r, s)$  é o valor da função heurística relativa ao movimento da cidade  $r$  para a cidade  $s$ . Esta função expressa uma informação puramente

local, independente da viagem da formiga como um todo. Neste caso, a função heurística é a distância ente as cidades, onde a menor distância é considerada melhor, então usa-se o inverso da distância.

Os parâmetros  $\delta$  e  $\beta$  pesam a importância relativa entre o aprendizado dos agentes, com os valores feromônio  $FE$ , e os valores heurísticos  $HE$ ,  $q$  é um valor escolhido aleatoriamente com probabilidade uniforme dentro do intervalo  $[0,1]$  e  $q^0$  ( $0 \leq q^0 \leq 1$ ) é um parâmetro de entrada do algoritmo e expressa o quanto o algoritmo vai ser determinístico ou probabilístico. Normalmente usa-se  $q^0 = 0,9$ . Quando  $q \leq q^0$  a regra de decisão utiliza o conhecimento disponível sobre o sistema, ou seja, o conhecimento heurístico e o conhecimento adquirido através do rastro de feromônio artificial. Quando  $q > q^0$  a regra de decisão passa a utilizar uma distribuição de probabilidades e, com isto, passa a explorar mais o espaço de busca do problema. O parâmetro  $q^0$  permite regular se a escolha do próximo estado será concentrada nas melhores soluções ou se vai explorar mais o espaço de busca.

A regra, representada pela Equação 5 define a política de decisão probabilística de movimento para os próximos estados, que é uma função local e leva em consideração duas informações básicas. A primeira delas, que são os valores feromônio  $FE$ , é o modo como os agentes alteram seu espaço de busca para beneficiar a descoberta dos menores caminhos. Esta informação é o feromônio artificial que está associado à técnica do reforço da aprendizagem. A segunda informação usada na escolha para o próximo movimento é a informação referente ao problema específico que está sendo otimizado, ou seja, a heurística  $HE$ .

Inicialmente, como ainda não há uma grande interação entre os agentes, os valores heurísticos  $HE$  predominam sobre os valores feromônio  $FE$ . Neste ponto, fazer uso de uma heurística representativa do problema em questão é extremamente importante, pois o passo inicial do algoritmo é dado a partir desta informação e não de modo aleatório como acontece em vários métodos de otimização.

Os valores  $FE$  são modificados através da cooperação entre os agentes para favorecer a descoberta de boas soluções. As duas regras a seguir são responsáveis pela atualização destes valores.

A regra de atualização local é dada pela Equação 6:

$$FE(r, s) = (1 - \rho) * FE(r, s) + \rho * FEzero \quad (6)$$

A regra de atualização global é dada pela Equação 7:

$$FE(r, s) = (1 - \alpha) * FE(r, s) + \alpha * W/melhorresultado \quad (7)$$

onde  $\rho$  é o parâmetro de evaporação de feromônio,  $FEzero$  é a quantidade inicial de feromônio,  $W$  é um parâmetro, definido pelo usuário, que

juntamente com o parâmetro  $\alpha$  expressam a velocidade de aprendizado do algoritmo, e o melhor resultado é a melhor aptidão encontrada na iteração anterior, e é inicializado com a lógica do vizinho mais próximo.

A regra de atualização local é utilizada após todos os agentes terem aplicado a regra de transição de estado e escolhido a próxima cidade a ser visitada. Desta forma, a atualização é aplicada enquanto a solução está sendo construída. O objetivo da regra de atualização local é estimular a busca por novas regiões do espaço de busca. Ela atua no sentido de diminuir o valor de  $FE(r, s)$  quando o agente que estava na cidade  $r$  e escolheu a cidade  $s$  como a próxima da lista, ou seja, os valores de  $FE$  relativos aos arcos do caminho já escolhidos pelos agentes são penalizados para evitar uma convergência prematura.

A quantidade de feromônio dos arcos é diminuída lentamente para permitir que as formigas artificiais ampliem seus espaços de busca. Este processo é denominado evaporação do feromônio artificial.

O segundo tipo de atualização é a regra de atualização global. Ela é aplicada após todos os agentes terem construído um caminho completo e este caminho ter sido avaliado por uma função objetivo. Esta regra só é aplicada sobre o melhor resultado e é para “premiar” a melhor formiga, colocando uma quantidade maior de feromônio no “caminho” percorrido por ela.

### 3.2.3 Colônia de abelhas artificiais – CAA

O algoritmo de otimização por Colônia de Abelhas Artificiais (CAA), do inglês, *Artificial Bee Colony* (ABC), é um dos mais recentes membros da Inteligência de Enxames. Proposto por [Karaboga \(2005\)](#) para otimização de problemas em domínio contínuo, o CAA tenta simular o comportamento inteligente de uma colônia de abelhas reais na busca por fontes de alimento.

Em virtude da simplicidade de implementação e do bom desempenho na otimização de funções numéricas multimodais e de múltiplas variáveis, o CAA tornou-se uma opção viável e eficiente para a solução de problemas complexos, como pode ser observado em ([Oliveira et al., 2009](#)).

Em sua versão básica, o CAA realiza um procedimento de busca estocástica combinada com busca por vizinhança, no qual indivíduos da população, chamados de posições de fontes de alimento, são modificados durante as iterações através de operadores genéticos, tais como cruzamento e mutação. O principal objetivo das abelhas artificiais é descobrir as regiões de fontes de alimento com grandes quantidades de néctar, e por fim, aquela com a maior quantidade de néctar entre todas.

No algoritmo CAA existem três tipos de abelhas: operárias, observadoras e exploradoras e a posição de uma fonte de alimento corresponde a uma possível solução para o problema de otimização. As abelhas operárias escolhem as fontes de alimento memorizando as regiões onde existem as melhores fontes e estão associadas com uma fonte de alimento particular,

a qual estão explorando. Elas compartilham informações sobre esta fonte particular com suas companheiras de colméia (observadoras) com certa probabilidade.

As abelhas observadoras escolhem as fontes de alimento dependendo da experiência de suas companheiras de colméia (operárias), ajustando deste modo, suas respectivas posições no espaço de busca. Quanto maior a quantidade de néctar de uma fonte de alimento apresentada às abelhas observadoras por uma abelha operária, maior a probabilidade de que uma abelha observadora escolha esta fonte para explorar.

As abelhas exploradoras são responsáveis pela descoberta de novas fontes, diversificando as regiões de busca e evitando desta forma, a convergência prematura do algoritmo. Elas realizam o procedimento de busca totalmente aleatório, sem a utilização da experiência de suas companheiras.

As abelhas utilizam sua capacidade de memorização de forma eficiente e as abelhas artificiais tentam simular este comportamento. Por exemplo, se a quantidade de néctar de uma nova fonte for maior que a quantidade de néctar de uma fonte anterior guardada na memória da abelha, ela então memoriza a posição da nova fonte e esquece a posição da fonte anterior.

Para a utilização do algoritmo CAA, primeiramente o problema de otimização considerado é convertido no problema de encontrar o melhor vetor de parâmetros que minimize ou maximize uma função objetivo. A partir daí, as abelhas artificiais aleatoriamente “descobrem” uma população inicial de vetores solução e iterativamente buscam melhorar estas soluções empregando estratégias que implicam em mover as abelhas na direção das melhores soluções, enquanto abandonam as piores soluções.

Com exceção dos parâmetros comuns, tais como: tamanho da população, número máximo de ciclos, o algoritmo CAA emprega um único parâmetro de controle, chamado limite para abandono de uma fonte. Se, após um número determinado de iterações, uma solução não for melhorada, esta solução é então abandonada. Normalmente define-se o parâmetro limite como sendo o produto entre a dimensão do problema e o tamanho da colônia.

As etapas básicas do Algoritmo CAA são:

No primeiro passo, o algoritmo gera aleatoriamente uma população inicial  $x_i$  de  $SN$  soluções (posições de fontes de alimento), onde  $SN$  corresponde ao número de abelhas operárias. Cada solução  $x_i$  ( $i = 1, 2, \dots, SN$ ) é um vetor  $D$ -dimensional, onde  $D$  corresponde à dimensão do problema.

Após a inicialização, a população de posições (soluções) é submetida a repetidos ciclos,  $C = 1, 2, \dots, MNC$  (Máximo Número de Ciclos) do processo de busca das operárias, observadoras e exploradoras. Uma abelha operária produz uma modificação na posição (solução) em sua memória e testa a quantidade de néctar (valor da função objetivo) da nova fonte (nova solução). Caso a quantidade de néctar da nova fonte seja maior que

o da fonte anterior, a abelha memoriza a posição da nova fonte e esquece a posição anterior. Caso contrário, a abelha mantém a posição da fonte anterior em sua memória.

Após todas as abelhas operárias completarem seu processo de busca, elas compartilham as informações sobre o néctar das fontes com as abelhas observadoras. As observadoras avaliam as informações sobre o néctar trazidas pelas operárias e então escolhem uma fonte de alimento com uma probabilidade relacionada com a quantidade de néctar de cada fonte apresentada. Do mesmo modo que as operárias, as abelhas observadoras também produzem uma modificação na posição da fonte escolhida e verificam a quantidade de néctar da nova fonte. Se esta quantidade for maior que a da fonte anterior, ela substitui esta nova posição em sua memória e esquece a anterior. Caso contrário, ela mantém a posição da fonte anterior.

Quando a quantidade de néctar de uma fonte é abandonada pelas abelhas, uma nova fonte é aleatoriamente determinada por uma abelha exploradora a fim de substituir a fonte que foi abandonada. Estes passos são repetidos até um pré-determinado número de ciclos, ou até que algum critério de parada seja satisfeito. Os principais passos do algoritmo são mostrados a seguir:

No primeiro passo do algoritmo,  $x_i$  ( $i = 1, 2, \dots, SN$ ) soluções são aleatoriamente produzidas no intervalo de parâmetros onde  $SN$  corresponde ao número de fontes de alimento.

No segundo passo do algoritmo, para cada abelha operária, cujo número total corresponde à metade do número de fontes de alimento, uma nova fonte é produzida através da Equação 8:

$$v_{i,j} = x_{i,j} + \varphi_{i,j} (x_{i,j} - x_{k,j}) \quad (8)$$

onde  $\varphi_{i,j}$  é um número real aleatório no intervalo  $[-1,1]$ ,  $k$  é o índice da solução escolhida aleatoriamente da colônia ( $k = \text{int}(\text{rand} * SN) + 1$ ),  $j = 1, \dots, D$  e  $D$  é a dimensão do problema. Depois de produzida, a solução  $v_i$  é então comparada com a solução  $x_i$  e a abelha operária escolhe a melhor fonte para explorar.

No terceiro passo do algoritmo, posições de fontes entre as soluções  $x_i$  são escolhidas com probabilidade de acordo com Equação 9 a fim de serem criadas cópias das melhores fontes. A partir das melhores fontes, novas fontes são produzidas nas vizinhanças destas fontes através da Equação 8, formando as observadoras. As soluções representadas pelas observadoras são comparadas com as soluções  $x_i$ . Do mesmo modo como realizado para as operárias, a melhor fonte é explorada.

$$p_i = \frac{\text{funcao objetivo}_i}{\sum_{j=1}^{SN} \text{funcao objetivo}_j} \quad (9)$$

onde *funcaoobjetivo* corresponde ao valor da função objetivo da solução  $x_i$ .

No quarto passo do algoritmo, fontes são examinadas para determinação das fontes abandonadas. Se o número de ciclos em que uma fonte não pode ser melhorada é maior que um limite pré-determinado, a fonte é considerada abandonada. A operária associada com a fonte abandonada torna-se uma exploradora, realizando uma busca aleatória no domínio do problema através da Equação 10. Os índices *min* e *max* correspondem aos limites inferiores e superiores do espaço de busca, respectivamente.

$$x_{i,j} = x_{i,j}^{min} + (x_{i,j}^{max} - x_{i,j}^{min}) * rand \quad (10)$$

### 3.3 Otimização quântica por colônia de formigas – QACO

O uso de alguns conceitos da computação quântica associados à estrutura dos algoritmos evolucionários deram origem aos algoritmos evolucionários com inspiração quântica (Han & Kim, 2002; Nicolau et al., 2011; da Silva et al., 2011). Eles utilizam uma representação probabilística conhecida como bit quântico, ou *qubit* que é sua unidade fundamental de informação, como mostrado na expressão a seguir:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

onde  $\alpha$  e  $\beta$  são números complexos, sendo que  $|\alpha|^2$  é a probabilidade de que o *qubit* seja encontrado no estado “0” e  $|\beta|^2$  é a probabilidade de que o *qubit* seja encontrado no estado “1”, obedecendo à restrição  $|\alpha|^2 + |\beta|^2 = 1$ .

Cada indivíduo é formado a partir da observação do vetor *qubit*. Este vetor é representado por uma cadeia de *qubits* (Equação 11) sendo, por isto, capaz de representar  $2^m$  estados.

$$qubit = \left[ \begin{array}{c|c|c|c} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{array} \right] \quad (11)$$

Por exemplo, um *qubit* onde  $m = 3$  é capaz de representar os  $2^3$  estados quânticos, representados como  $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle$  e  $|111\rangle$ .

A capacidade de gerar várias possíveis soluções num espaço de busca probabilístico a partir de um único vetor *qubit* corresponde à superposição linear de estados, outro conceito da computação quântica.

Um exemplo deste tipo de ferramenta é o *Quantum Ant Colony Optimization* (QACO) (Wang et al., 2008) que associa conceitos da computação quântica com a metáfora biológica do reforço de feromônio utilizada no *Ant Colony Optimization* (ACO) (Dorigo & Gambardella, 1996). O feromônio

é representado por um *string* de bits quânticos, como pode ser observado na Equação 12:

$$\tau_j = \left[ \begin{array}{c|c|c|c} \tau_{1,\alpha} & \tau_{2,\alpha} & \dots & \tau_{m,\alpha} \\ \tau_{1,\beta} & \tau_{2,\beta} & \dots & \tau_{m,\beta} \end{array} \right] \quad (12)$$

onde  $\tau_{1,\alpha}$  e  $\tau_{1,\beta}$  são números complexos sendo que  $|\tau_{1,\alpha}|^2$  é a probabilidade de que o *qubit* seja encontrado no estado “0” e  $|\tau_{1,\beta}|^2$  é a probabilidade de que o *qubit* seja encontrado no estado “1”, obedecendo a restrição  $|\tau_{1,\alpha}|^2 + |\tau_{1,\beta}|^2 = 1$ .

Para fazer com que o QACO seja capaz de resolver problemas de otimização é necessário converter a notação de *qubits* para a forma binária convencional. Isto é feito através da observação do feromônio  $\tau_j$ . Primeiramente, um número randômico  $p$  é gerado e então comparado com o parâmetro de probabilidade de exploração  $p_e$ . Se  $p$  for maior ou igual a  $p_e$ , o caminho do  $j$ -ésimo bit da formiga  $i$  é determinado pela Equação 13:

$$\text{Caminho}_{i,j} = \begin{cases} 0, & \text{se } \tau_{j,\beta} \leq \tau_{j,\alpha} \\ 1, & \text{se } \tau_{j,\beta} > \tau_{j,\alpha} \end{cases} \quad (13)$$

onde  $\text{Caminho}_{i,j}$  representa o estado do  $j$ -ésimo bit da  $i$ -ésima formiga. Se  $p$  for menor que  $p_e$ , o estado do  $j$ -ésimo bit da formiga  $i$  é determinado pela seguinte função limite, Equação 14:

$$\eta_c(x) = \begin{cases} 0, & \text{se } c < \eta_0 \\ 1 & \text{se } c > \eta_0 \end{cases} \quad (14)$$

onde,  $c$  é um número randômico e  $\eta_0$  é uma constante. A partir da observação do feromônio *qubit*, uma solução válida é gerada e mostrada na forma binária por uma formiga caminhando do formigueiro para uma fonte de alimento, como se pode observar na Figura 1.

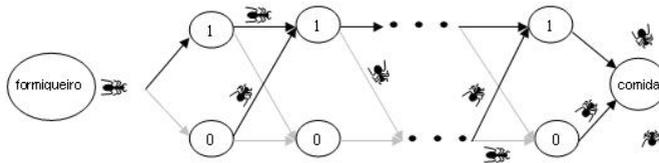


Figura 1. Caminho binário percorrido pelas formigas quânticas.

Uma vez que as soluções binárias convencionais são geradas, cada indivíduo é avaliado de acordo com seu nível de aptidão e os *qubits* são atualizados pelo portão de rotação quântico  $R(\theta_{id})$ , dado pela Equação 15.

$$\begin{bmatrix} \tau'_{i\alpha} \\ \tau'_{i\beta} \end{bmatrix} = R(\theta_{id}) \begin{bmatrix} \tau_{i\alpha} \\ \tau_{i\beta} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{id}) & -\text{sen}(\theta_{id}) \\ \text{sen}(\theta_{id}) & \cos(\theta_{id}) \end{bmatrix} \cdot \begin{bmatrix} \tau_{i\alpha} \\ \tau_{i\beta} \end{bmatrix} \quad (15)$$

onde  $\theta_{id}$  é o ângulo de rotação, responsável por fazer com que as componentes do feromônio *qubit* convirjam para o estado mais apto (0 ou 1). O ângulo de rotação não possui base teórica, sendo definido como mostrado na Equação 16:

$$S(\tau_{j,\alpha}, \tau_{j,\beta}) \cdot \Delta\theta_{id} \quad (16)$$

onde  $S(\tau_{j,\alpha}, \tau_{j,\beta})$  é o sinal de  $\theta_{id}$  que determina o sentido de rotação e  $\Delta\theta_{id}$  é o módulo do ângulo aplicado ao vetor *qubit*. Estes valores são determinados, no QACO, pela Tabela 1 (Wang et al., 2008), na qual  $x_i$  é o  $i$ -ésimo bit da formiga  $x$ ,  $b_i$  é o  $i$ -ésimo bit da melhor formiga,  $f(x)$  é a aptidão da formiga  $x$  e  $f(b)$  é a aptidão da melhor formiga.

Tabela 1. Ângulo de rotação do portão quântico.

$x_i$	$b_i$	$f(x) > f(b)$	$\Delta\theta_{id}$	$S(\tau_{i,\alpha}, \tau_{i,\beta})$			
				$\tau_{i,\alpha}, \tau_{i,\beta} > 0$	$\tau_{i,\alpha}, \tau_{i,\beta} < 0$	$\tau_{i,\alpha} = 0$	$\tau_{i,\beta} = 0$
0	0	Falso	$0,01\pi$	-1	+1	$\pm 1$	$\pm 1$
0	0	Verdadeiro	$0,01\pi$	-1	+1	$\pm 1$	$\pm 1$
0	1	Falso	$0,025\pi$	-1	+1	$\pm 1$	$\pm 1$
0	1	Verdadeiro	$0,025\pi$	+1	-1	$\pm 1$	$\pm 1$
1	0	Falso	$0,025\pi$	+1	-1	$\pm 1$	$\pm 1$
1	0	Verdadeiro	$0,025\pi$	-1	+1	$\pm 1$	$\pm 1$
1	1	Falso	$0,01\pi$	+1	-1	$\pm 1$	$\pm 1$
1	1	Verdadeiro	$0,01\pi$	+1	-1	$\pm 1$	$\pm 1$

#### 4. Metodologia Utilizada no Problema da Recarga

Para demonstrar o desempenho das metaheurísticas mencionadas na Seção 3 para a solução do PRRN, o ciclo 7 de operação da Usina Nuclear Angra 1 é utilizado como referência para este capítulo. Nas simulações realizadas, o código de física de reatores RECNOd (Chapot et al., 1999) foi utilizado.

A Figura 2 mostra uma vista superior do núcleo do reator da Usina Nuclear Angra 1, com seus 121 elementos e eixos de simetria. Este modelo fornece um número de aproximadamente  $10^{200}$  possíveis soluções.

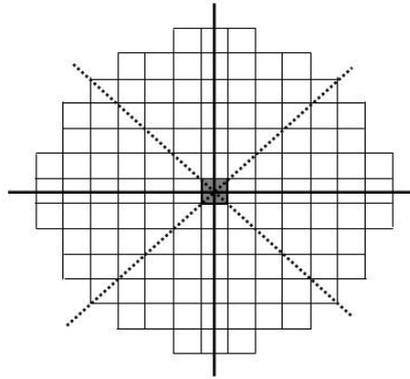


Figura 2. Núcleo de Angra 1.

Devido ao fato da distribuição de potência no núcleo do reator apresentar simetria, esta característica pode ser utilizada com objetivo de reduzir a complexidade deste problema de otimização. Existem dois eixos de simetria principais (linhas contínuas) dividindo o núcleo do reator em quatro regiões (Figura 2). Este modelo representa o que é conhecido como simetria de  $1/4$  de núcleo ou simetria de quarteto. Estes dois eixos principais juntamente com os outros dois eixos secundários (linhas pontilhadas), dividem o núcleo em oito regiões, formando o que é conhecido por simetria de  $1/8$  de núcleo ou simetria de octeto. A Figura 3 mostra a simetria de octeto em detalhes (vista superior). O modelo de octeto é considerado neste estudo e apresenta 21 elementos combustíveis: 10 em ambos os eixos de simetria, 10 entre os eixos de simetria e um elemento central, o qual não participa da permutação.

A simetria de octeto reduz a complexidade do problema uma vez que o número de elementos combustíveis a serem combinados diminui de 121 (todo o núcleo) para 20 (visto que o elemento central é fixo). De qualquer modo, ainda assim este problema continua sendo extremamente complexo, especialmente sob o ponto de vista do tempo gasto na avaliação da função objetivo, considerando-se o grande número de possíveis soluções. Na Figura 3, os elementos combustíveis que ocupam posições de 1 a 10 só podem ser substituídos por outros elementos destas mesmas posições, o mesmo ocorrendo para os elementos das posições de 11 a 20. O objetivo por trás da solução do problema é encontrar o melhor padrão de carregamento para os 20 elementos combustíveis de modo a maximizar a concentração de boro ao final do ciclo. A função objetivo utilizada neste estudo (Equação 17)



ECs de octetos, para evitar a queima assimétrica no núcleo. ECs classificados como de quarteto, são ECs que ocupam as posições de 1 a 10, pois só existem 4 simétricos, e ECs classificados como de octeto, são ECs que ocupam as posições de 11 a 20, como mostrado na Figura 3. Para resolver os problemas da repetição de ECs e o da proibição da troca entre os elementos de quarteto e octeto foi utilizado o modelo de *Random Keys* (Bean, 1994), que impede a formação de um mapa de núcleo com ECs repetidos e com trocas não permitidas. A seguir é mostrado um exemplo do modelo *Random Keys*.

Dado um vetor de número reais com 20 posições, representado na Figura 4, gerado por um dos 4 algoritmos, dividido em dois vetores menores de 10 posições, onde os primeiros 10 representam os ECs de quarteto e os de 11 a 20 representam os ECs de octeto.

1	2	3	4	5	6	7	8	9	10
0,8	0,4	0,2	0,1	0,3	0,8	0,8	0,4	0,1	0,9
Vetor de quartetos									
11	12	13	14	15	16	17	18	19	20
0,7	0,6	0,5	0,6	0,5	0,1	0,7	0,9	0,2	0,9
Vetor de octetos									

Figura 4. Vetor de número reais com 20 posições.

O modelo *Random Keys* consiste em ordenar o vetor de forma crescente, por exemplo, e criar um novo vetor com os índices deste ordenamento. Com esta técnica os dois vetores da Figura 4 são transformados nos vetores representados na Figura 5.

4	9	3	5	2	8	1	6	7	10
Vetor de quartetos									
16	19	13	15	12	14	11	17	18	20
Vetor de octetos									

Figura 5. Reordenação dos vetores da Figura 4, utilizando o modelo *Random Keys*.

A Figura 6 é uma representação do modelo de Núcleo da Usina Angra 1, com simetria de 1/8, com os 20 EC mapeados, onde as posições dos EC de quarteto (2) e octeto (14) aparecem em destaque.

Desta forma, com o uso do modelo *Random Keys* não se tem ECs repetidos e nem ECs de quarteto trocados com ECs de Octeto, onde no primeiro vetor sempre se tem números, sem repetições, entre 1 e 10, e no segundo vetor números, sem repetições, entre 11 e 20. Após este procedimento os

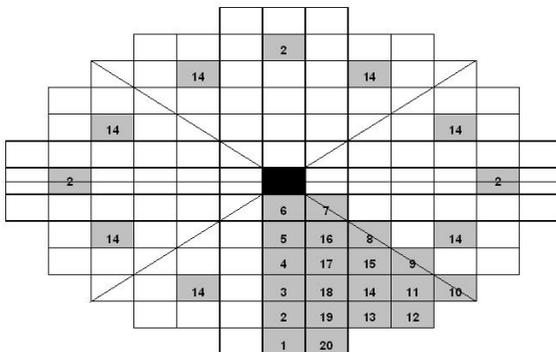


Figura 6. Representação do modelo de Núcleo da Usina Nuclear Angra 1.

vetores são agrupados novamente, criando um novo vetor de 20 posições e enviado para o código de Física de Reatores RECNOd de modo a gerar os valores de  $C_B$  e  $P_{rm}$  que serão utilizados no cálculo da função objetivo.

Quando se aplica o SCF ao problema da recarga, o agente  $k$  é uma formiga artificial que se move de posição em posição no núcleo, preenchendo-as com os ECs, até retornar à posição inicial, construindo um mapa de preenchimento de núcleo. A este agente  $k$  está associada uma lista  $J_k(r)$  (onde  $r$  é a posição atual do agente), que diz quais posições ainda não foram visitadas quando o agente  $k$  está localizado na posição  $r$ , e quais os ECs que ainda não foram utilizados.

O agente  $k$  seleciona uma posição para deslocar-se através de uma Regra de Transição de Estado descrita pela Equação 18 a seguir:

$$s = \begin{cases} \max \{ [FE(r, s, p)]^\delta \times [HE(r, s, p)]^\beta \} & \text{se } q \leq q_0 \\ \text{Roleta} & \text{se } q > q_0 \end{cases} \quad (18)$$

onde a roleta é dada pela distribuição de probabilidades pseudo-aleatória-proporcional, mostrada na Equação 19.

$$\text{Roleta} = \begin{cases} \frac{[FE(r, s, p)]^\delta \times [HE(r, s, p)]^\beta}{\sum_{z \in J_k(r)} [FE(r, z, p)]^\delta \times [HE(r, z, p)]^\beta} & \text{se } s \in J_k(r) \\ 0 & \text{se } s \notin J_k(r) \end{cases} \quad (19)$$

onde a variável  $p$  inserida nas Equações 18 e 19 representa a posição do núcleo em que o EC vai ser inserido, como mostrado nas Figuras 2 e 3.

As Equações 4, 5, 18 e 19 tornam-se semelhantes, exceto pela variável  $p$ , responsável pela transformação das matrizes  $HE$  e  $FE$  em matrizes cúbicas. No caso específico do PRRN, a matriz  $FE$  será preenchida igualmente.

A variável  $p$  foi inserida devido ao fato de que, a contribuição para a soma total do percurso, associada a se colocar a cidade  $r$  ao lado da cidade  $s$  (no caso do PCV), independe da posição que este par  $(r, s)$  aparece na lista de cidades que descreve o percurso do agente  $k$ . No caso da recarga, se colocar o EC  $r$  ao lado do EC  $s$  na posição  $p$  do núcleo, contribui de uma forma diferente se colocarmos o EC  $r$  ao lado do EC  $s$  na posição  $p + 1$  do núcleo.

Em outras palavras, na otimização do PRRN a solução final dada pelo fato do EC  $r$  ser precedido pelo EC  $s$  depende das posições destes ECs no núcleo do reator, isto é, a contribuição que dois ECs seguidos darão para a configuração final se eles estiverem ocupando posições mais internas no núcleo do reator é diferente da contribuição dada pelos mesmos dois ECs quando ocupam posições periféricas do núcleo do reator. Assim, o custo da distância associada entre as posições  $r$  e  $s$  é dependente da posição que estas distâncias ocupam na configuração. Desta forma, o feromônio associado a uma distância  $(r, s)$  nestas condições não expressa corretamente a informação de qualidade que este deveria expressar.

Deste modo, para incorporar esta característica do problema na modelagem, tornou-se necessário incluir uma nova variável nas equações que regem o SCF. Esta nova variável representa a posição do EC  $s$  no núcleo do reator e, assim, o percurso  $(r, s)$  transforma-se no percurso  $(r, s, p)$ , e então, o PRRN passa a ter uma dimensão a mais que o PCV.

Como as matrizes  $FE$  e  $HE$  tiveram a variável  $p$  inserida, as Equações 6 e 7 são transformadas nas Equações 20 e 21, que terão suas descrições idênticas às Equações 6 e 7, com exceção da variável  $p$  (posição), como mostrado a seguir.

A regra de atualização local é dada pela Equação 20:

$$FE(r, s, p) = (1 - \rho) * FE(r, s, p) + \rho * FEzero \quad (20)$$

A regra de atualização global é dada pela Equação 21:

$$FE(r, s, p) = (1 - \alpha) * FE(r, s, p) + \alpha * W/melhorresultado \quad (21)$$

## 5. Resultados

Ao longo de 10 anos de pesquisa em dissertações de mestrado e teses de doutorado no LMP/COPPE, foram estudados vários algoritmos evolucionários com intenção de resolver o problema da recarga de uma forma eficiente.

Pode-se observar através dos resultados, a melhora dos mesmos devido à evolução natural da pesquisa, até atingir níveis acima do esperado.

Todos os testes foram executados para o mesmo ciclo de operação, o ciclo 7 da Usina Nuclear Angra 1, com o mesmo código de física de reatores (RECNOd) e com a mesma função objetivo (Equação 17), com todos os parâmetros dentro do limite de segurança.

Até os dias de hoje, a grande maioria das usinas nucleares têm suas configurações de núcleo feitas por especialistas, que conseguem atingir resultados satisfatórios para a usina. Porém, com o uso de algoritmos evolucionários, estes resultados podem ser melhorados, como mostrado na Tabela 2.

Tabela 2. Resultados dos Algoritmos testados.

<b>Autores</b>	<b>Concentração de Boro</b>	<b>Método</b>	<b>Avaliações</b>
Especialista	955	Manual	—
Chapot (2000)	1026	AG	4000
Meneses et al. (2009)	1394	PSO	4000
de Lima et al. (2008)	1424	SCF	329000
da Silva et al. (2011)	1415	QACO	99240
Oliveira & Schirru (2011)*	1542	CAA	6807

\* Foi permitido misturar EC de quartetos com EC de octetos.

Os valores da concentração de boro nos fornecem uma estimativa da duração do ciclo, em DEPP, onde aproximadamente 4 ppm de boro, equivale a 1 DEPP. Como se pode observar na Tabela 2, ao longo dos anos, a eficiências dos algoritmos foi aumentando e chegando a valores de concentração de boro muito altos. Tem-se ganho de aproximadamente 400 ppm de boro, que equivaleriam em torno de 100 DEPP a mais para o ciclo 7 real da Usina Nuclear Angra 1.

Um DEPP equivale a 1 dia de operação de uma usina nuclear, e cada dia a mais equivale a cerca de 1 milhão de dólares a mais para a operadora da usina e um dia a menos que a usina fica desligada para recarga de combustível.

## 6. Conclusões

Neste capítulo foi mostrada a viabilidade do uso de metaheurísticas de otimização, tais como algoritmos genéticos, enxame de partículas, algoritmo de abelhas, colônia de formigas, e a versão quântica do algoritmo de colônia de formigas, na otimização de um problema de grande relevância econômica da Engenharia Nuclear: a otimização da recarga do combustível da

usina Angra 1. Os resultados apresentados na Seção 5 comprovam a eficácia e viabilidade do uso de tais métodos na otimização do caso particular do ciclo 7 de operação da Usina Nuclear Angra 1, principalmente quando comparado com aos resultados encontrados pelos especialistas.

Além disto, é importante destacar a aplicabilidade de um algoritmo que faz uso de uma das mais recentes tendências na tentativa de aprimorar a Computação Evolucionária, através do uso de conceitos oriundos da Computação Quântica como é o caso do *Quantum Ant Colony Optimization*.

## Referências

- Bean, J., Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2):154–160, 1994.
- Chapot, J., *Otimização Automática de Recargas de Reatores a Água Pressurizada Utilizando Algoritmos Genéticos*. Tese de doutorado, Programa de Engenharia Nuclear/COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2000.
- Chapot, J.; Silva, F. & Schirru, R., A new approach to the use of genetic algorithms to solve pressurized water reactor's fuel management optimization problem. *Annals of Nuclear Energy*, 26(7):641–665, 1999.
- da Silva, M.; Schirru, R. & de Lima, A., QACO\_alpha applied to the nuclear reactor core fuel reload optimization. *Progress in Nuclear Energy*, 53(1):80–85, 2011.
- de Lima, A.; Schirru, R.; Silva, F. & Medeiros, J., A nuclear reactor core fuel reload optimization using artificial ant colony connective networks. *Annals of Nuclear Energy*, 35(9):1606–1612, 2008.
- Dorigo, M. & Gambardella, L., Solving symmetric and asymmetric TSPs by ant colonies. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. Piscataway, USA: IEEE, p. 622–627, 1996.
- Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, USA: Addison Wesley, 1989.
- Han, K. & Kim, J., Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 6(6):580–593, 2002.
- Holland, J., *Adaptation In Natural and Artificial Systems*. Ann Arbor, USA: University of Michigan Press, 1975.
- Karaboga, D., An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report TR06, Computer Engineering Department, Erciyes University, Kayseri, Turkey, 2005.
- Kennedy, J. & Eberhart, R., Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*. Piscataway, USA: IEEE, v. 4, p. 1942–1948, 1995.

- Kropaczek, D. & Turinsky, P., In-core nuclear fuel management optimization for pressurized reactors utilizing simulated annealing. *Nuclear Technology*, 95:9–32, 1991.
- Lin, C.; Yang, J.I.; Lin, K.J. & Wang, Z.D., Pressurized water reactor loading pattern design using the simple tabu search. *Nuclear Science and Engineering*, 129(1):61–71, 1998.
- Meneses, A.; Machado, M. & Schirru, R., Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor. *Progress in Nuclear Energy*, 51(2):319–326, 2009.
- Nicolau, A.; Schirru, R. & Meneses, A., Quantum evolutionary algorithm applied to the transient identification of a nuclear power plant. *Progress in Nuclear Energy*, 53(1):86–91, 2011.
- Oliveira, I. & Schirru, R., Swarm intelligence of artificial bees applied to in-core fuel optimization. *Annals of Nuclear Energy*, 38(5):1039–1045, 2011.
- Oliveira, I.; Schirru, R. & Medeiros, J., On the performance of an artificial bee colony optimization algorithm to the accident diagnosis of a PWR nuclear power plant. In: *Proceedings of International Nuclear Atlantic Conference*. Rio de Janeiro, RJ: ABEN, 2009.
- Schirru, R.; de Lima, A. & Machado, M., Parallel evolutionary methods applied to a PWR core reload pattern optimization. In: Ruan, D.; D'hondt, P.; Fantoni, P.; Cock, M.; Nachtegaele, M. & Kerre, E. (Eds.), *Applied Artificial Intelligence – Proceedings of the 7<sup>th</sup> International FLINS Conference*. Singapore: World Scientific, p. 915–921, 2006.
- Wang, L.; Niu, Q. & Fei, M., A novel quantum ant colony optimization algorithm and its application to fault diagnosis. *Transactions of the Institute of Measurement and Control*, 30(3-4):313–329, 2008.

## Notas Biográficas

**Alan M. M. de Lima** é bolsista CAPES/FAPERJ, possui graduação em Física pela Universidade Federal Fluminense (UFF 1998), mestrado, doutorado e pós-doutorado em Engenharia Nuclear pela Universidade Federal do Rio de Janeiro – UFRJ (2000, 2005 e 2006). Tem experiência na área de Engenharia Nuclear, com ênfase em Núcleo do Reator e recarga de reatores nucleares do tipo PWR, atuando principalmente nos seguintes temas: otimização, algoritmos evolucionários e inteligência artificial.

**Andressa dos Santos Nicolau** é bolsista CNPQ, graduada em Física pela Universidade Estadual do Rio de Janeiro – UERJ (2007) e Mestre em Engenharia Nuclear pela Universidade Federal do Rio de Janeiro – UFRJ (2010). Atualmente é estudante de doutorado em Engenharia Nuclear pela Universidade Federal do Rio de Janeiro – UFRJ (2010). Possui experiência em Ciência da Computação com ênfase em Inteligência Artificial.

**Ioná Maghali Santos de Oliveira** é bolsista CNPQ, possui graduação em Engenharia Elétrica pelo Centro Universitário da Cidade (2002) e mestrado em Engenharia Nuclear na área de Análise de Segurança pela Universidade Federal do Rio de Janeiro (2007). Tem experiência na área de Engenharia Elétrica, com ênfase em Engenharia Eletrônica e doutorado em andamento na área de Engenharia de Fatores Humanos pela Universidade Federal do Rio de Janeiro.

**Jose Antonio Carlos Canedo Medeiros** é graduado em Engenharia Elétrica pela Universidade do Estado do Rio de Janeiro – UERJ (1977), tem mestrado e doutorado em Engenharia Nuclear pela Universidade Federal do Rio de Janeiro – UFRJ (1999 e 2005). Atualmente é professor da Universidade Federal do Rio de Janeiro – UFRJ.

**Márcio Henrique da Silva** é bolsista CNPQ, possui licenciatura em Física pela Universidade Federal do Rio de Janeiro (2007) e mestrado em Engenharia Nuclear pela Universidade Federal do Rio de Janeiro (2010).

**Roberto Schirru** é bolsista de produtividade em pesquisa do CNPQ – nível 1B. Tem bacharelado em Física pela Universidade Federal do Rio de Janeiro – UFRJ (1976), é Mestre e Doutor em Engenharia Nuclear pela Universidade Federal do Rio de Janeiro – UFRJ (1980 e 1991). Tem pós-doutorado em inteligência artificial pelo Instituto Della Molle Di Studi Di Intelligenza Artificiale (1994). É especializado em processamento em tempo real pela Digital Equipments Computer (1977) e inteligência artificial pelo Centro Europeu de Pesquisas Nucleares (1993). Atualmente é professor associado 3 da Universidade Federal do Rio de Janeiro – UFRJ. Possui experiência na área de Ciência da Computação, com ênfase em sistemas de computação, atuando principalmente nos seguintes temas: Representação de Conhecimento, Operação de Usinas Nucleares, Segurança de Usinas Nucleares.



## Resolução de Problemas Inversos em Processos Difusivos e Transferência Radiativa usando o Algoritmo de Evolução Diferencial

Fran Sérgio Lobato, Valder Steffen Jr. e Antônio J. Silva Neto\*

---

**Resumo:** A formulação e a resolução de problemas inversos em processos difusivos e de transferência radiativa têm um papel fundamental no desenvolvimento científico e tecnológico em diversas áreas, com aplicações em engenharia e medicina, tais como tomografia computadorizada, sensoriamento remoto e ciências ambientais. Tradicionalmente, estes problemas são resolvidos usando técnicas de otimização baseadas no uso de derivadas. Neste capítulo, apresenta-se como alternativa para o tratamento destes problemas o algoritmo de Evolução Diferencial, que se fundamenta na realização de operações vetoriais para a atualização de candidatos à solução do problema. Além disso, apresenta-se uma estratégia para a atualização dinâmica dos parâmetros desta estratégia.

**Palavras-chave:** Algoritmo de evolução diferencial, Problemas inversos, Processos difusivos, Transferência radiativa.

**Abstract:** *The formulation and resolution of inverse problems in diffusive processes and radiative transfer have a fundamental role in scientific and technological development in several areas, with applications in engineering and medicine, such as computerized tomography, remote sensing and environmental sciences. Traditionally, these problems are solved by using optimization techniques based on the use of derivatives. In this chapter, it is presented an alternative to the treatment of these problems by using the Differential Evolution algorithm. It is based on vector operations for dynamically updating candidates to the solution of the problem. In addition, a strategy for dynamic updating the parameters of this algorithm is also presented.*

**Keywords:** *Differential evolution algorithm, Inverse problems, Diffusive process, Radiative transfer.*

---

\* Autor para contato: [ajsneto@iprj.uerj.br](mailto:ajsneto@iprj.uerj.br)

## 1. Introdução

Dentre os métodos de otimização heurísticos desenvolvidos nos últimos anos, o algoritmo de Evolução Diferencial (ED), proposto em 1995 por Storn e Price (Storn & Price, 1995), se configura como uma das estratégias mais utilizadas para a resolução de problemas da ciência e engenharia. Seu sucesso se deve a sua concepção conceitual simples, facilidade de implementação, capacidade de estruturação em arquitetura paralela, habilidade de escapar de ótimos locais, e pelos resultados obtidos em aplicações com diferentes graus de complexidade.

Em linhas gerais, a idéia principal por trás deste algoritmo é o esquema proposto para atualização de cada indivíduo, a saber, por meio da realização de operações vetoriais. Basicamente, a diferença ponderada entre dois indivíduos da população é adicionada a um terceiro indivíduo da mesma população. Assim, o indivíduo gerado através deste esquema é avaliado segundo a função objetivo, podendo inclusive substituir indivíduos mal sucedidos nas gerações seguintes. Esta característica faz com que esta técnica seja reconhecida como uma abordagem puramente estrutural, o que a diferencia em relação às outras técnicas evolutivas, já que estas têm fundamentação teórica inspirada na natureza.

Na literatura recente inúmeras aplicações usando o algoritmo de ED podem ser encontradas. Os próprios autores deste capítulo já publicaram um elenco relativamente amplo de trabalhos relatando aplicações deste tipo, dentre as quais pode-se citar a estimação de parâmetros cinéticos em um secador rotatório (Lobato et al., 2008), a estimação de parâmetros de um filtro adaptativo utilizado como ferramenta para controle de qualidade em linhas de montagem de câmbios automotivos (Oliveira-Filho et al., 2008), e a estimação da difusividade térmica de frutas (Mariani et al., 2008). Na área de controle ótimo tem-se como exemplos a determinação do perfil ótimo de alimentação de substrato em fermentadores (Kapadi & Gudi, 2004) e a resolução de problemas de controle ótimo clássicos com restrições de desigualdade (Lobato et al., 2007b). No contexto multi-objetivo destacam-se a otimização estrutural de vigas (Lobato & Steffen Jr., 2007), a determinação do perfil ótimo de alimentação de substrato em um problema de controle ótimo com índice flutuante (Lobato et al., 2007a), a otimização das condições operacionais de um reator industrial utilizado para a produção de estireno (Babu et al., 2005) e o uso de técnicas de meta-modelagem associada ao enfoque multi-objetivo para o tratamento de um problema de interação fluido-estrutura (Lobato, 2008). Outras aplicações podem ser encontradas em Price et al. (2005), em Lobato (2008) e no site: <http://www.icsi.berkeley.edu/~storn/code/html>.

Apesar do número expressivo de aplicações em áreas distintas da literatura especializada, pode-se dizer que o número de contribuições em

problemas inversos, em especial no tratamento de processos difusivos e transferência radiativa, ainda é bastante limitado.

Neste contexto, o presente capítulo tem por objetivo apresentar aplicações do algoritmo de ED na resolução de problemas inversos em processos difusivos e transferência radiativa, com interesse específico no problema de transferência radiativa em meios homogêneos e meios heterogêneos, bem como em outro, acoplando radiação-condução. Além disso, também é apresentada uma estratégia para a atualização dinâmica dos parâmetros do algoritmo, sendo este ponto uma contribuição dos autores no sentido de melhorar o desempenho do algoritmo clássico. Este capítulo é estruturado como se segue. Na Seção 2 são apresentadas as motivações para o desenvolvimento do algoritmo de ED. A concepção do algoritmo, com a respectiva descrição dos operadores é mostrada na Seção 3. Na Seção 4 são apresentadas aplicações em problemas de transferência radiativa em meio participante com propriedades constantes e dependentes da variável óptica espacial, considerando a atualização dinâmica dos parâmetros do algoritmo de ED, em um problema de transferência radiativa composto por duas camadas e em um problema envolvendo a transferência de calor por radiação-condução. Finalmente, na Seção 5 são apresentadas as considerações finais.

## 2. Motivação para o Desenvolvimento do Algoritmo de ED

A principal motivação para o desenvolvimento do algoritmo de ED foi a lenta taxa de convergência e a dificuldade na determinação dos parâmetros do polinômio de Chebychev exibida pelo algoritmo híbrido denominado Recozimento Genético – *Genetic Annealing Algorithm* (Price, 1994).

Foi durante a resolução deste problema que Storn decidiu modificar este algoritmo híbrido de forma a trabalhar com codificação de ponto flutuante e com operações aritméticas. Neste sentido, ele acabou por desenvolver o operador de mutação diferencial, o que fundamenta o algoritmo de ED. Com o sucesso obtido com a aplicação deste operador, Storn sugeriu outras configurações para a determinação de candidatos em potencial para a solução do problema de otimização (Storn & Price, 1995).

O primeiro contato da comunidade científica com o algoritmo de ED foi no ICEO – *International Contest on Evolutionary Optimization*. Neste evento foi demonstrado o desempenho da técnica em uma série de funções matemáticas com diferentes graus de dificuldade e complexidade. Esta primeira aparição foi consolidada posteriormente (Storn & Price, 1997).

## 3. Concepção Conceitual

O algoritmo de ED baseia-se na realização de operações vetoriais na qual a diferença ponderada entre dois indivíduos distintos, adicionada a um

terceiro indivíduo, é o responsável pela geração de candidatos. De maneira geral, o algoritmo de ED apresenta as seguintes operações: inicialização da população, mutação, cruzamento, seleção, além do critério de parada do algoritmo. A seguir são descritas cada uma destas operações.

### 3.1 Inicialização do algoritmo de ED

O processo de inicialização da população no algoritmo de ED, assim como acontece em outras estratégias heurísticas, consiste na geração de indivíduos de forma aleatória. Neste caso, faz-se uso da definição do tamanho da população e do domínio de cada variável de projeto. A partir daí, geram-se números aleatórios que serão aplicados a este intervalo, obtendo-se assim um vetor de indivíduos da população, como mostrado na Equação 1, a seguir:

$$x_{i,j} = x_{i,L} + rand(x_{i,U} - x_{i,L}) \quad (1)$$

onde  $x_{i,L}$  e  $x_{i,U}$  são os limites inferiores e superiores das  $j$ -ésimas variáveis de projeto, respectivamente, e  $rand$  é um gerador de números aleatórios entre 0 e 1.

Deve ser ressaltado que, na literatura, outros mecanismos para a inicialização da população podem ser encontrados, como por exemplo, geração fundamentada no uso de distribuições Gaussianas, em que os indivíduos da população são gerados com média e desvio padrão pré-definidos pelo usuário (Abbass et al., 2001).

### 3.2 Operador de mutação

Uma vez inicializado o processo evolutivo, o algoritmo de ED realiza as operações de mutação e recombinação para a geração de uma nova população com  $NP$  indivíduos. O operador de mutação diferencial adiciona um vetor de referência, escolhido aleatoriamente na população, a um vetor diferença obtido a partir de outros dois vetores também escolhidos aleatoriamente na população. A Equação 2 mostra como esta combinação é realizada para gerar o novo vetor (candidato)  $v_{i,g}$ :

$$v_{i,g} = x_{r0,g} + F(x_{r1,g} - x_{r2,g}) \quad (2)$$

onde o escalar  $F$ , denominado taxa de perturbação, é um número real que controla a magnitude do vetor diferença obtido em cada operação aritmética realizada. Os vetores  $x_{r0,g}$ ,  $x_{r1,g}$  e  $x_{r2,g}$  são geralmente escolhidos aleatoriamente na população, apesar de poderem ser definidos de outras formas (Fan & Lampinen, 2003), como ilustrado na Tabela 1.

A convenção utilizada nesta tabela é DE/X/Y/Z, onde X representa o vetor que será perturbado (BEST ou RAND). Por exemplo, optando-se por RAND, o vetor que será perturbado é escolhido aleatoriamente na população. Y é o número de pares de vetores que são considerados durante

Tabela 1. Representação dos mecanismos de mutação no algoritmo de ED.

Estratégia	Representação	Mecanismo de Mutação
1	$x^{j+1} = x_{best}^j + F(x_{\kappa_1}^j - x_{\kappa_2}^j)$	DE/BEST/1/EXP
2	$x^{j+1} = x_{\kappa_3}^j + F(x_{\kappa_1}^j - x_{\kappa_2}^j)$	DE/RAND/1/EXP
3	$x^{j+1} = x_{old}^j + F(x_{best}^j - x_{old}^j) + F(x_{\kappa_1}^j - x_{\kappa_2}^j)$	DE/RAN-TO-BEST/2/EXP
4	$x^{j+1} = x_{best}^j + F(x_{\kappa_1}^j - x_{\kappa_2}^j) + F(x_{\kappa_3}^j - x_{\kappa_4}^j)$	DE/BEST/2/EXP
5	$x^{j+1} = x_{\kappa_5}^j + F(x_{\kappa_1}^j - x_{\kappa_2}^j) + F(x_{\kappa_3}^j - x_{\kappa_4}^j)$	DE/RAND/2/EXP
6	$x^{j+1} = x_{best}^j + F(x_{\kappa_1}^j - x_{\kappa_2}^j)$	DE/BEST/1/BIN
7	$x^{j+1} = x_{\kappa_3}^j + F(x_{\kappa_1}^j - x_{\kappa_2}^j)$	DE/RAND/1/BIN
8	$x^{j+1} = x_{old}^j + F(x_{best}^j - x_{old}^j) + F(x_{\kappa_1}^j - x_{\kappa_2}^j)$	DE/RAND-TO-BEST/2/BIN
9	$x^{j+1} = x_{best}^j + F(x_{\kappa_1}^j - x_{\kappa_2}^j) + F(x_{\kappa_3}^j - x_{\kappa_4}^j)$	DE/BEST/2/BIN
10	$x^{j+1} = x_{\kappa_5}^j + F(x_{\kappa_1}^j - x_{\kappa_2}^j) + F(x_{\kappa_3}^j - x_{\kappa_4}^j)$	DE/RAND/2/BIN

a perturbação e  $Z$  é o tipo de cruzamento usado para a geração do candidato, BIN no caso binomial e EXP no caso exponencial. Os subscritos  $\kappa_i$  ( $i=1, \dots, 5$ ) são índices escolhidos aleatoriamente na população.  $x_{best}$  é o melhor indivíduo da população na geração anterior e  $x_{old}$  é um indivíduo escolhido aleatoriamente dentro da população na geração anterior.

### 3.3 Operador de cruzamento

Para complementar a operação de mutação, o algoritmo de ED emprega o operador de cruzamento. Nesta operação, o vetor  $v_{i,g}$  gerado anteriormente pode ou não ser aceito na próxima geração de acordo com a seguinte condição:

$$u_{i,g} = \begin{cases} v_{i,g} & \text{se } rand \leq CR \text{ ou } j = j_{rand} \\ x_{i,g} & \text{caso contrário} \end{cases} \quad (3)$$

onde  $CR$  é denominada probabilidade de cruzamento, definida pelo usuário e contida no intervalo  $[0, 1]$ . Este parâmetro controla as informações dos pais que serão transmitidas aos filhos. Para determinar qual a contribuição de um determinado vetor gerado, o cruzamento compara  $CR$  com o gerador de números aleatórios  $rand$ . Se o número randômico gerado é menor ou igual a  $CR$ , o vetor  $v_{i,g}$  (com posição  $j_{rand}$ ) é aceito, caso contrário, o vetor  $x_{i,g}$  é mantido na população atual.

### 3.4 Operador de seleção

Se o vetor  $u_{i,g}$  tem melhor valor de função objetivo ( $f$ ) em relação ao vetor  $x_{i,g}$ , ele o substitui na próxima geração. Caso contrário,  $x_{i,g}$  é mantido na população por mais uma geração, como mostrado a seguir:

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{se } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{caso contrário} \end{cases} \quad (4)$$

Uma vez completado o processo de atualização da população que será considerada na próxima geração, todo o processo descrito acima é repetido até que um determinado critério de parada seja satisfeito.

### 3.5 Critério de parada

O critério de parada objetiva a convergência do algoritmo associada a um baixo esforço computacional, mensurado pelo número de avaliações da função objetivo. Tradicionalmente, o principal critério de parada utilizado em abordagens heurísticas é mesmo o número máximo de gerações, definido pelo usuário. Todavia, outros mecanismos podem ser utilizados para finalizar o processo evolutivo, dentre os quais se pode citar: o tempo de processamento, o número de avaliações da função objetivo, o uso de um valor de referência obtido da literatura para esta finalidade e o monitoramento do próprio usuário (Price et al., 2005).

Com relação à escolha dos parâmetros do algoritmo de ED, Storn & Price (1995) aconselham o uso dos seguintes valores: número de indivíduos da população como sendo igual a um valor entre 5 e 10 vezes o número de variáveis de projeto, taxa de perturbação  $F$  entre 0,2 e 2,0 e probabilidade de cruzamento  $CR$  entre 0,1 e 1,0. Com relação à escolha da estratégia DE/X/Y/Z, Babu et al. (2005) e Babu & Anbarasu (2005) constataram em seus trabalhos que os melhores resultados, em termos de convergência e diversidade para os problemas estudados por estes autores, foram obtidos quando se utiliza a estratégia DE/RAND/1/BIN. É importante ressaltar que outros valores para esses parâmetros podem ser atribuídos de acordo com uma aplicação particular.

Uma vez destacada a fundamentação teórica do algoritmo de ED e as principais características dos seus operadores, cabe ressaltar que no *site* <http://www.icsi.berkeley.edu/~storn/code/html> pode ser feito o *download* do algoritmo de ED implementado em várias linguagens de programação.

Na próxima seção serão apresentadas aplicações de problemas inversos dedicados a processos difusivos e transferência radiativa.

## 4. Aplicações

Para as aplicações que se seguem faz-se necessário enfatizar alguns pontos para a melhor compreensão dos estudos. Como não são conhecidos os pontos experimentais relativos a cada aplicação, são admitidos valores para as variáveis de projeto a serem determinadas a fim de resolver o problema direto (resolução do problema de simulação). Desta forma, conhecendo-se o perfil da intensidade da radiação (e da temperatura no problema de condução) em pontos do domínio e atribuindo-se um ruído definido pelo usuário define-se os pontos “experimentais sintéticos” que serão utilizados neste capítulo. Este ruído acrescido ao ponto calculado procura reproduzir os erros de medição que são inerentes a qualquer procedimento experimental. Deste modo, tal aspecto será considerado neste trabalho através da seguinte relação:

$$\Pi_{\text{exp}} = \Pi_{\text{cal}} + r_i \sigma \quad (5)$$

onde  $\Pi_{cal}$  e  $\Pi_{exp}$  são os valores calculados e experimentais para a grandeza considerada (radiação e/ou condução), respectivamente,  $r_i$  é um número aleatório entre -1 e 1 e  $\sigma$  é relacionado ao desvio padrão associado ao procedimento experimental.

O problema inverso consiste na minimização do funcional  $Q$ , isto é, minimização da diferença entre os valores “experimentais” e os valores calculados pelo sistema de equações que regem o fenômeno em análise. Matematicamente, tem-se:

$$Q = \sum_{i=1}^{N_d} (\Pi_{cal}^i - \Pi_{exp}^i)^2 \quad (6)$$

onde  $N_d$  representa o número total de dados experimentais. Deve ser ressaltado que, para cada aplicação a ser apresentada, um conjunto de variáveis de projeto específicas será analisado.

Para a resolução do problema direto, o termo da integral encontrado do lado direito de cada equação será substituído pela fórmula de Quadratura de Gauss-Legendre (Wylie & Barrett, 1995), transformando o problema integro-diferencial em um sistema de Equações Diferenciais Ordinárias (EDO's) de valor no contorno. Para a discretização da variável associada ao ângulo polar,  $\mu$ , são utilizados os seguintes valores:  $\{-0,9983 -0,9830 -0,9426 -0,8765 -0,7864 -0,6750 -0,5451 -0,4003 -0,2446 -0,0823 0,0823 0,2446 0,4003 0,5451 0,6750 0,7864 0,8765 0,9426 0,9830 0,9983\}$  (Chalhoub et al., 2007). Este sistema de EDO's será resolvido usando o Método da Colocação Normal com 10 pontos (Villadsen & Michelsen, 1978; Wylie & Barrett, 1995).

Todos os estudos de caso apresentados neste capítulo foram executados 10 vezes (por ser o processo aleatório) para a obtenção dos resultados apresentados nas tabelas.

#### 4.1 Problema de transferência radiativa em um meio participante

Este primeiro estudo de caso considera um meio plano-paralelo, espalhador anisotrópico, cujas propriedades não dependem do comprimento de onda da radiação, as superfícies de contorno são refletoras difusas e estão sujeitas a radiação externa isotrópica, como ilustrado na Figura 1, a seguir.

Matematicamente, o fenômeno de transferência radiativa é modelado por uma equação integro-diferencial, conhecida como equação de Boltzmann (Özişik, 1973):

$$\mu \frac{\partial I(\tau, \mu)}{\partial \tau} + I(\tau, \mu) = \frac{\omega(\tau)}{2} \int_{-1}^1 p(\mu, \mu') I(\tau, \mu') d\mu', \quad (7)$$

$$0 < \tau < \tau_o, \quad -1 \leq \mu \leq 1$$

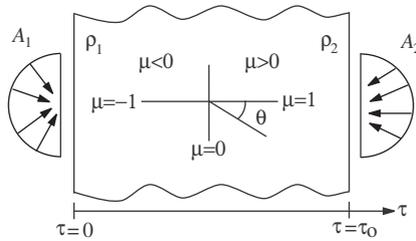


Figura 1. Meio participante uni-dimensional.

$$I(0, \mu) = A_1 + 2\rho_1 \int_0^1 I(0, -\mu') \mu' d\mu', \quad \mu > 0 \quad (8)$$

$$I(\tau_o, \mu) = A_2 + 2\rho_2 \int_0^1 I(\tau_o, \mu') \mu' d\mu', \quad \mu < 0 \quad (9)$$

onde  $I(\tau, \mu)$  é a intensidade de radiação,  $\tau$  é a variável óptica,  $\mu$  é o cosseno entre a intensidade da radiação e o eixo  $\tau$ ,  $\omega$  é o albedo de espalhamento,  $\rho_1$  e  $\rho_2$  são as reflectividades nos contornos em  $\tau=0$  e  $\tau=\tau_o$ , respectivamente,  $A_1$  e  $A_2$  são as intensidades da radiação externa incidente nestes contornos, e  $p(\mu, \mu')$  é a função de fase de espalhamento.

No presente estudo foram admitidos os seguintes valores para as variáveis de projeto para a determinação dos pontos “experimentais sintéticos”:  $\omega$  constante=0,1,  $\tau_o=5$ ,  $A_1=1$ ,  $A_2=0$ ,  $\rho_1=\rho_2=0$ ,  $p(\mu, \mu')=1$  e  $N_d=400$ . Como são considerados 20 pontos na discretização do domínio angular, tem-se então 20 pontos no interior e no contorno onde são feitas as medidas experimentais. O espaço de projeto é assim definido:  $0 < \omega < 1$ ,  $4 < \tau_o < 6$ ,  $0 < A_1 < 1,5$  e  $0 < A_2 < 1$ . Cabe ressaltar que foram utilizados pontos interiores (Detectores Internos – DI) ao domínio de forma a se evitar a não-unicidade da solução.

Para avaliar a qualidade dos resultados obtidos pelo algoritmo de ED, será utilizado o algoritmo de Recozimento Simulado – RS (*Simulated Annealing* – SA) com os seguintes parâmetros:  $T_o=5$  (temperatura inicial),  $r_t=0,75$  (taxa de resfriamento),  $N_s=50$  (número de temperaturas),  $N_t=5$  (número de vezes que o procedimento é repetido antes da temperatura ser reduzida),  $\epsilon=10^{-6}$  (tolerância) e estimativa inicial:  $[\omega \ \tau_o \ A_1 \ A_2]=[0,25 \ 4,5 \ 0,5 \ 0,5]$ . Por outro lado, no algoritmo de ED foram utilizados os parâmetros: população inicial com 10 indivíduos, probabilidade de cruzamento igual a 0,5, taxa de perturbação igual a 0,8, estratégia DE/RAND/1/BIN e 100 gerações.

A Tabela 2 apresenta os resultados obtidos pelos algoritmos evolutivos, onde  $N_{eval}$  é o número médio de avaliações da função objetivo e  $cpitime$  é o tempo médio de execução do algoritmo em minutos, utilizando um microcomputador com processador Pentium 1 GHz.

Tabela 2. Resultados obtidos pelos algoritmos ED e RS.

Exato	Erro nos dados experimentais		$\omega$	$\tau_o$	$A_1$	$A_2$	$Q$ (Eq. 6)
			0,1	5,0	1,0	0,0	-
DE*	0,0	Pior	0,1024	4,9982	0,9988	0,0013	6,3559E-6
		Médio	0,1004	4,9976	0,9992	0,0000	2,6107E-6
		Melhor	0,0998	5,0036	1,0008	0,0000	1,1856E-7
	0,5%	Pior	0,0978	4,9438	0,9844	0,0007	8,0356E-4
		Médio	0,0984	4,9470	0,9847	0,0008	8,0333E-4
		Melhor	0,0983	4,9494	0,9850	0,0010	8,0310E-4
	5,0%	Pior	0,0453	4,9678	0,9683	0,0000	0,0878
		Médio	0,0454	4,9675	0,9682	0,0000	0,0878
		Melhor	0,0455	4,9674	0,9680	0,0000	0,0878
RS**	0,0	Pior	0,0997	5,0097	1,0026	0,0004	8,6468E-7
		Médio	0,0998	4,9981	0,9995	0,0003	7,7231E-7
		Melhor	0,0994	4,9956	0,9988	0,0005	7,1664E-7
	0,5%	Pior	0,0929	4,9487	0,9789	0,0009	9,4786E-3
		Médio	0,0971	4,9256	0,9848	0,0005	8,0999E-3
		Melhor	0,0987	4,9390	0,9841	0,0004	8,0645E-4
	5,0%	Pior	0,0483	4,9578	0,9689	0,0001	0,0892
		Médio	0,0484	4,9575	0,9685	0,0001	0,0890
		Melhor	0,0485	4,9554	0,9680	0,0001	0,0888

\* $N_{eval}=1010$ ,  $cpitime=21,5$  min

\*\* $N_{eval}=8478$ ,  $cpitime=62,2$  min

A Figura 2 apresenta os perfis de intensidade de radiação em  $\tau=0$  e  $\tau=\tau_o$  para dados experimentais sem ruído.

De forma geral, pode ser observado com este estudo que ambos os algoritmos conseguem obter resultados satisfatórios. Entretanto, o algoritmo de ED o faz com menor esforço computacional, a saber, 1010 contra 8478 avaliações da função objetivo requeridas pelo algoritmo de RS.

Mais detalhes sobre esta aplicação, além de outros casos teste, podem ser encontrados em Lobato et al. (2009a, 2010a).

## 4.2 Problema de transferência radiativa com propriedades dependentes da variável óptica espacial

Neste estudo, o algoritmo de ED é aplicado na estimação do albedo de espalhamento dependente da variável óptica espacial, isto é;  $\omega(\tau)$ . Além disso, considera-se que somente o contorno em  $\tau=0$  é sujeita a radiação isotrópica incidente, isto é;  $I(0, \mu)=A_1=1$ , com  $\mu>0$  e  $I(\tau_o, \mu)=A_2=0$ , com  $\mu<0$ , ( $\rho_1=\rho_2=0$ ,  $p(\mu, \mu')=1$  e  $N_d = 400$ ).

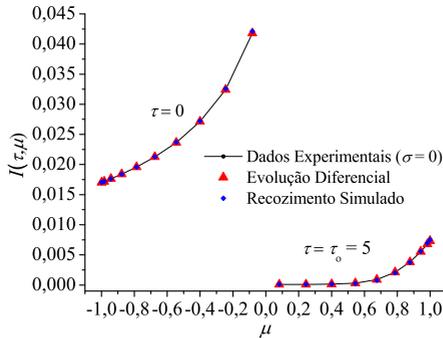


Figura 2. Perfil de intensidade de radiação.

A função que representa o albedo de espalhamento simples considerado nesta análise foi estimada e resolvida por [Stephany et al. \(2010\)](#) usando o algoritmo de Colônia de Formigas (CF) e sua hibridização com o Método de Levenberg-Marquardt (LM):

$$\omega(\tau) = 0.2 + 0.2\tau + 0.6\tau^2 \quad (10)$$

A Tabela 3 apresenta os resultados obtidos usando o algoritmo de Colônia de Formigas (120 formigas e 500 gerações), a hibridização entre este algoritmo (com 10 formigas e 40 gerações) e o Método de Levenberg Marquardt e com o algoritmo de ED (10 indivíduos, taxa de perturbação e probabilidade de cruzamento, ambas iguais a 0,8, estratégia DE/RAND/1/BIN e 100 gerações).

Tabela 3. Resultados obtidos pelos algoritmos de CF, ED e CF-Levenberg Marquardt.

Algoritmo	Nível de Ruído		$N_{eval}$
	0%	5%	
CF	7,22E-06	-	60000
CF-LM	4,39E-10	1,31E-03	400+19
ED	1,63E-09	1,27E-05	1010

Comparando as técnicas heurísticas, pode ser observado que o algoritmo de ED encontra melhores resultados, em termos da função objetivo e do número de avaliações da função objetivo ( $N_{eval}$ ), do que o algoritmo de CF. Com relação ao híbrido CF-LM, o algoritmo de ED consegue obter resultados com a mesma ordem de grandeza.

Mais detalhes sobre este estudo de caso pode ser encontrado em [Lobato et al. \(2010b\)](#).

### 4.3 Problema de transferência radiativa com atualização dinâmica de parâmetros no algoritmo de ED

Este estudo tem por objetivo apresentar uma estratégia para a atualização dinâmica de parâmetros do algoritmo de ED usando o conceito de diversidade da população proposto por [Zahaire \(2003\)](#). A metodologia proposta é aplicada para a estimação simultânea da função de fase de espalhamento anisotrópico, do albedo de espalhamento simples e da espessura óptica, em problemas de transferência de calor por radiação. Segundo [Zahaire \(2002, 2003\)](#), o valor esperado para a variância da população ( $x$ ) obtida após recombinação é dado por:

$$E(Var(x)) = \left( 2F^2 CR + 1 - \frac{2CR}{NP} + \frac{CR^2}{NP} \right) Var(x) \quad (11)$$

onde  $E$  é a esperança matemática,  $Var$  é a variância de  $x$ ,  $F$  é taxa de perturbação,  $CR$  é a probabilidade de cruzamento e  $NP$  é o tamanho da população. Definindo  $\gamma$  como sendo:

$$\gamma \equiv \frac{Var(x(g+1))}{Var(x(g))} \quad (12)$$

onde  $x(g)$  e  $x(g+1)$  representam as populações nas gerações  $g$  e  $g+1$ .

Na equação acima, se  $\gamma < 1$  compensa-se o aumento da variância, desta forma acelerando o processo de convergência, porém com risco de obter convergência prematura. Por outro lado, se  $\gamma > 1$  compensa-se a diminuição da variância, podendo desta forma evitar convergência prematura. Neste sentido, pode-se, através do monitoramento da população e controle dos parâmetros do algoritmo de ED, compensar o efeito da aplicação dos operadores de recombinação e seleção do algoritmo. Segundo esta idéia, [Zahaire \(2002\)](#) propôs as seguintes relações para a atualização da taxa de perturbação e da probabilidade de cruzamento no algoritmo de ED:

$$F = \begin{cases} \sqrt{\frac{1}{NP}} \sqrt{\frac{\eta}{2CR}} & \text{se } \eta \geq 0 \\ F_{\min} & \text{se } \eta < 0 \end{cases} \quad (13)$$

$$CR = \begin{cases} -(NPF^2 - 1) + \sqrt{(NPF^2 - 1)^2 - NP(1 - c)} & \text{se } c \geq 1 \\ CR_{\min} & \text{se } c < 1 \end{cases} \quad (14)$$

onde  $\eta \equiv NP(c - 1) + CR(2 - CR)$ ,  $F_{\min}$  é o valor mínimo para  $F$  ( $F_{\min} = \sqrt{1/NP}$ ) e o valor máximo como recomendado por [Price et al.](#)

(2005) ( $F_{max}=2$ ). No caso de  $CR$ , este deve pertencer ao intervalo  $]0\ 1]$ , e o parâmetro  $c$  é definido como:

$$c \equiv \gamma \frac{Var(x(g+1))}{Var(x(g))} \tag{15}$$

No problema inverso considerado neste estudo de caso, o objetivo é a determinação da espessura óptica ( $\tau_o$ ), do albedo de espalhamento simples ( $\omega$ ), das reflectividades  $\rho_1$  e  $\rho_2$ , e dos coeficientes da função de fase de espalhamento ( $b_k, k=1, 2, \dots, M$ ). O meio é considerado homogêneo, i.e.  $\omega=$ constante. São analisadas duas funções de fase, FF-1 e FF-2, cujos coeficientes são listados na Tabela 4. Além disso, para a geração dos pontos experimentais sintéticos, são considerados os seguintes valores para as variáveis de projeto:  $\tau_o=1, \omega=0,5, \rho_1=\rho_2=0,2$ . Para as funções de fase são feitas aproximações polinomiais de ordem  $M$ ,

$$p(\mu, \mu') = \sum_{k=0}^M b_k P_k(\mu) P_k(\mu') \tag{16}$$

onde  $b_o = 1$  e  $P_k$  representam o polinômio de Legendre.

Tabela 4. Coeficientes das funções de fase FF-1 e FF-2 (Chu et al., 1957).

Coeficiente	<i>Forward Scattering</i>	<i>Backward Scattering</i>
	<b>FF-1</b> ( $M=4, m_1=1,4$ e $\nu=1$ )	<b>FF-2</b> ( $M=5, m_1=\infty$ e $\nu=1$ )
$b_1$	0,57024	-0,56524
$b_2$	0,56134	0,29783
$b_3$	0,11297	0,08571
$b_4$	0,01002	0,01003
$b_5$	0,00000	0,00063
$b_6$	-	0,00000

$m_1$  é o índice de refração,  $\nu$  é  $\pi D/\lambda$ ,  $D$  é o diâmetro da partícula e  $\lambda$  é o comprimento de onda da radiação incidente.

Para comparar os resultados obtidos pelo algoritmo ADE - *Adaptive Differential Evolution* (com parâmetros auto-adaptativos) utiliza-se três combinações do algoritmo de ED canônico (com parâmetros constantes): ED-1  $\{F=0,5$  e  $CR=0,5\}$ , ED-2  $\{F=0,5$  e  $CR=0,8\}$  e ED-3  $\{F=0,8$  e  $CR=0,5\}$ . Os outros parâmetros usados são: 10 indivíduos, 1000 gerações e estratégia DE/RAND/1/BIN para a geração de candidatos em potencial. No algoritmo ADE,  $\gamma$  foi considerado 1 em todas as execuções. Em todos os casos foi considerado  $A_1=1$  e  $A_2=0$ . Além disso, foram considerados os pontos experimentais sem ruído e com ruído de 5%. Finalmente, o critério de parada considerado foi a homogeneidade para população, isto é, a diferença entre o melhor e o pior valor da população, em termos de função objetivo, ser menor que  $10^{-9}$ .

A Tabela 5 apresenta os resultados obtidos pelos algoritmos ED e ADE para as funções de fase FF-1 e FF-2.

Tabela 5. Resultados obtidos usando os algoritmos ED e ADE, onde *Erro* é dado em %,  $N_{eval}$  é o número de avaliações da função objetivo e  $Q$  se refere à Equação 6.

	FF-1						FF-2					
	Erro	Exato	ED-1	ED-2	ED-3	ADE	Exato	ED-1	ED-2	ED-3	ADE	
$\tau_0$	0	1,0	0,9999	0,9983	0,9999	1,0000	1,0	0,9992	0,9991	0,9990	0,9997	
	5		0,9786	1,0195	0,9879	0,9867		0,0004	0,9944	0,9945	0,9927	
$\omega$	0	0,5	0,4999	0,4995	0,4999	0,5000	0,5	0,4996	0,4991	0,4997	0,4999	
	5		0,4914	0,5108	0,4952	0,4955		0,4912	0,4887	0,4886	0,4836	
$\rho_1$	0	0,2	0,1999	0,2000	0,2000	0,1999	0,2	0,2000	0,2000	0,1999	0,2000	
	5		0,1998	0,1994	0,2003	0,2001		0,2046	0,2046	0,2046	0,2054	
$\rho_2$	0	0,2	0,1950	0,1996	0,2100	0,1998	0,2	0,1980	0,1986	0,1997	0,1997	
	5		0,2244	0,2307	0,1823	0,1988		0,2200	0,2140	0,2012	0,1948	
$b_1$	0	0,5702	0,5701	0,5607	0,5702	0,5703	-0,5652	-0,5708	-0,5700	-0,5704	-0,5704	
	5		0,4961	0,4924	0,4840	0,4932		-0,6217	-0,6669	-0,6669	-0,6669	
$b_2$	0	0,5613	0,5611	0,5587	0,5610	0,5612	0,2978	0,2948	0,2958	0,2949	0,2968	
	5		0,5104	0,5329	0,5117	0,5369		0,2680	0,2650	0,2650	0,2650	
$b_3$	0	0,1129	0,1117	0,1101	0,1116	0,1132	0,0857	0,0848	0,0845	0,0845	0,0846	
	5		0,0931	0,0990	0,1021	0,1041		0,0771	0,0882	0,0933	0,0794	
$b_4$	0	0,0100	0,0099	0,0105	0,0105	0,0099	0,0100	0,0099	0,0099	0,0099	0,0099	
	5		0,0082	0,0125	0,0107	0,0094		0,0110	0,0117	0,0118	0,0115	
$b_5$	0	0,0000	0,0000	0,0000	0,0000	0,0000	0,0006	0,0006	0,0006	0,0006	0,0006	
	5		0,0013	0,0012	0,0014	0,0010		0,0006	0,0006	0,0006	0,0006	
$b_6$	0	-	-	-	-	-	0,0000	0,0000	0,0000	0,0000	0,0000	
	5		-	-	-	-		0,0009	0,0007	0,0001	0,0001	
$Q$	0	-	6,5E-10	2,5E-10	8,1E-10	4,3E-10	-	4,7E-10	5,7E-10	4,4E-10	3,7E-10	
	5		1,2E-02	1,3E-02	1,3E-02	1,2E-02		1,2E-02	1,2E-02	1,2E-02	1,2E-02	
$N_{eval}$	0	-	2960	3050	2580	2010	-	2420	2880	2380	1920	
	5		1220	1660	1480	1060		1200	1680	1540	1100	

Na Tabela 5 é importante observar que, considerando dados sem ruído, ambos os algoritmos ED e ADE foram capazes de estimar os parâmetros satisfatoriamente. Entretanto, o algoritmo ADE o faz, à custa de um menor número de avaliações da função objetivo quando comparado ao algoritmo de ED canônico (redução de 32%, 34% e 22% no número de avaliações da função objetivo relativos aos algoritmos ED-1, ED-2 e ED-3, respectivamente). Boas estimativas também são obtidas quando ruído é considerado.

A Tabela 6 apresenta os erros médios obtidos pelos algoritmos de ED e ADE considerando as funções de fase FF-1 e FF-2.

Tabela 6. Erros médios obtidos usando os algoritmos de ED e ADE (todos valores dados em %).

Erro	FF-1				FF-2			
	ED-1	ED-2	ED-3	ADE	ED-1	ED-2	ED-3	ADE
0	0,0743	0,1981	0,1353	0,0067	0,1267	0,1108	0,1081	0,0783
5	2,2508	2,0608	2,0407	1,4522	1,7072	1,7501	1,7657	1,6689

Na Tabela 6 é possível observar que ambos os algoritmos foram capazes de estimar satisfatoriamente as funções de fase, do albedo de espalhamento

e da espessura óptica. Entretanto, é importante enfatizar que o algoritmo ADE o faz com um menor erro médio com relação ao algoritmo de ED canônico.

**4.4 Problema de transferência radiativa composto por duas camadas**

Este estudo considera um meio participante com duas camadas sujeito à radiação externa em ambos os lados com intensidade  $A_1(\mu)$  em  $x=0$  e  $A_2(\mu)$  em  $x=L_1+L_2$ , onde  $\mu$  é o cosseno do ângulo polar,  $L_1$  e  $L_2$  representam a espessura das camadas 1 e 2, respectivamente, e  $\rho_j$  são as reflectividades difusas ( $j=1, \dots, 4$ ), como apresentado na Figura 3.

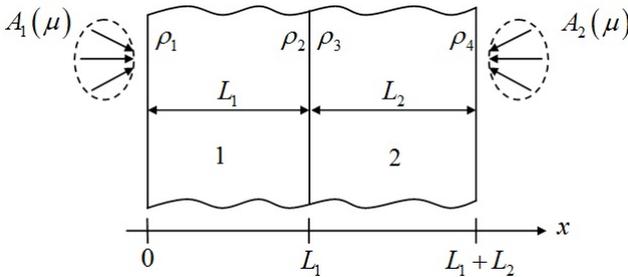


Figura 3. Meio semitransparente a duas camadas.

Este problema pode ser formulado matematicamente como:

Região 1 ( $0 \leq x \leq L_1$ )

$$\mu \frac{\partial I_1(x, \mu)}{\partial x} + \beta_1 I_1(x, \mu) = \frac{\sigma_{s1}}{2} \int_{-1}^1 I_1(x, \mu') d\mu', \quad -1 \leq \mu \leq 1 \quad (17)$$

$$I_1(0, \mu) = A_1(\mu) + 2\rho_1 \int_0^1 I_1(0, -\mu') \mu' d\mu', \quad \mu > 0 \quad (18)$$

$$I_1(L_1, \mu) = 2(1 - \rho_3) \int_0^1 I_2(L_1, -\mu') \mu' d\mu' + 2\rho_2 \int_0^1 I_1(L_1, \mu') \mu' d\mu', \quad \mu < 0 \quad (19)$$

Região 2 ( $L_1 \leq x \leq L_1 + L_2$ )

$$\mu \frac{\partial I_2(x, \mu)}{\partial x} + \beta_2 I_2(x, \mu) = \frac{\sigma_{s2}}{2} \int_{-1}^1 I_2(x, \mu') d\mu', \quad -1 \leq \mu \leq 1 \quad (20)$$

$$I_2(L_1, \mu) = 2(1 - \rho_2) \int_0^1 I_1(L_1, \mu') \mu' d\mu' + 2\rho_3 \int_0^1 I_2(L_1, -\mu') \mu' d\mu', \quad \mu > 0 \quad (21)$$

$$I_2(L_1 + L_2, \mu) = A_2(\mu) + 2\rho_4 \int_0^1 I_2(L_1 + L_2, \mu') \mu' d\mu', \quad \mu < 0 \quad (22)$$

onde  $I(x, \mu)$  representa a intensidade da radiação na camada  $i$ , com  $i=1$  ou  $2$ ,  $\beta_i$  é o coeficiente de extinção dado por:

$$\beta_i = k_{ai} + \sigma_{si} \quad (23)$$

onde  $k_{ai}$  representa o coeficiente de absorção e  $\sigma_{si}$  o coeficiente de espalhamento. Nas Equações 19 e 21 foi considerado que tanto a reflexão quanto a transmissão são difusas na interface entre as duas camadas.

Na presente aplicação são usados os seguintes parâmetros no algoritmo de ED: 10 indivíduos, 100 gerações, taxa de perturbação e probabilidade de cruzamento iguais a 0,8 e a estratégia DE/RAND/1/BIN para a geração de candidatos em potencial.

As propriedades consideradas no presente estudo de caso para a resolução do problema direto são:  $L_1=0,8$  cm;  $L_2=3,2$  cm;  $\sigma_{s1}=0,8$  cm<sup>-1</sup>;  $k_{a1}=0,5$  cm<sup>-1</sup>;  $\sigma_{s2}=0,9$  cm<sup>-1</sup>;  $k_{a2}=0,3$  cm<sup>-1</sup>;  $\rho_1=0,1$ ;  $\rho_2=0,0$ ;  $\rho_3=0,0$ ;  $\rho_4=0,6$ ;  $A_1(\mu)=0,3$  e  $A_2(\mu)=1,0$ . O seguinte espaço de projeto é definido:  $0,4 \leq \sigma_{s1} \leq 1,2$ ,  $0 \leq k_{a1} \leq 1$ ,  $0,4 \leq \sigma_{s2} \leq 1,5$  e  $0 \leq k_{a2} \leq 1$ .

A Tabela 7 apresenta os resultados para o problema do meio semitransparente em duas camadas para os dados sem ruído.

Tabela 7. Resultados para o problema do meio semitransparente em duas camadas (dados sem ruído), onde  $Q$  se refere à Equação 6.

	$\sigma_{s1}$	$k_{a1}$	$\sigma_{s2}$	$k_{a2}$	$Q$	DI
Exato	0,8	0,5	0,9	0,3	-	-
ED	0,7999	0,5000	0,9000	0,3000	1,14E-10	Sim

Na Tabela 8, quando ruído é acrescentado ( $\sigma=0,1\%$  e  $\sigma=0,2\%$ ), o algoritmo de ED consegue obter bons resultados quando detectores internos são usados.

Este problema inverso foi proposto e resolvido por Soeiro & Silva Neto (2006) usando as seguintes técnicas: Rezoimento Simulado (RS), o Método de Levenberg-Marquardt (LM) e um híbrido envolvendo o Rezoimento Simulado e Levenberg-Marquardt (RS-LM). Entretanto, é importante ressaltar que estes autores utilizaram outras condições de contorno, o que inviabilizou a comparação com os resultados obtidos neste capítulo pelo algoritmo de ED.

Tabela 8. Resultados para o problema do meio semitransparente em duas camadas (dados com ruído).

	$\sigma_{s1}$	$k_{a1}$	$\sigma_{s2}$	$k_{a2}$	$Q$	Solução	DI
	Ruído ( $\sigma$ ) igual a 0,1%						
Exato	0,8	0,5	0,9	0,3	-	-	-
ED	0,8010	0,4990	0,8980	0,2990	1,96E-06	Boas estimativas	Sim
	Ruído ( $\sigma$ ) igual a 0,2%						
Exato	0,8	0,5	0,9	0,3	-	-	-
ED	0,7989	0,4979	0,8960	0,2983	4,56E-04	Boas estimativas	Sim

Mais detalhes sobre essa aplicação pode ser encontrada em Lobato et al. (2010c).

### 4.5 Problema de transferência de calor por radiação-condução

Este último estudo de caso ilustra a aplicação do algoritmo de ED no contexto multi-objetivo para a determinação de propriedades radiativas e térmicas no problema acoplado de transferência de calor por radiação-condução. A equação de transferência de calor por radiação considerando simetria azimutal e espalhamento isotrópico é dado por (Silva Neto & Özişik, 1993; Silva Neto & Moura Neto, 2005):

$$\mu \frac{\partial}{\partial \tau} I(\tau, \mu) + I(\tau, \mu) = H(\Theta) + \frac{\omega}{2} \int_{-1}^1 I(\tau, \mu') d\mu', \tau \in (0, \tau_o), \mu \in [-1, 1] \tag{24}$$

sujeito às condições de contorno  $I(0, \mu) = A_1 = 1$  e  $I(\tau_o, \mu) = A_2 = 0$  ( $\rho_1 = \rho_2 = 0$ ). O termo  $H(\Theta)$  é o termo fonte relacionado à temperatura adimensional definido como  $(1 - \omega)\Theta^4(\tau)$ .

A formulação matemática uni-dimensional em estado estacionário do problema combinado condução-radiação, na forma adimensional, é dado pela equação de Poisson (Silva Neto & Özişik, 1993; Silva Neto & Moura Neto, 2005):

$$\frac{d^2 \Theta}{d\tau^2}(\tau) - \frac{(1 - \omega)}{N} [\Theta^4(\tau) - G(\tau)] = 0, 0 < \tau < \tau_o \tag{25}$$

sujeito às condições de contorno  $\Theta(0) = 1$  e  $\Theta(\tau_o) = T_2/T_1$ .

$$G = \frac{1}{2} \int_{-1}^1 I(\tau, \mu) d\mu, N = \frac{k\beta}{4n_i^2 \sigma T_1^3}, \Theta = \frac{T}{T_1} \tag{26}$$

onde  $N$  é o parâmetro de condução-radiação,  $\Theta$  é a temperatura adimensional,  $k$  é a condutividade térmica,  $\beta$  é o coeficiente de extinção,  $n_i$  é o

índice de refração,  $\hat{\sigma}$  é a constante de Stefan-Boltzmann e os contornos  $\tau=0$  e  $\tau=\tau_o$  são mantidos nas temperaturas  $T_1$  e  $T_2$ , respectivamente.

O problema inverso multi-objetivo proposto consiste na determinação do parâmetro de condução-radiação ( $N$ ), da espessura óptica ( $\tau_o$ ) e do albedo de espalhamento simples ( $\omega$ ), através da minimização dos funcionais:

$$Q_1 = \sum_{i=1}^{N_{dI}} \left( I_i^{cal}(\tau_o, \omega, N) - I_i^{exp} \right)^2 \quad (27)$$

$$Q_2 = \sum_{i=1}^{N_{d\theta}} \left( \Theta_i^{cal}(\tau_o, \omega, N) - \Theta_i^{exp} \right)^2 \quad (28)$$

onde  $N_{dI}$  e  $N_{d\theta}$  são respectivamente o número de pontos experimentais considerados para a intensidade da radiação e para a temperatura.

Até o presente momento, tinha sido considerado apenas o problema de otimização com um único objetivo. Como discutido acima, este estudo é um problema multi-objetivo que necessariamente necessita de um tratamento diferenciado. Neste contexto, será utilizado o algoritmo MODE (*Multi-Objective Optimization Differential Evolution*) desenvolvido por [Lobato \(2008\)](#) e que consiste basicamente do acoplamento entre o algoritmo de ED e o critério de dominância de Pareto, associado a estratégias para a exploração de vizinhos em potencial e de um operador para o truncamento das soluções não-dominadas. Mais detalhes a respeito do algoritmo MODE podem ser encontrados no trabalho de [Lobato \(2008\)](#).

Os parâmetros usados pelo algoritmo NSGA II (*Nondominated Sorting Genetic Algorithm*) ([Deb et al., 2002](#)) foram: 50 indivíduos, 250 gerações, probabilidade de cruzamento igual a 0,8 e probabilidade de mutação igual a 0,01. Para os parâmetros considerados, o número de avaliações da função objetivo é 12550.

Os parâmetros usados pelo algoritmo MODE foram: 10, 20 e 50 indivíduos, 150 gerações, taxa de perturbação e probabilidade de mutação iguais a 0,8, taxa de redução e número de pseudo curvas iguais a 0,9 e 10, respectivamente, e estratégia DE/RAND/1/BIN para a geração de candidatos em potencial. Para os parâmetros considerados, o número de avaliações da função objetivo são 3010, 6020 e 15050, respectivamente.

O parâmetro  $\Gamma$  é calculado usando como referência a origem do eixo coordenado, isto é, o ponto (0,0) é usado para obter a distância entre este ponto e cada ponto na curva de Pareto. A menor distância obtida foi escolhida como critério apresentado nas tabelas a seguir.

Foram usados 120 pontos experimentais para a intensidade de radiação e 25 pontos experimentais para o perfil de temperatura. O critério de parada utilizado foi o número máximo de gerações.

Neste estudo de caso é considerado o problema sem ruído e com ruído que corresponde a 5% de erro. Considerando os dados sem ruído (Tabela 9), ambos os algoritmos NSGA II e MODE foram capazes de estimar os parâmetros satisfatoriamente como pode ser observado pelo valor das funções objetivo listadas nesta tabela. Contudo, o algoritmo MODE com 10 e 20 indivíduos nos conduz a um melhor número de avaliações da função objetivo quando comparado com o algoritmo NSGA II (uma redução de 76% e 52% no número de avaliações da função objetivo, respectivamente).

Tabela 9. Resultados obtidos usando os algoritmos NSGA II e MODE considerando os dados sem ruído.

	<i>NP</i>	$\Gamma$					
		$\omega$	$\tau_o$	<i>N</i>	$Q_1$ (Eq. 27)	$Q_2$ (Eq. 28)	$N_{eval}$
Valores Exatos		0,3	0,5	0,01	-	-	-
NSGA II	50	0,2822	0,5001	0,0098	4,7016E-07	6,6917E-07	12550
	10	0,2999	0,5000	0,0099	7,8588E-14	2,7787E-13	3010
MODE	20	0,2999	0,4999	0,0099	1,4787E-15	7,7707E-15	6020
	50	0,2999	0,5000	0,0100	7,7777E-16	1,4887E-17	15050
Valores Exatos		0,9	1,5	0,10	-	-	-
NSGA II	50	0,9000	1,5000	0,0100	3,6336E-09	6,5855E-09	12550
	10	0,9000	1,4999	0,0100	2,5689E-13	2,5478E-13	3010
MODE	20	0,9000	1,5000	0,0100	9,6985E-14	2,3222E-14	6020
	50	0,9000	1,5000	0,0999	7,4414E-15	2,5665E-16	15050

*NP*: número de indivíduos

Na Tabela 10, quando ruído de 5% é considerado, ambos os algoritmos NSGA II e MODE são capazes de obter boas estimativas. Como observado na tabela anterior, a mesma qualidade de solução obtida pelo algoritmo NSGA II é também observada pelo algoritmo MODE, contudo com um menor número de avaliações da função objetivo para os casos onde são usados 10 e 20 indivíduos na população. Esta vantagem não é observada para populações com mais indivíduos devido ao mecanismo de refinamento de soluções proposto pelo algoritmo MODE, o que nos conduz naturalmente ao aumento do número de avaliações da função objetivo.

Mais detalhes sobre esta aplicação pode ser encontrada em Lobato et al. (2009).

## 5. Considerações Finais

Este capítulo teve por objetivo a resolução de problemas inversos em processos difusivos e transferência radiativa usando o algoritmo de ED. Neste contexto, o algoritmo mostrou-se eficiente quando comparado com outras técnicas clássicas e heurísticas, além de poder ser verificada uma redução no número de avaliações da função objetivo quando comparado, por exemplo, com o algoritmo de Recozimento Simulado. Deve ser ressaltado que

Tabela 10. Resultados obtidos usando os algoritmos NSGA II e MODE considerando os dados com ruído de 5%.

	$\Gamma$						
	$NP$	$\omega$	$\tau_o$	$N$	$Q_1$ (Eq. 27)	$Q_2$ (Eq. 28)	$N_{eval}$
Valores Exatos		0,3	0,5	0,01	-	-	-
NSGA II	50	0,3074	0,4978	0,0086	1,2629E-02	2,7281E-03	12550
	10	0,3066	0,4980	0,0088	9,4785E-03	2,1286E-03	3010
MODE	20	0,3066	0,4981	0,0087	7,7871E-03	1,6968E-03	6020
	50	0,3067	0,4980	0,0089	8,7847E-03	1,8678E-03	15050
Valores Exatos		0,9	1,5	0,10	-	-	-
NSGA II	50	0,8966	1,4889	0,0978	1,3514E-02	3,5346E-03	12550
	10	0,8977	1,4890	0,0981	2,2547E-03	2,6774E-03	3010
MODE	20	0,8978	1,4890	0,0981	2,2546E-03	2,5899E-03	6020
	50	0,8978	1,4890	0,0982	2,1487E-03	1,2588E-03	15050

$NP$ : número de indivíduos

este critério é importante porque, no caso de sistemas de engenharia representados por modelos complexos (envolvendo, por exemplo, a resolução de equações diferenciais, de equações integro-diferenciais, simulações de Monte Carlo, ou a resolução de sistemas de equações de grande dimensão), a avaliação da função objetivo frequentemente é, computacionalmente, a parte mais dispendiosa do processo de otimização.

Com relação à atualização dos parâmetros do algoritmo de ED, mostrou-se que é possível aumentar a diversidade da população, refletida na qualidade dos resultados obtidos, e consequentemente reduzir as chances de convergência prematura, fato este que pode ocorrer durante a execução dos algoritmos heurísticos. Outro ponto a se destacar é a eliminação da dependência do usuário na escolha dos parâmetros do algoritmo de ED, apesar de existir uma faixa *default* que é comumente utilizada em aplicações afins.

Finalmente, deve ser destacado que o algoritmo de ED há muito se tornou uma realidade concreta, com aplicações em áreas distintas da ciência e da engenharia, ao ponto de alguns autores (Babu et al., 2005) afirmarem que este algoritmo já superou os algoritmos clássicos fundamentados na genética de populações, no que diz respeito à convergência, ao número de avaliações da função objetivo e ao tempo de processamento.

## Agradecimentos

O Prof. Lobato e o Prof. Steffen Jr agradecem pelo apoio do INCT-EIE, através de recursos do CNPq e FAPEMIG. O Prof. Silva Neto agradece o suporte financeiro advindo da FAPERJ e do CNPq.

## Referências

- Abbass, H.; Sarker, R. & Newton, C., PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. In: *Proceedings of the 2001 Congress on Evolutionary Computation*. Piscataway, USA: IEEE Press, p. 971–978, 2001.
- Babu, B. & Anbarasu, B., Multi-objective differential evolution (MODE): An evolutionary algorithm for multi-objective optimization problems (MOOPS). In: *Proceedings of The Third International Conference on Computational Intelligence, Robotics, and Autonomous Systems*. Singapore, 2005.
- Babu, B.; Chakole, P. & Mubeen, J., Multio-objective differential evolution (MODE) for optimization of adiabatic styrene reactor. *Chemical Engineering Science*, 60(17):4822–4837, 2005.
- Chalhoub, E.; Campos Velho, H. & Silva Neto, A., A comparison of the one-dimensional radiative transfer problem solutions obtained with the monte carlo method and three variations of the discrete ordinates method. In: *Proceedings of 19<sup>th</sup> International Congress of Mechanical Engineering - COBEM*. Brasília, DF, p. 1–10, 2007.
- Chu, C.M.; Clark, G. & Churchill, S., *Tables of Angular Distribution Coefficients for Light Scattering by Spheres*. Ann Arbor, USA: Engineering Research Institute, University of Michigan, 1957.
- Deb, K.; Pratap, A.; Agarwal, S. & Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- Fan, H.Y. & Lampinen, J., A trigonometric mutation operation to differential evolution. *Journal of Global Optimization*, 27(1):105–129, 2003.
- Kapadi, M. & Gudi, R., Optimal control of fed-batch fermentation involving multiple feeds using differential evolution. *Process Biochemistry*, 39(11):1709–1721, 2004.
- Lobato, F., *Otimização Multi-objetivo para o Projeto de Sistemas de Engenharia*. Tese de doutorado, Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia, MG, 2008.
- Lobato, F.; Murata, V.; Oliveira-Lopes, L. & Steffen Jr., V., Solution of multi-objective optimal control problems with index fluctuation using differential evolution. In: *Proceedings in 6<sup>th</sup> Brazilian Conference on Dynamics, Control and Applications - DINCON*. São José do Rio Preto, SP, p. 1–8, 2007a.
- Lobato, F.; Oliveira-Lopes, L.; Murata, V. & Steffen Jr., V., Evolução diferencial aplicada à solução de problemas de controle ótimo com restrições de desigualdade. In: *Anais do X Encontro de Modelagem Computacional*. Nova Friburgo, Brasil, p. 1–10, 2007b.

- Lobato, F. & Steffen Jr., V., Engineering system design with multi-objective differential evolution. In: *Proceedings in 19<sup>th</sup> International Congress of Mechanical Engineering - COBEM*. Brasília, Brazil, 2007.
- Lobato, F.; Steffen Jr, V.; Arruda, E. & Barrozo, M., Estimation of drying parameters in rotary dryers using differential evolution. *Journal of Physics: Conference Series*, 135(1):012063, 2008.
- Lobato, F.; Steffen Jr., V. & Silva Neto, A., Solution of the coupled inverse conduction-radiation problem using multi-objective optimization differential evolution. In: *Proceedings of 8<sup>th</sup> World Congress on Structural and Multidisciplinary Optimization*. Lisboa, Portugal, p. 1–10, 2009.
- Lobato, F.; Steffen Jr., V. & Silva Neto, A., Evolução diferencial (*Differential Evolution*). In: Silva Neto, A. & Becceneri, J. (Eds.), *Técnicas de Inteligência Computacional Inspiradas na Natureza – Aplicações em Problemas Inversos em Transferência Radiativa*. São Carlos: SBMAC, v. 41 de *Notas em Matemática Aplicada e Computacional*, p. 91–100, 2009a.
- Lobato, F.; Steffen Jr., V. & Silva Neto, A., A comparative study of the application of differential evolution and simulated annealing in inverse radiative transfer problems. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 32(special):518–526, 2010a.
- Lobato, F.; Steffen Jr, V. & Silva Neto, A., Estimation of space-dependent single scattering albedo in radiative transfer problems using differential evolution algorithm. In: *Proceedings of Inverse Problems, Design and Optimization Symposium*. João Pessoa, PB, p. 1–8, 2010b.
- Lobato, F.; Steffen Jr., V. & Silva Neto, A., Solution of inverse radiative transfer problems in two-layer participating media with differential evolution. *Inverse Problems in Science and Engineering*, 18(2):183–195, 2010c.
- Mariani, V.; Lima, A. & Coelho, L., Apparent thermal diffusivity estimation of the banana during drying using inverse method. *Journal of Food Engineering*, 85(4):569–579, 2008.
- Oliveira-Filho, R.; Lobato, F.; Duarte, M. & Steffen Jr., V., Estimação dos parâmetros de um filtro adaptativo utilizado como ferramenta para controle de qualidade em linha de montagem de câmbios automotivos através da técnica de evolução diferencial. In: *Anais do V Congresso Nacional de Engenharia Mecânica*. Salvador, BA, 2008.
- Özişik, M., *Radiation Transfer and Iterations with Conduction and Convection*. 1a edição. John Wiley & Sons, 1973.
- Price, K., Genetic annealing. *Dr Bobb's Journal*, 19(10):127–132, 1994.
- Price, K.; Storn, R. & Lampinen, J., *Differential Evolution - A Practical Approach to Global Optimization*. Natural Computing Series. Berlin, Germany: Springer-Verlag, 2005.

- Silva Neto, A. & Moura Neto, F., *Problemas Inversos – Conceitos Fundamentais e Aplicações*. 1a edição. Rio de Janeiro, RJ: Editora da Universidade do Estado do Rio de Janeiro, 2005.
- Silva Neto, A. & Özişik, M., An inverse problem of estimating thermal conductivity, optical thickness, and single scattering albedo of a semi-transparent medium. In: *Proceedings of First International Conference on Inverse Problems in Engineering: Theory and Practice*. p. 267–273, 1993.
- Soeiro, F. & Silva Neto, A., Inverse radiative transfer problems in two-layer participating media. In: Motasoaes, C.; Martins, J.; Rodrigues, H.; Ambrósio, J.A.; Pina, C.B.; Motasoaes, C.; Pereira, E. & Folgado, J. (Eds.), *Proceedings of III European Conference on Computational Mechanics*. Springer Netherlands, p. 477, 2006.
- Stephany, S.; Beccegeri, J.; Souto, R.; Campos Velho, H. & Silva Neto, A., A pre-regularization scheme for the reconstruction of a spatial dependent scattering Albedo using a hybrid ant colony optimization implementation. *Applied Mathematical Modelling*, 34(3):561–572, 2010.
- Storn, R. & Price, K., Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, USA, 1995.
- Storn, R. & Price, K., Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.
- Villadsen, J. & Michelsen, M., *Solution of Differential Equation Models by Polynomial Approximation*. Englewood Cliffs, USA: Prentice-Hall, 1978.
- Wylie, C. & Barrett, L., *Advanced Engineering Mathematics*. 6a edição. New York, USA: McGraw-Hill, 1995.
- Zahaire, D., Critical values for the control parameters of differential evolution algorithms. In: R. Matousek, P.O. (Ed.), *Proceedings of the 8<sup>th</sup> International Conference on Soft Computing*. Brno, Czech Republic, p. 62–67, 2002.
- Zahaire, D., Control of population diversity and adaptation in differential evolution algorithms. In: R. Matousek, P.O. (Ed.), *Proceedings of 9<sup>th</sup> International Conference on Soft Computing*. Brno, Czech Republic, p. 41–46, 2003.

## Notas Biográficas

**Fran Sérgio Lobato** tem mestrado em Engenharia Química e doutorado em Engenharia Mecânica. Atualmente é Professor Adjunto da Faculdade de Engenharia Química da Universidade Federal de Uberlândia. Tem interesse no desenvolvimento e aplicação de métodos de simulação e otimização, especialmente em problemas inversos, e em problemas de controle ótimo singulares.

**Valder Steffen Jr.** obteve seu doutorado na *Franche-Comté University* (Besançon, França). É Professor Titular de Engenharia Mecânica da Universidade Federal de Uberlândia. Tem interesse em dinâmica de sistemas mecânicos, em algoritmos de otimização, no estudo de dinâmica de sistemas mecânicos e mecatrônicos, e em problemas inversos.

**Antônio J. Silva Neto** concluiu o doutorado em Engenharia Mecânica na *North Carolina State University*. É professor Adjunto da Universidade do Estado do Rio de Janeiro. Atua nas áreas de transferência de calor e massa e de matemática aplicada e computacional, com ênfase em métodos numéricos e problemas inversos.



## Algoritmo Genético com Interação Social na Resolução de Problemas de Otimização Global com Restrições

Otávio Noura Teixeira\*, Walter Avelino da Luz Lobato,  
Hitoshi Seki Yanaguibashi, Rodrigo Vieira Cavalcante,  
Deam James Azevedo da Silva e Roberto Célio Limão de Oliveira

---

**Resumo:** Este capítulo apresenta uma nova meta-heurística híbrida de otimização baseada na natureza, denominada de Algoritmo Genético com Interação Social (SIGA), e é aplicada na resolução de quatro problemas de engenharia bastante difundidos na literatura: (a) projeto de viga de aço; (b) projeto de vaso de pressão; (c) minimização do peso da tensão/compressão sobre mola; (d) projeto de redutor de velocidade. Esta nova abordagem é uma tentativa de imitar um pouco mais de perto o processo evolutivo de uma população ao longo do tempo, com ênfase na interação social entre os indivíduos, como na evolução social humana. Assim, este capítulo apresenta os fundamentos necessários para a concepção do SIGA, a sua estrutura e os resultados obtidos nas simulações.

**Palavras-chave:** Algoritmo genético, teoria dos jogos, meta-heurística híbrida, algoritmo SIGA.

**Abstract:** *This chapter presents a new hybrid nature-inspired meta-heuristic named Genetic Algorithm with Social Interaction (SIGA). It is used to solve four engineering problems usually found in the literature: (a) welded beam design; (b) design of a pressure vessel; (c) minimization of the weight of a tension/compression spring; (d) design of a speed reducer. This new approach is an attempt to mimic a little more closer how a population evolves over time with emphasis on social interaction between individuals, as in human social evolution. Therefore, this chapter presents the necessary fundamentals for the conception of SIGA, its structure and the results obtained in the simulations.*

**Keywords:** *Genetic algorithm, game theory, hybrid meta-heuristic, SIGA's algorithm.*

---

\* Autor para contato: onoura@gmail.com

## 1. Introdução

Nos últimos dez anos, alguns trabalhos foram desenvolvidos utilizando conceitos da Teoria dos Jogos Evolucionários (TJE) e Teoria dos Jogos (TJ) nos Algoritmos Genéticos (AG), através do uso dos jogos Hawk-Dove, Dilema do Prisioneiro e ainda outros tipos de jogos (Lehrer, 2000; Teixeira, 2005; Brito et al., 2005; Teixeira et al., 2006b,a; Lahoz-Beltra et al., 2009; Teixeira et al., 2010b,a).

Estas abordagens foram estabelecidas com base na observação de como a natureza funciona, como, por exemplo, uma colônia de formigas, um enxame de abelhas e, até mesmo, a sociedade humana. Além disto, o Algoritmo Genético de Holland (1975), o Algoritmo Genético Simples (*Simple Genetic Algorithm - SGA*) de Goldberg (1989) e todos os seus descendentes diretos não implementam, de fato, o conceito de fenótipo e, sim, somente afirmam que ele é o valor de *fitness* do indivíduo.

No entanto, a definição de fenótipo é bem mais abrangente na Biologia, do que esta empregada na teoria dos Algoritmos Genéticos, sendo que ele, o fenótipo, contribui, e muito, para o processo evolutivo dos indivíduos da população. Segundo Dawkins (1999), o comportamento de um animal tende a maximizar a sobrevivência de seus genes para aquele comportamento, mesmo que eles estejam ou não no seu corpo. Ou seja, o conceito de fenótipo está também intimamente relacionado ao comportamento dos indivíduos, que é fator determinante para a sobrevivência dos seus genes.

Desta forma, uma questão pode ser feita e necessita de resposta: no contexto dos Algoritmos Genéticos há alguma maneira que permita aos indivíduos alterar o seu valor de *fitness* durante o processo evolutivo? A resposta é: “Não, não há! Mas, sim, é possível! Mas como?”. A Teoria dos Jogos pode ser utilizada como pilar teórico fundamental para caracterizar uma nova abordagem fenotípica para os Algoritmos Genéticos, pois formaliza matematicamente situações de conflito de interesses, onde agentes racionais devem tomar decisões com o objetivo de maximizar os seus ganhos.

Além disto, é importante e necessário também mencionar os três principais aspectos que possibilitam a evolução de uma população de indivíduos, estabelecidos por (Darwin, 1859), que são: (a) diversidade em uma população de uma mesma espécie; (b) luta pela sobrevivência; (c) seleção natural. Assim, ao compreender melhor a estrutura básica e os fundamentos dos Algoritmos Genéticos, é possível afirmar que estes três aspectos são contemplados sim, porém não como ocorre na natureza.

Como consequência se faz necessário a constatação de alguns fatos e uma breve discussão sobre estes três aspectos, aplicados ao contexto dos Algoritmos Genéticos. Neles, a diversidade populacional tende a decrescer ao longo do processo evolutivo e, com isto, é possível e comum ocorrer, ao seu final, uma homogeneidade dos indivíduos, ou seja, eles passam a

ser idênticos. Isto é, há uma convergência para ótimos locais, pois não há nenhum mecanismo de manutenção de variabilidade em sua estrutura clássica como, por exemplo: a ocorrência de um desastre natural, uma epidemia virótica ou, simplesmente, a possibilidade de alteração do valor de *fitness* dos indivíduos.

Em relação à luta pela sobrevivência, Mitchell (1998) afirma que todos os indivíduos nascem com seus valores de *fitness* fixos e imutáveis e, assim, pode-se afirmar que os AG não têm, também, nenhum mecanismo que permita a ocorrência de disputas entre os indivíduos, de forma que lhes permita evoluírem e melhorarem a sua adaptabilidade, com a alteração de seus valores de *fitness*, sem que se transformem em outros indivíduos.

E o processo de seleção natural não é tão natural quanto o que ocorre na natureza, pois é um modelo bastante simplificado e que privilegia e muito os indivíduos mais bem adaptados, com uma possibilidade maior de reproduzir e passar, para as próximas gerações o seu material genético. Isto ocorre independentemente do mecanismo de seleção utilizado como, por exemplo, os métodos: (a) roleta, que é o pior método existente e, nele, todos os indivíduos têm alguma chance de serem selecionados; (b) por torneio, onde somente o pior indivíduo não tem chance alguma de ser escolhido.

Na espécie humana, por exemplo, um indivíduo tem a possibilidade de evoluir e alterar o seu valor de *fitness* através de ascensão social, mesmo que não seja geneticamente bem adaptado. Desta forma, não modifica necessariamente a sua estrutura genotípica, mas lhe permite uma maior chance de gerar descendentes e o seu material genético combinado com o material genético de seu parceiro podem gerar descendentes mais evoluídos.

Em síntese, todos estes aspectos serviram de inspiração e justificativa para a concepção e desenvolvimento de uma nova meta-heurística<sup>1</sup> denominada de Algoritmo Genético com Interação Social (SIGA). Assim, este capítulo objetiva apresentá-lo em todos os seus detalhes, além de demonstrar a sua aplicabilidade em quatro problemas da área de engenharia. No entanto, antes se faz necessário compreender conceitos fundamentais relacionados aos Algoritmos Genéticos e à Teoria dos Jogos.

## 2. Algoritmos Genéticos: Uma Visão Geral

Os Algoritmos Genéticos (AGs) são os algoritmos mais utilizados da Computação Evolucionária e estão baseados nos princípios da evolução bioló-

---

<sup>1</sup> De acordo com Blum & Roli (2003) uma Metaheurística é um conjunto de conceitos usados para definir métodos heurísticos, ou seja, é uma estrutura genérica de um algoritmo que pode ser aplicada em diferentes problemas de otimização, principalmente da classe *NP*, com o objetivo de encontrar uma solução factível – não necessariamente a ótima – em tempo de processamento aceitável.

gica dos seres vivos. Os AGs são utilizados, por exemplo, na resolução de problemas de otimização combinatória e otimização global com e sem restrições. Estes problemas são frequentemente os que requerem uma busca através de um amplo espaço de busca de soluções candidatas. Normalmente os AGs são aplicados em situações onde não há solução através de métodos algébricos ou estatísticos (Goldberg, 1989; Eberhart et al., 1996).

Os AGs são inspirados na Teoria da Origem das Espécies de Charles Darwin, onde uma população desenvolve-se durante várias gerações, segundo os princípios da Seleção Natural e da “sobrevivência do mais apto”. Os AGs são capazes de desenvolver soluções para problemas do mundo real, tais como problemas de busca e otimização (Goldberg, 1989). O detalhe é que não há garantia de se encontrar a melhor solução do problema, ou seja, o ótimo global e, sim, a melhor solução possível – que pode ser um ótimo local ou, eventualmente, o global, pois não fazem a varredura completa do espaço de busca.

Esta técnica de resolução de problemas foi desenvolvida por Holland (1975), onde os principais objetivos de sua pesquisa eram: (a) abstrair e explicar rigorosamente os processos adaptativos dos sistemas naturais; (b) desenvolver *software* de sistemas artificiais capazes de simular a robustez dos mecanismos dos sistemas naturais. Desta forma, ele criou os Algoritmos Genéticos, como uma abstração da evolução biológica, proporcionando, assim, toda a sua base teórica.

Segundo Goldberg (1989), os AGs são algoritmos de busca baseados nos mecanismos de seleção natural e genética, que têm a capacidade de evoluir uma população de indivíduos. Essa evolução ocorre através da utilização de um tipo de “seleção natural” em conjunto com outros operadores genéticos, tais como: mutação (onde um gene é substituído por um alelo<sup>2</sup> de forma aleatória), e recombinação ou cruzamento (onde os indivíduos escolhidos trocam pedaços de seus cromossomos e, assim, geram novas soluções).

Um AG busca produzir indivíduos melhores com base na combinação dos melhores existentes na população e, através da estratégia da sobrevivência do mais apto, elimina os menos aptos. Não são produzidas apenas boas soluções, mas sempre soluções melhores. Isto ocorre devido ao fato de combinarem as melhores características dos pais, na produção de filhos superiores. A seguir é apresentada a sua estrutura.

## 2.1 Estrutura básica

Atualmente, apesar da teoria dos AG ter absorvido conceitos mais aprofundados de outras áreas, tais como a biologia evolutiva e a genética das populações, por exemplo; e, além disso, ter se aproximado mais ainda da

---

<sup>2</sup> Um alelo pode ser definido como os possíveis valores que um gene pode ter. Por exemplo, se um AG tiver representação binária, os valores {0,1} são os alelos possíveis para os genes dos indivíduos.

simulação biológica, a estrutura proposta em [Holland \(1975\)](#) ainda é válida e defendida por muitos autores, conforme mostrado na Figura 1.

```

1. Início;
2. Gerar população inicial;
3. Avaliar cada indivíduo da população inicial;
4. Repetir até  $m$  gerações
// Etapa de reprodução
  4.1. Repetir até o número de descendentes ser igual à quantidade
      desejada (% de cruzamento)
    4.1.1. Selecionar dois indivíduos conforme o método de seleção;
    4.1.2. Realizar a operação de cruzamento nos indivíduos selecionados no
          passo anterior;
    4.1.3. Realizar a operação de mutação nos descendentes gerados no passo
          anterior;
    4.1.4. Avaliar os descendentes;
  4.2. Substituir os indivíduos da população pelos novos indivíduos
      gerados na etapa de reprodução;
5. O melhor indivíduo da população é a solução do problema;
6. Fim.

```

Figura 1. Estrutura básica de um Algoritmo Genético. Fonte: adaptado de [Holland \(1975\)](#); [Goldberg \(1989\)](#)

É importante observar que a sua estrutura é bastante simplificada e composta por apenas três etapas fundamentais, que são: geração da população inicial, avaliação da população inicial e a etapa de reprodução. Esta última é dividida em: seleção, cruzamento e mutação. Apesar de sua simplicidade, o poder computacional inerente a esta estrutura permite a resolução de problemas complexos, pois justamente encontram a melhor solução possível até o presente momento, enquanto o critério de parada<sup>3</sup> do algoritmo não for satisfeito.

## 2.2 Implementação de um algoritmo genético

Segundo [Haupt & Haupt \(2000\)](#), esta técnica consiste em: (a) gerar uma população de soluções (cromossomos ou indivíduos) modeladas de acordo com o problema que se deseja resolver; (b) avaliar a adaptabilidade de cada solução, ou seja, o seu valor de *fitness*, que pode ser compreendida como a probabilidade de uma solução gerar novos descendentes com seu material genético; e, (c) aplicar os operadores de seleção, cruzamento e mutação sobre as soluções, repetindo o processo até que seja alcançado o critério de parada.

De acordo com [Lehrer \(2000\)](#), a seleção é responsável por selecionar indivíduos da população, utilizando, na maioria das vezes, o valor de *fitness*, o que possibilita aos indivíduos mais aptos gerarem um número maior

<sup>3</sup> Como critério de parada, normalmente, é utilizado um número pré-definido de gerações ou então o algoritmo fica executando até encontrar um determinado valor de *fitness*.

de descendentes. Assim, é possível estabelecer uma relação direta desta operação com o processo de seleção natural proposto por Darwin (1859). É possível destacar dois métodos principais de seleção, que são: o método da Roleta ou Seleção Proporcional ao *Fitness*, e o método do Torneio.

A operação de cruzamento é a principal característica que distingue os AG de outros métodos de busca, e é considerado o mais importante entre os operadores genéticos. É através dele que o algoritmo explora a superfície de busca, dentro do material genético (alelos) que já se encontra nos indivíduos da população, o que o conduz à convergência de resultados (Haupt & Haupt, 2000). Esta operação consiste na troca de pedaços de cromossomos entre os indivíduos selecionados como pais, para a criação de novos indivíduos melhor adaptados. Existem diversos métodos de cruzamento, mas como exemplo, têm-se os tradicionais: de um ponto e de dois pontos de corte.

O operador de mutação é a segunda maneira que um AG utiliza para explorar a superfície de busca do problema. Ele introduz pequenas alterações nos cromossomos dos indivíduos ao alterar, de forma aleatória e com uma pequena probabilidade, o conteúdo de um gene por um elemento do conjunto de alelos (Eberhart et al., 1996).

É possível encontrar na literatura de AGs discussões relativas à definição dos valores para os diversos parâmetros de inicialização do algoritmo. Neste sentido, uma série de estudos já foi realizada na tentativa de encontrar um conjunto de valores ótimos. Porém, não há trabalhos conclusivos, apenas indicativos. Desta forma, os parâmetros são normalmente definidos a partir de experiências anteriores e através de casos relatados.

Os principais parâmetros de um AG são: (a) Tamanho da População: define a quantidade de indivíduos da população que serão criados a cada geração; (b) Quantidade de Gerações: é uma das condições de parada do algoritmo mais utilizadas e define a quantidade máxima de execuções da etapa de reprodução da estrutura do AG, conforme pode ser observado no passo 4 da Figura 1; (c) Taxa de Cruzamento: define a porcentagem de indivíduos que serão selecionados para gerar descendentes através do cruzamento de seus cromossomos; (d) Taxa de Mutação: define a porcentagem de indivíduos que serão selecionados para sofrer mutação em genes do seu cromossomo.

### 2.3 Considerações

Esta seção apresentou brevemente os conceitos pertinentes ao contexto dos Algoritmos Genéticos, que são necessários para a sua compreensão e que darão embasamento para a compreensão do SIGA. A partir da próxima seção são apresentados aspectos relativos aos fundamentos da Teoria dos Jogos, essenciais para a compreensão da sua aplicabilidade ao contexto dos AG.

### 3. Fundamentos da Teoria dos Jogos

Antes de serem abordados diretamente os fundamentos da Teoria dos Jogos, em Darwin (1859) é apresentado um trecho sobre a luta pela sobrevivência que ocorre na natureza e diz o seguinte:

*“[...] todos os seres vivos estão expostos a uma rigorosa competição. Nada mais fácil do que admitir a verdade da luta universal pela sobrevivência; por outro lado, nada mais difícil do que trazer a mente, o tempo todo, esta conclusão. Contudo, se assim não se fizer, ou seja, se não se cogitar tanto esta ideia até que ela fique, assim por se dizer, arraigada em nossa mente, estou convencido de que nos parecerão obscuros ou serão inteiramente mal interpretados todos os fatos relacionados com a economia da natureza, com a distribuição, com a raridade, a abundância, a extinção e a variação. A natureza nos parece brilhante e jubilosa quando em situação de superabundância de alimentos, então não vemos, ou não nos passa pela ideia, que as aves cantando alegremente ao nosso redor vivem geralmente de insetos e sementes, e que assim estão constantemente destruindo a vida, ou comumente nos esquecemos de como é frequente serem esses pássaros canoros, e também seus ovos e filhotes, destruídos pelos animais predadores; tampouco trazemos continuamente na mente a lembrança de que embora o alimento esteja então abundante, nem sempre tal circunstância ocorre durante sucessivas estações do ano”.*

A partir do que foi exposto é possível associar a luta pela sobrevivência com os aspectos pertinentes à Teoria dos Jogos que, de acordo com Borges (1996), lida com situações de conflito de interesses, onde dois ou mais agentes racionais disputam entre si algum recurso limitado ou escasso presente no ambiente.

Os agentes racionais, segundo Rapoport (1998), são indivíduos que agem racionalmente, ou seja, levam em consideração todas as possíveis consequências de seus atos e, a partir disto, estabelecem certo grau de preferência, baseado em seus próprios atos e na ação que gerou o seu melhor resultado. Isto é, ele age de forma a maximizar seus ganhos (Rapoport, 1999). Em alguns casos, o resultado não depende única e exclusivamente da escolha feita por um único indivíduo mas, sim, da ação escolhida pelos outros, sobre os quais o primeiro não tem qualquer controle.

O estudo da Teoria dos Jogos foi matematicamente formalizado através dos artigos de John Von Neumann, publicados em 1928 e 1937. Entretanto, o livro *Theory of Games and Economic Behavior*, publicado com Oskar Morgenstern em 1944, é normalmente referenciado como a primeira abordagem completa e sistemática sobre o tema (von Neumann & Morgenstern, 1944).

### 3.1 Componentes de um jogo estratégico

Segundo Luce & Raiffa (1989), um jogo pode ser caracterizado como um ambiente formal que representa uma situação de conflito de interesses, com regras bem estabelecidas, onde agentes racionais se comportam estrategicamente buscando obter os maiores ganhos possíveis disponibilizados neste ambiente, através de uma função ou tabela de pagamentos. A seguir, cada um de seus componentes é brevemente apresentado.

- **Jogo:** é um modelo formal, o que significa que a Teoria dos Jogos envolve descrições e análises técnicas. É importante notar que os únicos tipos de jogos tratados por ela são os do tipo estratégicos;
- **Interações:** são as ações individuais de cada um dos jogadores que afetam os outros jogadores;
- **Jogadores:** qualquer indivíduo ou grupo de indivíduos que tenha capacidade de tomar decisões que afetem os demais;
- **Racionalidade:** assumindo que todos os jogadores são racionais, significa dizer que eles utilizam o método mais adequado para satisfazer os seus desejos, ou seja, maximizar seus ganhos;
- **Comportamento Estratégico:** cada jogador leva em consideração o fato de que todos eles interagem entre si. Desta maneira, a decisão tomada por um deles gera consequências aos outros jogadores. Existem dois tipos de estratégias de comportamento, que são: as estratégias puras, onde os jogadores sempre agem através do mesmo comportamento (por exemplo, sempre cooperar); e as estratégias mistas, onde os jogadores ora traem, ora cooperam, conforme a sua conveniência.

### 3.2 O Paradigma do dilema do prisioneiro (DP)

O dilema do prisioneiro foi apresentado pela primeira vez em 1950 por Melvin Dresher e Merrill Flood, cientistas da *RAND Corporation* e é considerado o mais famoso e mais conhecido jogo de duas pessoas de soma-não-zero não-cooperativo da Teoria dos Jogos. Ele define uma situação de conflito de interesses, onde dois indivíduos são presos e colocados em celas diferentes e sem comunicação entre eles (Poundstone, 1993).

Alan Tucker originalmente foi quem apresentou o DP neste sentido e foi então proposto a cada preso que: (a) se um deles confessasse o crime e o outro não, o que tivesse confessado seria preso por três meses por sua cooperação e o outro ficaria preso por dois anos; (b) se ambos confessassem o crime, então a cooperação individual perderia a força e ambos ficariam presos por um ano; (c) senão, caso nenhum deles confessasse o crime, eles seriam presos por apenas seis meses.

Tabela 1. Tabela de pagamentos do Dilema do Prisioneiro.

		Jogador 2	
		Cooperar (C)	Trair (D)
Jogador 1	Cooperar (C)	( R, R )	( S, T )
	Trair (D)	( T, S )	( P, P )

Na Tabela 1 é possível visualizar os possíveis pagamentos para o dilema do prisioneiro, que combina todos os pares de estratégias correlacionando os valores de pagamentos com cada um dos jogadores.

Cada jogador pode expressar dois comportamentos diferentes, que são: cooperar (C: *cooperate*) e trair (D: *defect*). Além disto, para constituir um Dilema do Prisioneiro, os valores  $T$ ,  $R$ ,  $P$  e  $S$  nas células da tabela devem obedecer às seguintes relações:

$$T > R > P > S \quad (1)$$

$$R > \frac{T + S}{2} \quad (2)$$

$$\frac{T + S}{2} > S \quad (3)$$

Para cada par de comportamentos possível, o pagamento de linha-coluna está listado na célula apropriada da tabela, onde cada valor significa:  $T$  é a tentação que cada jogador tem, caso traia sozinho;  $R$  é a recompensa que cada jogador recebe se ambos os jogadores cooperarem;  $P$  é a punição que cada um dos jogadores recebe quando ambos traem; e,  $S$  é o “pagamento do otário”, que coopera sozinho.

Além disto, existem variações para este jogo, onde é possível flexibilizar as relações existentes entre os quatro termos ( $T$ ,  $R$ ,  $P$  e  $S$ ), da relação na Equação 1, de forma a gerar diferentes configurações de jogos. Também, pode-se variar no que se refere à quantidade de iterações entre os jogos, pois na forma clássica do DP os jogadores disputam em apenas um único encontro.

As estratégias de comportamento dos jogadores, no caso do Dilema do Prisioneiro, podem ser de dois tipos: (a) puras, onde o jogador sempre coopera (*ALL-C*) ou sempre trai (*ALL-D*); e, mistas, onde o jogador pode combinar cooperar e trair, de acordo com aquilo que no momento lhe atribua um maior ganho. Neste caso é possível citar a estratégia *Aleatório*, onde ora o jogador coopera, ora trai, sem seguir um padrão pré-definido; e, também, a estratégia *TFT* (*Tit-For-Tat*), onde o jogador primeiramente coopera e, a partir do segundo encontro passa a agir com o mesmo comportamento que o seu adversário teve no encontro anterior ([Rapoport, 1999](#)).

### 3.3 Considerações

Esta seção apresentou conceitos pertinentes ao contexto da Teoria dos Jogos, mais especificamente em relação aos aspectos do Paradigma do Dilema do Prisioneiro, que constitui o segundo pilar fundamental para a construção da estrutura do SIGA, que é detalhado na próxima seção.

## 4. O Algoritmo Genético com Interação Social (SIGA)

O SIGA (*Social Interaction Genetic Algorithm*) é baseado nos fundamentos apresentados nas Seções 2 e 3 e justificado pelos motivos expressos na Seção 1. Nele, um indivíduo pode ter o seu valor de *fitness* alterado durante o processo evolutivo, ou seja, no decorrer da execução do AG. Com esta possibilidade, os indivíduos podem agora aumentar as suas chances de sobreviver e gerar descendentes, através da disputa de jogos com o objetivo de maximizar os seus ganhos individuais.

### 4.1 Estrutura básica

Para que isto ocorra, foi inserida uma nova etapa na estrutura do algoritmo apresentado na Figura 1, sendo que esta é anterior à fase de reprodução (seleção, cruzamento e mutação). Nela, os indivíduos são expostos a um ambiente, que nada mais é do que um jogo estratégico, onde eles têm a oportunidade de lutar pela sua existência durante algum período de tempo. Na Figura 2 é apresentada a estrutura completa do algoritmo.

Esta nova fase é chamada de Interação Social, conforme destacado na Figura 2, e é composta basicamente de três passos, que são: (a) selecionar aleatoriamente dois indivíduos; (b) obter o comportamento de cada um dos indivíduos, a partir de suas estratégias de comportamento; (c) alterar as suas adaptabilidades (valores de *fitness*) com base no comportamento adotado e na tabela de pagamento do jogo.

Além disto, esta estrutura permite o uso de qualquer tipo de jogo, de acordo com sua tabela de pagamentos e, ainda, quaisquer métodos de seleção, tais como: roleta ou torneio. Desta forma, na Seção 4.2 são apresentados detalhes de implementação do SIGA voltados especificamente para os problemas de otimização com restrições já mencionados.

### 4.2 Processo de implementação

Com base na estrutura apresentada na Figura 2, os aspectos do SIGA que se diferenciam da implementação clássica dos AGs são apresentados nesta Seção.

#### 4.2.1 Indivíduos e estratégia de comportamento

Na abordagem clássica dos AGs, os indivíduos são representados por apenas um cromossomo, que contém informações referentes ao problema em questão. No caso do SIGA, há a necessidade de se atribuir aos indivíduos

```

1. Início;
2. Gerar população inicial;
3. Avaliar cada indivíduo da população inicial;
4. Repetir até  $m$  gerações
  // Etapa de Interação Social
  4.1. Repetir até  $d$  disputas
    4.1.1. Selecionar aleatoriamente dois indivíduos;
    4.1.2. Repetir até  $r$  rodadas
      a. Obter o comportamento de cada um dos indivíduos, a partir de
      suas estratégias de comportamento
      b. Alterar as suas adaptabilidades, com base nos seus
      comportamentos e na tabela de pagamentos do jogo
  // Etapa de reprodução
  4.2. Repetir até o número de descendentes ser igual à quantidade desejada
  (% de cruzamento)
    4.2.1. Selecionar dois indivíduos conforme o método de seleção;
    4.2.2. Realizar a operação de cruzamento nos indivíduos selecionados
    no passo anterior;
    4.2.3. Realizar a operação de mutação nos descendentes gerados no
    passo anterior;
    4.2.4. Avaliar os descendentes;
  4.3. Substituir os indivíduos da população pelos novos indivíduos gerados
  na etapa de reprodução;
5. O melhor indivíduo da população é a solução do problema;
6. Fim.

```

Figura 2. Estrutura do algoritmo SIGA.

a capacidade de se comportarem estrategicamente. Assim, cada um deles passa a contar com mais um cromossomo, que é responsável pelo código genético referente a sua estratégia. Esta é gerada aleatoriamente na etapa de geração da população inicial, e é transmitida para os descendentes, onde cada um dos filhos recebe diretamente a estratégia de cada um dos pais e, posteriormente, esta sofre mutação em um dos dois de seus genes.

Desta forma, as estratégias de comportamento foram codificadas geneticamente, conforme mostrado na Tabela 2, através de um cromossomo de duas posições e um alfabeto composto por um sistema ternário, contendo os valores 0, 1 e 2, suficientes para codificar todas as quatro estratégias, que são: *ALL-D*, *ALL-C*, *TFT* e *Aleatório*, com uma distribuição igual a 3:3:2:1, respectivamente, na população inicial. Esta distribuição pode ser observada na quantidade de codificações genéticas, presentes na Tabela 2, para cada uma das estratégias. A notação genética para o gene é representada pela letra C ou c (comportamento) e a relação de dominância entre os alelos se dá da seguinte forma  $C^h = C^d > c$ .

#### 4.2.2 Cálculo do valor de *fitness*

O cálculo de *fitness*, no contexto da teoria clássica dos Algoritmos Genéticos, é baseado no genótipo da solução de um indivíduo e, a partir do

Tabela 2. Codificação genética das estratégias de comportamento.

Genótipo	Cromossomo	Fenótipo	Observação
$C^h C^h$	00	ALL-D	Conhece apenas comportamento trair.
$C^h C^d$	01	TFT	Conhece os dois comportamentos e os aplica de acordo com a última jogada do adversário, mas prioriza a cooperação.
$C^h c$	02	ALL-D	Conhece apenas comportamento trair.
$C^d C^h$	10	TFT	Conhece os dois comportamentos e os aplica de acordo com a última jogada do adversário, mas prioriza a cooperação.
$C^d C^d$	11	ALL-C	Conhece apenas comportamento cooperar.
$C^d c$	12	ALL-C	Conhece apenas comportamento cooperar.
$C^h c$	20	ALL-D	Conhece apenas comportamento trair.
$C^d c$	21	ALL-C	Conhece apenas comportamento cooperar.
$cc$	22	Aleatório	

Fonte: Brito et al. (2005)

momento que ele passa a contar com mais um cromossomo, como no caso do SIGA, há a necessidade de se alterar a maneira de calcular o valor de *fitness* de um indivíduo, denominado aqui de *fitness<sub>Total</sub>*.

Desta forma, além de possuírem um valor de *fitness* relacionado à solução do problema, agora passam a ter um valor de *fitness* baseado na soma dos ganhos obtidos na etapa de interação social, conforme a Equação 4.

$$fitness_{Total} = \alpha \times fitness_{Solucao} + \beta \times fitness_{Estrategia} \quad (4)$$

Assim, o *fitness* total de um indivíduo é calculado a partir da soma ponderada dos valores de *fitness* da solução do problema (*fitness<sub>Solucao</sub>*) e dos ganhos obtidos nas disputas dos jogos, através da estratégia de comportamento (*fitness<sub>Estrategia</sub>*).

É importante mencionar que, para a categoria de problemas aqui utilizada, problemas de otimização global com restrições, além do cálculo de *fitness*, a quantidade de violações de restrições de cada um dos indivíduos é levada em consideração para o processo de seleção. Desta forma, há a necessidade de se minimizar prioritariamente as violações e, posteriormente, o valor de *fitness<sub>Total</sub>*. Além disto, a abordagem do SIGA não leva em consideração a aplicação de penalidade ao valor de *fitness*, em decorrência da quantidade de restrições violadas, comumente encontrada em outras abordagens disponíveis na literatura, tais como em Coello Coello (2002).

### 4.3 Simulação da etapa de interação social

Como forma de exemplificar o funcionamento da etapa de interação social, é apresentada uma simulação na Figura 3. A população é composta por quatro indivíduos (id, valor de *fitness* e estratégia de comportamento), os valores  $T$ ,  $R$ ,  $P$  e  $S$  são, respectivamente, 30, 25, 15 e 10 e, a cada iteração são realizadas quatro rodadas de disputas do jogo Dilema do Prisioneiro Iterativo.

É interessante observar que na primeira iteração, por exemplo, os indivíduos 2 e 3 foram selecionados. Em cada rodada da disputa, o indivíduo 2 sempre trai enquanto que o indivíduo 3 coopera na primeira rodada e, posteriormente, age da mesma forma que o seu adversário na rodada anterior, ou seja, passa também a trair somente. A cada rodada lhes são atribuídos valores de ganho e, conseqüentemente, esta alteração influencia no resultado da operação de seleção por torneio.

Além disto, é importante frisar que primeiramente é executada a etapa de interação social, para somente depois ser realizada a operação de seleção por torneio. Este método de seleção foi utilizado para a realização dos testes. Desta forma, são selecionados dois indivíduos aleatoriamente e verificados os seus valores de *fitness* e a quantidade de violações de restrições. Então, tem-se as seguintes situações:

1. caso ambos não violem nenhuma restrição, então o de menor *fitness* ganha o torneio;
2. caso um não tenha violações e o outro tenha, então o indivíduo sem violações vence;
3. se ambos violarem restrições, vence aquele com a menor quantidade de violações;
4. caso violem a mesma quantidade de restrições, então o de menor *fitness* ganha.

Na parte inferior da Figura 3, é possível visualizar a mudança da configuração no gráfico de vencedores de cada uma das disputas, através dos resultados dos torneios parciais. Cada indivíduo é representado por uma textura diferente. É possível observar a influência da etapa de interação social, que permite aos indivíduos menos aptos evoluírem e passarem a ser selecionados para cruzamento, como é o caso do indivíduo 1 que era menos apto do que o indivíduo 4 e passou a ser mais apto.

### 4.4 Considerações

Esta Seção apresentou o Algoritmo Genético com Interação Social (SIGA), onde foi incluída a etapa de interação social antes da reprodução dos indivíduos. Assim, foram descritos os aspectos que diferem dos AGs tradicionais, no sentido de definir um indivíduo composto por dois cromossomos, um para a solução do problema e o outro para codificar geneticamente a

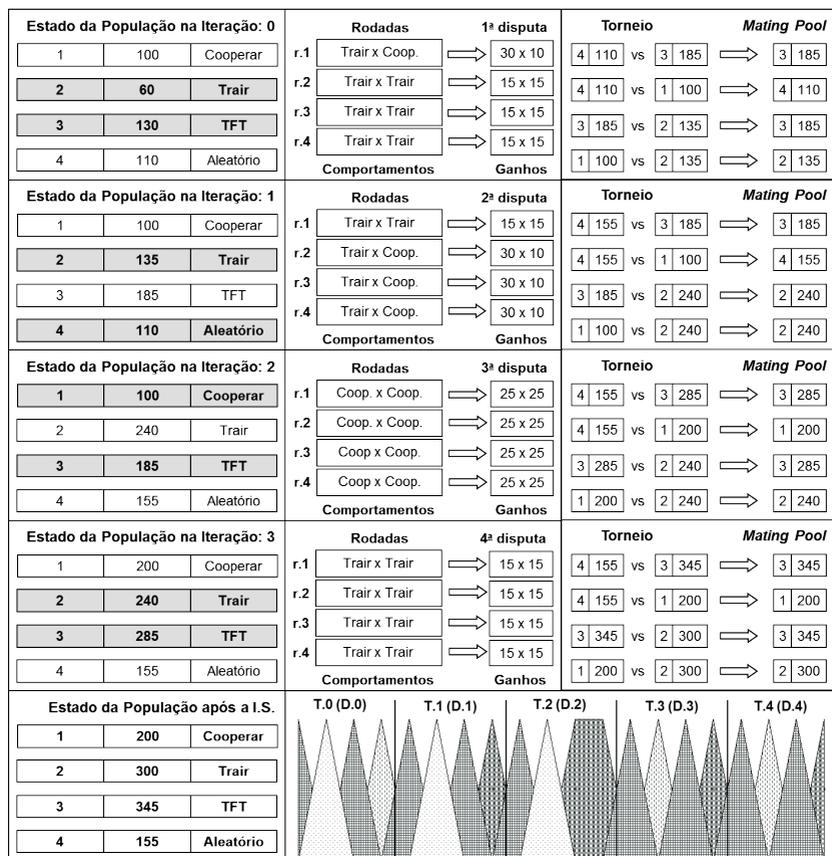


Figura 3. Exemplo de execução da etapa de interação social do SIGA.

sua estratégia de comportamento. Na sequência, foi definido o novo cálculo de *fitness*, que passou a contar com os valores ganhos nas disputas dos jogos. Por fim, foi exemplificada a execução da etapa de interação social e demonstrada a sua influência no processo de seleção dos indivíduos.

## 5. Avaliação do Algoritmo SIGA

Como forma de avaliar o Algoritmo Genético com Interação Social foram utilizados quatro problemas de otimização global com restrições, que são amplamente utilizados para avaliação de meta-heurísticas.

### 5.1 Problema 1: projeto de viga de aço

Segundo [Rao \(1996\)](#), este problema objetiva minimizar o custo de fabricação de uma viga de aço, que está sujeita a algumas restrições, tais como: tensão do cisalhamento, esforços de flexão na viga, flambagem de carga na barra e, deflexão do feixe de extremidade e restrições laterais. Além disto, são definidas quatro variáveis: espessura da solda ( $h$ ); largura do feixe ( $t$ ); espessura da viga ( $b$ ); comprimento da junta soldada ( $l$ ), conforme mostrado na Figura 4.

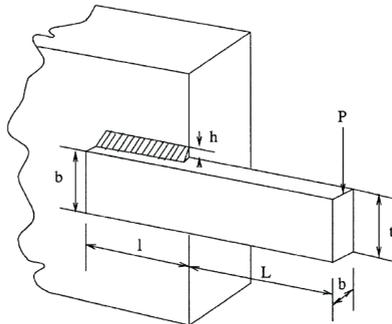


Figura 4. Projeto de viga de aço.  
Fonte: [Alfares & Esat \(2006\)](#).

Este problema pode ser formalizado da seguinte forma:

$$\text{Minimizar } f(X) = 1,10471x_1^2x_2 + 0,04811x_3x_4(14,0 + x_2) \quad (5)$$

Sujeito a:

$$g_1 = \tau - \tau_{max} \leq 0 \quad (6)$$

$$g_2 = \sigma - \sigma_{max} \leq 0 \quad (7)$$

$$g_3 = h - b \leq 0 \quad (8)$$

$$g_4 = 0,10471h^2 + 0,04811tb(14,0 + l) - 5,0 \leq 0 \quad (9)$$

$$g_5 = 0,125 - h \leq 0 \quad (10)$$

$$g_6 = \delta - \delta_{max} \leq 0 \quad (11)$$

$$g_7 = P - P_c \leq 0 \quad (12)$$

Onde:

$$\tau = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{1}{2R} + (\tau'')^2} \quad (13)$$

$$\tau' = \frac{P}{\sqrt{2hl}} \quad (14)$$

$$\tau'' = \frac{MR}{J} \quad (15)$$

$$M = P \left( L + \frac{1}{2} \right) \quad (16)$$

$$R = \sqrt{\frac{t^2}{4} + \left( \frac{h+t}{2} \right)^2} \quad (17)$$

$$J = 2 \left\{ \sqrt{2hl} \left[ \frac{t^2}{12} + \left( \frac{h+t}{2} \right)^2 \right] \right\} \quad (18)$$

$$\sigma = \frac{6PL}{bt^2} \quad (19)$$

$$\delta = \frac{4PL^3}{Eb^3} \quad (20)$$

$$P_c = \frac{4,013E\sqrt{\frac{t^2b^6}{36}}}{L^2} + \left( 1 - \frac{t}{2L} \sqrt{\frac{E}{4G}} \right) \quad (21)$$

Além disto, os seguintes valores foram considerados:  $P = 6000$  lb,  $L = 14$  in,  $E = 30 \times 10^6$  psi,  $G = 12 \times 10^6$  psi,  $\tau_{max} = 13600$  psi,  $\sigma_{max} = 30000$  psi,  $\delta_{max} = 0,25$  in e os intervalos de valores para cada uma das quatro variáveis:  $0,1 \leq h \leq 2,0$ ;  $0,1 \leq l \leq 10,0$ ;  $0,1 \leq t \leq 10,0$ ;  $0,1 \leq b \leq 2,0$ .

O SIGA foi executado 30 vezes para diferentes configurações de parâmetros e, assim, ficou estabelecido o conjunto que apresentou os melhores

Tabela 3. Resultados comparativos para o problema de projeto de viga de aço.

Variáveis	SIGA	He & Wang (2007)	Coello Coello & Montes (2002)	Coello Coello (2000)	Deb (1991)
$x_1(h)$	0,171937	0,202369	0,205986	0,208800	0,248900
$x_2(l)$	4,122129	3,544214	3,471328	3,420500	6,173000
$x_3(t)$	9,587429	9,048210	9,020224	8,997500	8,178900
$x_4(b)$	0,183010	0,205723	0,206480	0,210000	0,253300
$g_1(x)$	-8,067400	-12,839796	-0,074092	-0,337812	-5758,603777
$g_2(x)$	-39,336800	-1,247467	-0,266227	-353,902604	-255,576901
$g_3(x)$	-0,011070	-0,001498	-0,000495	-0,001200	-0,004400
$g_4(x)$	-3,467150	-3,429347	-3,430043	-3,411865	-2,982866
$g_5(x)$	-0,236390	-0,079381	-0,080986	-0,083800	-0,123900
$g_6(x)$	-16,024300	-0,235536	-0,235514	-0,235649	-0,234160
$g_7(x)$	-0,046940	-11,681355	-58,666440	-363,232384	-4465,270928
$f(x)$	1,664373	1,728024	1,728226	1,748309	2,433116

resultados: método de seleção = torneio; tamanho do torneio = 2; quantidade de gerações = 2000; tamanho da população = 200; taxa de cruzamento = 85%; taxa de mutação = 10%; alfa = 1,0; beta = 1,0; jogos = 100; rodadas = 10; R = 5; T = 3; P = 1; S = 0.

O melhor resultado obtido dentre as 30 execuções realizadas foi igual a 1,664373, que é o melhor dentre os resultados disponíveis na literatura apresentados na Tabela 3.

## 5.2 Problema 2: projeto de vaso de pressão

Este problema tem por objetivo minimizar o custo total, incluindo o custo do material, modelagem e soldagem, de um recipiente cilíndrico, que é limitado em suas extremidades por cabeças hemisféricas. Ele é composto por quatro variáveis de projeto, que são: espessura da casca ( $T_s$ ,  $x_1$ ), espessura da cabeça ( $T_h$ ,  $x_2$ ), raio interno ( $R$ ,  $x_3$ ) e comprimento da seção cilíndrica do recipiente ( $L$ ,  $x_4$ ), não incluindo a cabeça, conforme mostrado na Figura 5. É importante observar que os valores das variáveis  $T_s$  e  $T_h$  são múltiplos de 0,0625 polegadas, pois referem-se às espessuras de chapas de aço laminadas padronizadas, e  $R$  e  $L$  são variáveis contínuas.

Este problema pode ser formulado da seguinte forma:

$$\text{Minimizar } f(X) = 0,6224x_1x_3x_4 + 1,7781x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3 \quad (22)$$

Sujeito a:

$$g_1 = -x_1 + 0,0193x_3 \leq 0 \quad (23)$$

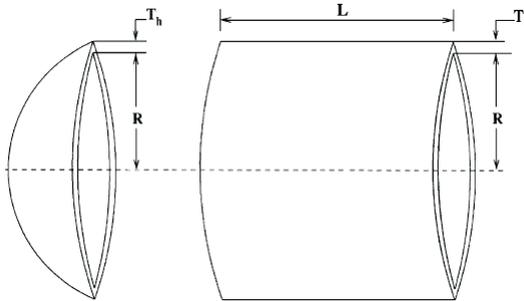


Figura 5. Projeto de vaso de pressão.

Fonte: [He & Wang \(2007\)](#).

$$g_2 = -x_2 + 0,00954x_3 \leq 0 \quad (24)$$

$$g_3 = -\pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 + 1296000 \leq 0 \quad (25)$$

$$g_4 = x_4 - 240 \leq 0 \quad (26)$$

Além disto, os seguintes intervalos de valores foram utilizados para cada uma das quatro variáveis:  $1 \times 0,0625 \leq x_1 \leq 99 \times 0,0625$ ;  $1 \times 0,0625 \leq x_2 \leq 99 \times 0,0625$ ;  $10 \leq x_3 \leq 200$ ;  $10 \leq x_4 \leq 200$ .

O SIGA foi executado 50 vezes para diferentes configurações de parâmetros e ficou estabelecido o conjunto que apresentou os melhores resultados: método de seleção = torneio; tamanho do torneio = 2; quantidade de gerações = 2000; tamanho da população = 200; taxa de cruzamento = 85%; taxa de mutação = 20%; alfa = 1,0; beta = 1,0; jogos = 100; rodadas = 10; R = 5; T = 3; P = 1; S = 0.

O melhor resultado obtido dentre as 30 execuções realizadas foi igual a 6066,029360, que é o terceiro melhor dentre os resultados disponíveis na literatura e apresentados na Tabela 4.

### 5.3 Problema 3: minimização do peso da tensão/compressão sobre mola

Este problema tem por objetivo minimizar o peso da tensão/compressão sobre uma mola, que está sujeita a algumas restrições, tais como: deflexão mínima, tensão do cisalhamento, a frequência de onda, limites de diâmetro externo e variáveis de projeto, conforme mostrado em [Arora \(2004\)](#); [Belegundu & Arora \(1985\)](#). As variáveis de projeto são: o diâmetro do fio ( $d$ ,  $x_1$ ), a bobina de diâmetro médio ( $D$ ,  $x_2$ ) e o número de bobinas ativas ( $P$ ,  $x_3$ ), conforme mostrado na Figura 6.

Tabela 4. Comparação dos resultados para o problema de projeto de vaso de pressão.

Variáveis	SIGA	He & Wang (2007)	Coello Coello & Montes (2002)	Deb (1997)	Kannan & Kramer (1994)
$x_1(Ts)$	0,812500	0,812500	0,812500	0,937500	1,125000
$x_2(Th)$	0,437500	0,437500	0,437500	0,500000	0,625000
$x_3(R)$	42,092732	42,091266	42,097398	48,329000	58,291000
$x_4(L)$	176,947780	176,746500	176,654050	112,679000	43,690000
$g_1(x)$	-0,000110	-0,000139	-0,000020	-0,004750	0,000016
$g_2(x)$	-0,035935	-0,035949	-0,035891	-0,038941	-0,068904
$g_3(x)$	-1337,994634	-116,382700	-27,886075	-3652,876838	-21,220104
$g_4(x)$	-63,052220	-63,253500	-63,345953	-127,321000	-196,310000
$f(x)$	6066,029360	6061,077700	6059,946300	6410,381100	7198,042800

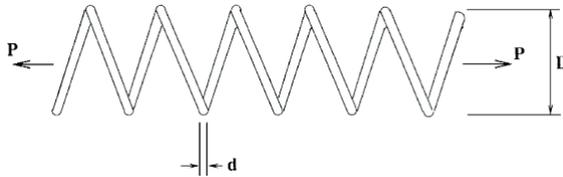


Figura 6. Minimização do peso da tensão/compressão sobre mola.  
 Fonte: He & Wang (2007).

Este problema pode ser formalizado pelas Equações 27 a 31 seguintes:

$$\text{Minimizar } f(X) = (x_3 + 2)x_2x_1^2 \tag{27}$$

Sujeito a:

$$g_1 = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \tag{28}$$

$$g_2 = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \tag{29}$$

$$g_3 = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \tag{30}$$

$$g_4 = \frac{x_1 - x_2}{1.5} - 1 \leq 0 \tag{31}$$

O SIGA foi executado 20 vezes para diferentes configurações de parâmetros e ficou estabelecido o conjunto que apresentou os melhores resultados: método de seleção = torneio; tamanho do torneio = 2; quantidade de gerações = 2000; tamanho da população = 300; taxa de cruzamento = 85%; taxa de mutação = 10%; alfa = 1,0; beta = 1,0; jogos = 100; rodadas = 10; R = 5; T = 3; P = 1; S = 0.

Tabela 5. Comparação dos resultados para o problema de minimização do peso da tensão/compressão sobre mola.

Variáveis	SIGA	He & Wang (2007)	Coello Coello & Montes (2002)	Arora (2004)	Belegundu & Arora (1985)
$x_1(d)$	0,050180	0,051728	0,051989	0,053396	0,050000
$x_2(D)$	0,279604	0,357644	0,363965	0,399180	0,315900
$x_3(P)$	2,087959	11,244543	10,890522	9,185400	14,250000
$g_1(x)$	-0,002840	-0,000845	-0,000013	0,000019	-0,000014
$g_2(x)$	-0,249450	-0,0000126	-0,000021	-0,000018	-0,003782
$g_3(x)$	-42,176000	-4,051300	-4,061338	-4,123832	-3,938302
$g_4(x)$	-0,780140	-0,727090	-0,722698	-0,698283	-0,756067
$f(x)$	0,002878	0,0126747	0,0126810	0,0127303	0,0128334

O melhor resultado obtido dentre as 20 execuções realizadas para essa configuração está apresentado na Tabela 5, onde é possível compará-lo com os resultados disponíveis na literatura.

Em um primeiro momento, o resultado obtido pelo SIGA foi surpreendente, por ser 77,29% melhor do que o melhor resultado dentre os outros quatro resultados. No entanto, ele foi validado conforme a função objetivo, sem que houvesse a violação de nenhuma das restrições.

#### 5.4 Problema 4: projeto de redutor de velocidade

Na Figura 7 é possível visualizar o projeto de um redutor de velocidade, onde o seu peso deve ser minimizado, segundo algumas restrições: flexão de estresse dos dentes da engrenagem, tensão superficial, desvios transversais das hastes e as tensões no eixo. Neste problema são consideradas sete variáveis: a largura do rosto ( $x_1$ ), o módulo de dentes ( $x_2$ ), o número de dentes do pinhão ( $x_3$ ), o comprimento do primeiro eixo entre os rolamentos ( $x_4$ ), o comprimento do segundo eixo entre os rolamentos ( $x_5$ ), o diâmetro do primeiro eixo ( $x_6$ ) e o diâmetro do segundo eixo ( $x_7$ ).

Este problema pode ser formalizado da seguinte forma:

$$\begin{aligned} \text{Minimizar } f(X) = & 0,7854x_1x_2^2(3,3333x_3^2 + 14,9334x_3 - 43,0934) \quad (32) \\ & -1,508x_1(x_6^2 + x_7^2) + 7,4777(x_6^2 + x_7^2) \\ & +0,78054(x_4x_6^2 + x_5x_7^2) \end{aligned}$$

Sujeito a:

$$g_1 = \frac{27}{x_1x_2^2x_3} - 1 \leq 0 \quad (33)$$

$$g_2 = \frac{397,5}{x_1x_2^2x_3} - 1 \leq 0 \quad (34)$$

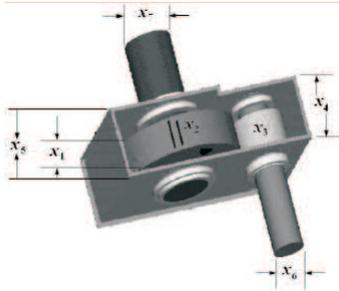


Figura 7. Projeto de redutor de velocidade.  
Fonte: Brajevic et al. (2010).

$$g_3 = \frac{1,93x_4^3}{x_2x_3x_6^3} - 1 \leq 0 \quad (35)$$

$$g_4 = \frac{1,93x_5^3}{x_2x_3x_7^3} - 1 \leq 0 \quad (36)$$

$$g_5 = \frac{1,0}{110x_6^3} \sqrt{\left(\frac{750,0x_4}{x_2x_3}\right)^2 + 16,9 \times 10^6} - 1 \leq 0 \quad (37)$$

$$g_6 = \frac{1,0}{85x_7^3} \sqrt{\left(\frac{750,0x_5}{x_2x_3}\right)^2 + 157,5 \times 10^6} - 1 \leq 0 \quad (38)$$

$$g_7 = \frac{x_2x_3}{40} - 1 \leq 0 \quad (39)$$

$$g_8 = \frac{5x_2}{x_1} - 1 \leq 0 \quad (40)$$

$$g_9 = \frac{x_1}{12x_2} - 1 \leq 0 \quad (41)$$

$$g_{10} = \frac{1,5x_6 + 1,9}{x_4} - 1 \leq 0 \quad (42)$$

$$g_{11} = \frac{1,1x_7 + 1,9}{x_5} - 1 \leq 0 \quad (43)$$

Além disto, os seguintes intervalos de valores foram utilizados para cada uma das sete variáveis:  $2,6 \leq x_1 \leq 3,6$ ;  $0,7 \leq x_2 \leq 0,8$ ;  $17 \leq x_3 \leq 28$ ;  $7,3 \leq x_4 \leq 8,3$ ;  $7,3 \leq x_5 \leq 8,3$ ;  $2,9 \leq x_6 \leq 5,0$ ;  $7,3 \leq x_7 \leq 5,5$ .

O SIGA foi executado 50 vezes para diferentes configurações de parâmetros e ficou estabelecido o conjunto que apresentou os melhores resultados:

Tabela 6. Comparação dos resultados para o problema de projeto de redutor de velocidade.

Variáveis	SIGA	Brajevic et al. (2010)	Cagnina et al. (2008)
$x_1$	3,500459	3,500000	3,500000
$x_2$	0,700020	0,700000	0,700000
$x_3$	17,005030	17,000000	17,000000
$x_4$	7,300251	7,300000	7,300000
$x_5$	7,800195	7,800000	7,800000
$x_6$	2,900041	3,350215	3,350214
$x_7$	5,286863	5,286683	5,286683
$g_1(x)$	-0,074364	-0,073915	-0,073915
$g_2(x)$	-0,198624	-0,197996	-0,197998
$g_3(x)$	-0,108202	-0,499172	-0,499172
$g_4(x)$	-0,901443	-0,901471	-0,901471
$g_5(x)$	-1,000000	-2,220E-16	0,000000
$g_6(x)$	-0,000102	-3,331E-16	-5,000E-16
$g_7(x)$	-0,702403	-0,702500	-0,702500
$g_8(x)$	-0,000103	0,000000	-1,000E-16
$g_9(x)$	-0,795801	-0,583333	-0,583333
$g_{10}(x)$	-0,143857	-0,051326	-0,051325
$g_{11}(x)$	-0,011074	-0,010852	-0,010852
$f(x)$	2897,531422	2996,348165	2996,348165

método de seleção = torneio; tamanho do torneio = 3; quantidade de gerações = 2000; tamanho da população = 200; taxa de cruzamento = 75%; taxa de mutação = 5%; alfa = 1,0; beta = 1,0; jogos = 100; rodadas = 10; R = 5; T = 3; P = 1; S = 0.

O melhor resultado obtido dentre as 50 execuções realizadas para esta configuração está apresentado na Tabela 6, onde é possível compará-lo com os resultados disponíveis na literatura.

O resultado obtido pelo SIGA para este problema é bastante significativo, pois o valor 2996,348165 é considerado o ótimo global nas referências bibliográficas pesquisadas e utilizadas para comparação. No entanto, foi encontrado um valor melhor e que foi validado a partir do modelo matemático do problema, sem a ocorrência de violação em nenhuma das restrições.

## 6. Considerações Finais

Este capítulo apresentou uma nova meta-heurística híbrida de otimização baseada na natureza, denominada de Algoritmo Genético com Interação Social (SIGA), aplicada na resolução de quatro problemas de engenharia bastante difundidos na literatura.

Para a sua concepção foram observados aspectos presentes, por exemplo, na dinâmica das interações sociais entre seres humanos. No entanto, a abordagem aqui apresentada é um modelo bastante simplificado do que

é encontrado na natureza. Desta forma, foram introduzidas novas características no Algoritmo Genético tradicional, no sentido de permitir que os indivíduos da população possam evoluir ao longo do processo evolutivo, conforme os ganhos obtidos na etapa de Interação Social.

Assim, cada indivíduo passou a ser caracterizado por dois cromossomos, um para representar a possível solução do problema em questão e o outro para codificar geneticamente a sua estratégia de comportamento. Neste estudo foram utilizadas apenas quatro estratégias, sendo duas puras, *ALL-D* e *ALL-C*, e duas mistas, *Aleatório* e *TFT*.

Por conseguinte, o cálculo de *fitness* passou a contar não mais somente com o valor gerado a partir do cromossomo solução, aqui denominado de *fitness<sub>Solucao</sub>*, mas também com a soma dos ganhos obtidos nas disputas dos jogos, denominado de *fitness<sub>Estrategia</sub>*. Sendo assim, o *fitness<sub>Total</sub>* passou a ser o valor de *fitness* do indivíduo calculado através da soma ponderada dos *fitness<sub>Solucao</sub>* e *fitness<sub>Estrategia</sub>*, conforme expresso na Equação 4.

Em relação aos resultados dos experimentos realizados para os quatro problemas definidos, é possível perceber uma significativa melhora nos resultados dos problemas relatados nas Seções 5.1, 5.3 e 5.4 que são, respectivamente, projeto de viga de aço, minimização do peso da tensão/compressão sobre mola e o projeto de redutor de velocidade.

Para cada um deles foram encontrados novos ótimos globais, em comparação com os melhores resultados presentes em estudos similares, conforme mostrado em cada uma das tabelas comparativas dos resultados obtidos. Somente para o problema da Seção 5.2, projeto de vaso de pressão, que o ótimo global conhecido até o momento não foi superado, mas o resultado obtido ficou próximo. No entanto, acredita-se que o valor ótimo não foi encontrado pelo fato do SIGA não ter tido a capacidade de realizar uma efetiva busca pelo espaço de soluções candidatas, que dentre todos os problemas é o maior.

É importante frisar que estes resultados foram gerados até o presente e, apesar de melhores em três dos quatro problemas, eles apenas indicam a viabilidade do algoritmo proposto e motivam a realização de estudos posteriores, no sentido de explorar todas as suas características.

Além disto, em estudos anteriores, com a aplicação do SIGA em problemas de otimização combinatória (Lehrer, 2000; Teixeira, 2005; Brito et al., 2005; Teixeira et al., 2006b,a, 2010b,a), foi possível constatar um aumento da variabilidade populacional no decorrer do processo evolutivo e, neste sentido, há indícios que o mesmo tenha ocorrido nos problemas de otimização global com restrições aqui apresentados. Desta forma, há necessidade de realizar estudos analíticos para constatar esta hipótese. Consequentemente, há ainda a possibilidade de explorar outras configurações dos parâmetros na busca por resultados melhores, de forma a aprimorar a técnica e a sua aplicabilidade em diversas categorias de problemas.

## Agradecimentos

Este trabalho tem o suporte acadêmico do Laboratório de Computação Natural (LCN), do Centro Universitário do Estado do Pará (CESUPA), e do Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), do Instituto de Tecnologia (ITEC) da Universidade Federal do Pará (UFPA).

## Referências

- Alfares, F. & Esat, I., Real-coded quantum inspired evolution algorithm applied to engineering optimization problems. In: *Proceedings of the Second International Symposium on Leveraging Applications and Formal Methods, Verification and Validation*. Washington, USA: IEEE Computer Society, p. 169–176, 2006.
- Arora, J., *Introduction to Optimum Design*. 2a edição. New York, USA: Academic Press, 2004.
- Belegundu, A. & Arora, J., *A study of mathematical programming methods for structural optimization. Part I: Theory*. *International Journal for Numerical Methods in Engineering*, 21(9):1583–1599, 1985.
- Blum, C. & Roli, A., Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.
- Borges, P., *A Model of Strategy Games based on the Paradigm of the Iterated Prisoner's Dilemma Employing Fuzzy Sets*. Tese de doutorado, Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, SC, 1996.
- Brajevic, I.; Tuba, M. & Subotic, M., Improved artificial bee colony algorithm for constrained problems. In: *Proceedings of the 11<sup>th</sup> WSEAS International Conference on Neural Networks, Fuzzy Systems and Evolutionary Computing*. Stevens Point, USA: WSEAS, p. 185–190, 2010.
- Brito, F.; Teixeira, O. & Oliveira, R., A introdução da interação fenotípica em algoritmos genéticos através dos jogos evolucionários e da codificação e transmissão genética do comportamento. In: *Anais do VII Simpósio Brasileiro de Automação Inteligente*. 2005.
- Cagnina, L.; Esquivel, S. & Coello Coello, C., Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica*, 32(3):319 – 326, 2008.
- Coello Coello, C., Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2):113–127, 2000.
- Coello Coello, C., Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002.

- Coello Coello, C. & Montes, E., Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3):193–203, 2002.
- Darwin, C., *On the Origin of Species by Means Natural Selection, or preservation of favored races in the struggle for life*. London, UK: John Murray, 1859.
- Dawkins, R., *The Extended Phenotype: the long reach of the gene*. New York, USA: Oxford University Press, 1999.
- Deb, K., Optimal design of a welded beam via genetic algorithms. *AIAA Journal*, 29(11):2013–2015, 1991.
- Deb, K., Geneas: a robust optimal design technique for mechanical component design. In: Dasgupta, D. & Michalewicz, Z. (Eds.), *Evolutionary Algorithms in Engineering Applications*. Berlin, Germany: Springer-Verlag, p. 497–514, 1997.
- Eberhart, R.; Simpson, P. & Dobbins, R., *Computation Intelligence PC Tools*. San Diego, USA: Academic Press, 1996.
- Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, USA: Addison Wesley, 1989.
- Haupt, R. & Haupt, S., *Practical Genetic Algorithms*. New York, USA: John Wiley & Sons, 2000.
- He, Q. & Wang, L., An effective co-evolutionary particle swarm optimization for constrained engineering design problem. *Engineering Applications of Artificial Intelligence*, 20(1):89–99, 2007.
- Holland, J., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- Kannan, B. & Kramer, S., An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, 116(2):405–411, 1994.
- Lahoz-Beltra, R.; Ochoa, G. & Aickelin, U., Cheating for problem solving: a genetic algorithm with social interactions. In: *Proceedings of Genetic and Evolutionary Computation Conference*. p. 811–818, 2009.
- Lehrer, C., *Operador de Seleção para Algoritmos Genéticos baseado no Jogo Hawk-Dove*. Dissertação de mestrado, Programa de Pós-graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, SC, 2000.
- Luce, D. & Raiffa, H., *Games and Decisions: Introduction and Critical Survey*. Mineola, USA: Dover Publications, 1989.
- Mitchell, M., *An Introduction to Genetic Algorithms*. Cambridge, USA: MIT Press, 1998.

- Poundstone, W., *Prisoner's Dilemma: John Von Neumann, Game Theory, and the Puzzle of the Bomb*. Sydney, Australia: Anchor Books, 1993.
- Rao, S., *Engineering Optimization*. 3a edição. New York, USA: Wiley Interscience, 1996.
- Rapoport, A., *Lutas, Jogos e Debates*. 2a edição. Brasília, DF: Editora da UNB, 1998.
- Rapoport, A., *Two-Person Game Theory*. Mineola, USA: Dover Publications, 1999.
- Teixeira, O., *Proposta de um Novo Algoritmo Genético baseado na Teoria dos Jogos*. Dissertação de mestrado, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Pará, Belém, PA, 2005.
- Teixeira, O.; Brito, F.; Teixeira, A. & Oliveira, R., Genetic algorithms with social interaction phase as phenotype characterization. In: Press, X.U. (Ed.), *Proceedings of Advances in Natural Computation and Data Mining*. Xi'an, China, 2006a.
- Teixeira, O.; Lobato, W.; Yasojima, C.; Brito, F.; Teixeira, A. & Oliveira, R., Fuzzy social interaction genetic algorithm. In: *Proceedings of the 10<sup>th</sup> Annual Conference on Genetic and Evolutionary Computation*. p. 2113–2114, 2010a.
- Teixeira, O.; Lobato, W.; Yasojima, C.; Brito, F.; Teixeira, A. & Oliveira, R., A new hybrid nature-inspired metaheuristic for problem solving based on the social interaction genetic algorithm employing fuzzy systems. In: *Proceedings of the 10<sup>th</sup> International Conference on Hybrid Intelligent Systems*. Atlanta, USA, p. 31–36, 2010b.
- Teixeira, O.; Teixeira, A.; Brito, F. & Oliveira, R., Game theory as a new paradigm for phenotype characterization of genetic algorithms. In: *Proceedings of the 8<sup>th</sup> Annual Conference on Genetic and Evolutionary Computation*. New York, USA, p. 1431–1432, 2006b.
- von Neumann, J. & Morgenstern, O., *Theory of Games and the Economic Behavior*. Princeton, USA: Princeton University Press, 1944.

## Notas Biográficas

**Otávio Noura Teixeira** é graduado em Tecnologia em Processamento de Dados (1998), pelo Centro Universitário do Estado do Pará (CESUPA), graduado em Bacharelado em Ciência da Computação (2003), mestre em Engenharia Elétrica (2005) e doutorando em Engenharia Elétrica, pela Universidade Federal do Pará (UFPA). Atualmente é Professor em tempo integral, e com experiência de mais de 11 anos de docência, no Centro Universitário do Estado do Pará (CESUPA), onde é também coordenador e pesquisador do Laboratório de Computação Natural. Ele já orientou mais de 40 trabalhos de conclusão de curso e participou de 85 bancas de defesa, além de ter publicado 40 artigos em eventos e periódicos.

**Walter Avelino da Luz Lobato** é graduado em Bacharelado em Ciência da Computação (2009) pelo Centro Universitário do Estado do Pará (CESUPA). Atualmente é aluno do curso de especialização em Java Corporativo e membro do Laboratório de Computação Natural, pela mesma instituição em que se graduou. Também é mestrando em Ciência da Computação do Programa de Pós-graduação em Ciência da Computação (PPGCC) da Universidade Federal do Pará (UFPA).

**Hitoshi Seki Yanaguibashi** é graduando em Bacharelado em Ciência da Computação e é bolsista de iniciação científica do Laboratório de Computação Natural, do Centro Universitário do Estado do Pará (CESUPA).

**Rodrigo Vieira Cavalcante** é graduando em Bacharelado em Ciência da Computação, pelo Centro Universitário do Estado do Pará (CESUPA), e em Engenharia de Automação e Controle pelo Instituto de Estudos Superiores da Amazônia (IESAM).

**Deam James Azevedo da Silva** é graduado em Tecnologia em Processamento de Dados (1993), pelo Centro Universitário do Estado do Pará (CESUPA) e mestre em Informática (2002) pela Universidade Federal de Campina Grande (UFCG). É professor Assistente III da Universidade Federal do Oeste do Pará (UFOPA). Atualmente é doutorando do Programa de Pós-graduação em Engenharia Elétrica (PPGEE) da Universidade Federal do Pará (UFPA).

**Roberto Célio Limão de Oliveira** é mestre em Engenharia Eletrônica (1991) pelo Instituto Tecnológico de Aeronáutica, e doutor em Engenharia Elétrica (1999) pela Universidade Federal de Santa Catarina. Atualmente é professor Associado da Faculdade de Engenharia de Computação da Universidade Federal do Pará. Ele orientou 20 alunos de mestrado e doutorado nas áreas de Controle Inteligente e Inteligência Computacional, onde publicou cerca de 80 artigos em eventos e periódicos.



# Controlador Preditivo Neural do Nível do Molde do Lingotamento Contínuo Sintonizado Via Algoritmo Genético

Fábio B. Sanchotene, Gustavo M. de Almeida e José Leandro F. Salles\*

---

**Resumo:** O Controlador Preditivo Baseado em Modelo (MPC) tem sido cada vez mais aplicado nos processos industriais, pois geralmente exibe um bom desempenho e robustez, desde que os seus parâmetros sejam sintonizados corretamente. Este capítulo utiliza um Algoritmo Genético para realizar a sintonia dos parâmetros do MPC Neural aplicado no controle do nível do molde do lingotamento contínuo, já que não existe nenhum método analítico para resolver este problema de sintonia. O nível do molde é uma das malhas de controle mais importantes na siderurgia, pois variações no nível acima de 10 mm prejudicam a qualidade do aço produzido. A estabilidade e a robustez de dois tipos de algoritmos MPC neural são comparadas, quando o nível do molde é afetado pelos distúrbios de *clogging* e *bulging*.

**Palavras-chave:** Controle Preditivo, Algoritmo genético, Lingotamento contínuo, Siderurgia.

**Abstract:** *The Model Based Predictive Control (MPC) is increasingly applied in industry, because it usually exhibits a good performance and robustness, provided their parameters are correctly tuned. In this chapter, a Genetic Algorithm is used to tune the Neural MPC parameters, which will be applied in the control of the mold level in the continuous casting process, since there is no analytical method that can be used to solve this tuning problem. The mold level control is one of the most important control loop present in the steel industry, since variations in the level above 10 mm reduce the quality of steel. The stability and robustness of two Neural MPC algorithms are compared when the mold level is affected by clogging and bulging disturbs.*

**Keywords:** *Predictive controller, Genetic algorithm, Continuous casting, Steel Industry.*

---

\* Autor para contato: jleandro@ele.ufes.br

## 1. Introdução

O Controlador Preditivo baseado em Modelo (MPC) originou-se nos anos 70 e, desde então, tem sido bastante estudado no meio acadêmico e aplicado no controle de diversas plantas industriais (Qin & Badgwell, 2003). O termo MPC designa uma ampla variedade de métodos de controle que fazem uso explícito do modelo de um processo para obter o sinal de controle pela minimização de uma função objetivo (Camacho & Bordons, 2004). Esta metodologia é de natureza aberta, dentro da qual diferentes algoritmos de controle têm sido desenvolvidos, tais como: Controle Preditivo Generalizado (GPC) (Clarke et al., 1987), Controle por Matriz Dinâmica (DMC) (Cutler & Ramaker, 1979) e Controle Auto Adaptativo Estendido Preditivo (Keyser & van Cuawenberghe, 1985), entre outros. O Controlador Preditivo pode ser aplicado em processos lineares, não lineares, mono e multivariáveis, com restrições e atrasos, desde aqueles com dinâmica estável até os de fase não mínima e instáveis em malha aberta (Camacho & Bordons, 2004).

O MPC geralmente exibe um bom desempenho e robustez, desde que os parâmetros de sintonia sejam adequadamente escolhidos pelo projetista (Maciejowski, 2000). Entretanto, a seleção dos parâmetros não é uma tarefa fácil, na medida em que não existe um método geral e preciso de sintonia que assegure a estabilidade (Bitmead et al., 1990). O método mais usado na literatura para o ajuste dos parâmetros do MPC atualmente é o da tentativa e erro, exigindo, além de um bom conhecimento do processo por parte do operador, também um certo tempo até que a resposta obtida seja satisfatória. Em alguns processos instáveis em malha aberta, como o nível do molde a ser estudado neste capítulo e também em sistemas multivariáveis como o laminador apresentado por de Almeida et al. (2009b), não se consegue determinar os parâmetros de sintonia através de tentativa e erro, a fim de obter uma resposta estável e otimizada. Por isto, observa-se na literatura que é ainda pouco explorada a aplicação do MPC em sistemas instáveis em malha aberta, lineares ou não.

O objetivo é utilizar o MPC com modelo neural, a fim de realizar o controle do nível do molde do lingotamento contínuo, que é um processo instável em malha aberta e não linear (Sanhotene & Salles, 2009; Mattedi et al., 2010). Para este caso, o método de sintonia do MPC baseado em tentativa e erro é praticamente inviável de ser aplicado, sendo fundamental utilizar ferramentas computacionais baseadas em computação evolucionária para obter automaticamente os parâmetros do controlador. Exemplos de artigos que utilizam o Algoritmo Genético para sintonizar os parâmetros do MPC são encontrados em Filali & Wertz (2001), de Almeida et al. (2006) e de Almeida et al. (2009a).

O processo de lingotamento contínuo consiste, basicamente, da transformação do aço, inicialmente em seu estado líquido e temperatura superi-

ores a  $1600^{\circ}\text{C}$ , em placas de aço de formato, espessura, superfície, consistência e propriedades mecânicas conhecidas, segundo padrões existentes. É no molde da máquina de lingotamento contínuo que se realiza o processo de resfriamento primário, formando uma “casca” de aço sólido. Para que isto seja realizado é necessário manter um nível constante e predeterminado de aço líquido no seu interior em diferentes condições de processo.

Apesar da aplicação de controlador clássico Proporcional-Integral-Derivativo (PID) ser ainda encontrado no controle do molde, a sua frágil robustez diante dos efeitos e variações de parâmetros presentes no processo de lingotamento tem impulsionado pesquisadores do mundo inteiro, tanto na siderurgia quanto no meio acadêmico, ao desenvolvimento de técnicas mais sofisticadas de controle. Em [Yoo & Kueon \(1997\)](#) mostra-se a aplicação de uma técnica de “controle por modo deslizante”. Em [Kurokawa et al. \(1993\)](#) é desenvolvido um controlador  $H_{\infty}$  e em [Tao & Wang \(2005\)](#) são utilizadas a Lógica *Fuzzy* e Redes Neurais para resolver o problema de controle do nível do molde. Não se encontram na literatura estudos sobre aplicações do MPC não linear do nível do molde do lingotamento contínuo. Os artigos existentes sobre este assunto estão relacionados ao controlador GPC para o molde linear, conforme mostrado em [Loureiro \(1994\)](#), [Guo et al. \(2000\)](#) e [Sanchotene \(2009\)](#).

Duas técnicas MPC com Redes Neurais serão estudadas: uma delas utiliza Linearização instantânea do modelo neural do molde e a outra utiliza o gradiente da função objetivo para determinar a ação de controle. Tais métodos serão comparados quanto a robustez, estabilidade e qualidade de controle.

O capítulo é organizado da seguinte forma: na Seção 2 apresentam-se os conceitos básicos do MPC, os algoritmos MPC neural a serem implementados, e mostra-se como o Algoritmo Genético foi utilizado na sintonia destes controladores. Utilizando-se de dados reais, apresenta-se na Seção 3 a modelagem da malha de controle do nível do molde usando redes neurais e também é feita a representação matemática dos distúrbios existentes que afetam a qualidade do processo. Na Seção 4 discutem-se as simulações computacionais obtidas, apresentando a comparação entre as duas técnicas MPC neural. Na Seção 5 é feita a conclusão.

## 2. O Controlador Preditivo Baseado em Modelo (MPC)

Os algoritmos de controle pertencentes à família MPC são caracterizados pelos seguintes elementos:

- i) O Modelo do processo, que tem um papel decisivo no desempenho do controlador, devendo ser capaz de capturar a dinâmica do processo de forma a prever precisamente as saídas futuras, assim como ser simples na sua implementação e entendimento. O MPC é uma téc-

nica que engloba diferentes metodologias, pois utiliza vários tipos de modelos, nas diversas formulações.

- ii) O Preditor, onde se estima a saída futura, que é definida por  $\hat{y}(t+k)$ ,  $k = N_1, \dots, N_2$ , onde  $N_1$  e  $N_2$  são os horizontes de previsão mínimo e máximo, respectivamente. Esta estimação é feita a cada instante discreto de tempo  $t$  usando o modelo do processo.
- iii) A Trajetória de Referência, que é o sinal desejável que a saída do processo siga. A saída poderá ou não ser igual à Referência, dependendo do sinal de controle calculado pelo Otimizador. A Trajetória de Referência definida por  $w$  é, normalmente, uma aproximação suave do valor atual da saída, definida por  $y(t)$  em direção à referência conhecida (denominada  $r(t)$ ), por meio de um sistema de primeira ordem representado por:

$$\begin{aligned} w(t) &= y(t) \\ w(t+k) &= \alpha w(t+k-1) + (1-\alpha)r(k) \quad k = 1, \dots, N_2 - N_1 \end{aligned} \quad (1)$$

onde  $\alpha$  é um parâmetro entre 0 e 1 que suaviza a resposta do sistema à medida que se aproxima de 1.

- v) O Otimizador, através do qual se determina uma sequência de ações de controle definida por  $u(t+k)$ ,  $k = 0, 1, 2, \dots, N_u$ , onde  $N_u$  é denominado horizonte de controle. Estas ações são obtidas minimizando uma função custo, que geralmente é uma função quadrática dada por:

$$J = \sum_{k=N_1}^{N_2} \delta [\hat{y}(t+k) - w(t+k)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(t+k-1)]^2 \quad (2)$$

onde  $\Delta u(k) = u(k) - u(k-1)$ ;  $\delta$  e  $\lambda$  são parâmetros que ponderam os erros de previsão e o sinal de controle, respectivamente.

Uma solução explícita pode ser obtida se o modelo é linear e não existem restrições; caso contrário um método de otimização iterativo tem que ser usado. O primeiro sinal ( $u(t)$ ) da sequência das ações de controle  $u(t+k)$ ,  $k = 0, \dots, N_u - 1$  é enviado ao processo, enquanto que o restante da sequência é descartada. Tal estratégia é denominada de “Controle por Horizonte Retrocedente”.

## 2.1 Sintonia do MPC

O bom desempenho do controlador preditivo é obtido somente se o projetista definir adequadamente os parâmetros de sintonia: horizonte de predição ( $N_2$ ), horizonte de controle ( $N_u$ ), ponderação do erro de previsão ( $\delta$ ), fator de supressão ( $\lambda$ ) e constante da trajetória de referência ( $\alpha$ ). Entretanto, a seleção destes parâmetros não é uma tarefa fácil, na medida em que não há nenhum guia preciso para a seleção que assegure a estabilidade (Bitmead et al., 1990). Em geral, esta sintonia é feita através de heurística, baseada em tentativa e erro, sendo viável na sintonia do controlador preditivo de processos que possuem comportamento estável em malha aberta. No entanto, nos casos em que o processo é instável em malha aberta (como é o nível do molde do lingotamento contínuo), este procedimento torna-se entediante, podendo não obter uma resposta estável em malha fechada, ou quando o processo é estabilizado em malha fechada, o desempenho do controlador preditivo fica abaixo das expectativas.

Em Dougherty & Cooper (2003) é proposto um método para calcular os parâmetros de sintonia do DMC ( $N_u$ ,  $N_2$ ,  $\lambda$ ) analiticamente. No entanto, esta metodologia é aplicada nos casos em que os processos são estáveis em malha aberta e que possam ser aproximados por uma função de transferência de primeira ordem com atraso. Além disto, este método fornece elevados horizontes de controle, o que dificulta o cálculo computacional da ação de controle e favorece à obtenção de soluções inactivíveis, conforme mostra de Almeida et al. (2009a).

Uma outra metodologia utiliza os Algoritmos Genéticos (AG) para sintonizar os parâmetros do MPC, a qual foi inicialmente testada por Filali & Wertz (2001). Neste artigo, o controlador linear GPC foi aplicado em sistemas monovariáveis e irrestritos, considerando-se apenas os horizontes de controle e de previsão. Em de Almeida (2006), mostra-se que a inclusão dos parâmetros de sintonia ( $\alpha$ ,  $\delta$  e  $\lambda$ ) diminuem os horizontes de controle e previsão do GPC, o que facilita o esforço computacional realizado na obtenção da ação de controle. Além disto, em de Almeida et al. (2009a) mostra-se a eficiência da sintonia do algoritmo DMC usando AG em relação à metodologia proposta por Dougherty & Cooper (2003).

### 2.1.1 Sintonia do MPC através do algoritmo genético (AG)

Uma das principais características do AG é realizar ajuste de todos os parâmetros de maneira automática. A execução do algoritmo é feita da seguinte maneira: primeiramente deve-se fornecer o modelo do processo. Em seguida define-se os parâmetros necessários para a execução do AG tais como: o tamanho da população ( $M$ ), o número de gerações ( $G$ ), a taxa de *crossover* e a taxa de mutação, o tipo de função *fitness* considerada e o critério de seleção. Feito isto,  $M$  indivíduos são criados aleatoriamente por AG, que tem o seguinte formato:

$$[N_u, N_2, \alpha, \delta, \lambda], \quad (3)$$

sendo que a codificação é feita com variáveis reais.

Depois disto, o AG calcula o *fitness* de cada indivíduo dentro da população inicial, onde os parâmetros sintonizados por AG são colocados dentro do algoritmo de MPC que separa de acordo com o *fitness* de cada um os melhores dos piores indivíduos. A função *fitness*, denominada *Fit*, é dada por:

$$Fit(N_u, N_2, \alpha, \delta, \lambda) = \frac{1}{\sum (y - w)^2}, \quad (4)$$

onde  $y$  é a saída do processo e  $w$  é a referência.

Em seguida, o AG evolui a espécie através da realização das três operações genéticas (reprodução, *crossover* de ponto único e mutação aleatória), onde os melhores indivíduos tem maior probabilidade de ter descendentes nas gerações futuras. Neste trabalho, o método de seleção usado é o da roleta onde, através da Equação 4, se determina a porção que o indivíduo terá dentro da roleta. São feitas  $M - 1$  execuções da roleta para selecionar os indivíduos. Paralelo à seleção por roleta, foi utilizado o elitismo, que assegura que os resultados nunca serão piores do que os da geração anterior.

O AG executa um loop de  $G$  gerações e, ao final da última, apresentar-se-á o melhor indivíduo. O algoritmo utilizado na sintonia do MPC é o seguinte:

- Passo 0:** Determinar o tamanho da população  $M$ , a taxa de *crossover* a taxa de mutação;
- Passo 1:** Criar a população inicial;
- Passo 2:** Aplicar o vetor de parâmetros (Equação 3) calculados por AG dentro do algoritmo MPC Neural usado para obter o sinal de controle  $u$ ;
- Passo 3:** Determinar a saída  $y$  simulando o modelo do processo;
- Passo 4:** Determinar a referência futura pela Equação 1;
- Passo 5:** Calcular o *fitness* de cada indivíduo através da Equação 4;
- Passo 6:** Distinguir os melhores e os piores indivíduos;
- Passo 7:** Criar uma nova população usando as operações genéticas;
- Passo 8:** Voltar ao passo 2 até o número máximo de gerações ser alcançado;
- Passo 9:** Apresentar o resultado.

Com este procedimento, basta entrar com o modelo da planta que se deseja controlar para que o algoritmo calcule os valores dos parâmetros do MPC de forma automática sem a necessidade de um conhecimento adicional do processo, além do seu modelo. É importante mencionar que o algoritmo desenvolvido neste trabalho pode ser aplicado em qualquer tipo de processo onde se possa utilizar o MPC.

## 2.2 MPC neural

No MPC neural o modelo é obtido através de uma rede neural *feedforward* com pelo menos um neurônio na camada oculta, tendo em vista que tais redes são consideradas estimadores universais (Farago & Lugosi, 1993). No caso do modelo neural do nível do molde, a rede terá a estrutura NNARX, cuja previsão da saída  $k$  passos a frente pode ser escrita como:

$$\begin{aligned} \hat{y}(t+k) &= g[\hat{y}(t+k-1), \dots, \hat{y}(t+k-\min[k, n]), y(t-1), \\ &\dots, y(t-\max[n-k, 0]), u(t+k-d), \dots, u(t+k-d-m)], \end{aligned} \quad (5)$$

onde:  $n$  é o número de regressores da saída,  $m$  é o número de regressores da entrada,  $d$  é o atraso,  $g$  é uma rede *perceptron* de duas camadas.

As estratégias de controle MPC neural se diferem, basicamente, na forma de calcular a ação de controle que minimize a função custo  $J$ . Uma classe utiliza a regra do gradiente, cujo cálculo da ação de controle é feito iterativamente na direção contrária ao gradiente da função custo (Equação 2). Uma outra classe realiza uma linearização local da função não linear  $g$  da rede e determina a ação de controle usando a técnica GPC (Controle Preditivo Generalizado), que é um controlador preditivo linear baseado no modelo linear CARIMA. Os algoritmos destas duas estratégias de controle neural são descritos a seguir.

### 2.2.1 MPC neural com regra baseada no gradiente (PNRG)

Seja  $U^i(t) = [u^i(t), u^i(t+1), \dots, u^i(t+N_u-1)]$  o vetor de ações de controle a ser determinado a cada iteração  $i$  pelo algoritmo PNRG através da seguinte regra:

$$U^{(i+1)}(t) = U^{(i)}(t) + \mu^i d^i, \quad (6)$$

onde  $\mu^i$  é o passo com que  $U^i$  caminha para o mínimo de  $J$  e  $d^i$  é o vetor direção contrário ao gradiente da função  $J$  em relação a  $U^i$ , o qual é denotado por  $G[U^i]$ .

O algoritmo PNRG se baseia no método *quasi-Newton*, que considera  $d^i = -B^i G(U^i)$ , onde  $B^i$  é a aproximação da inversa da Hessiana de  $J$  em relação a  $U^i$ , calculada pelo método de Broyden-Fletcher-Goldfarb-Shanno (BFGS). O Algoritmo PNRG foi proposto por M.Nørgaard et al. (2000), sendo descrito resumidamente a seguir:

**Passo 0** Selecionar uma sequência de controle  $U^0$ , determinar  $G[U^0]$  e fazer  $B^0 = I$  e  $i = 0$ ;

**Passo 1** Obter  $d^i = -B^i G[U^i]$ , determinar o passo  $\mu^i$  e atualizar a sequência de controle conforme a Equação 6;

**Passo 2** Ir para o Passo 4 se  $|U^{i+1} - U^i| < \epsilon$  ou o número máximo de iterações for atingido.

**Passo 3** Fazer  $i = i + 1$ , atualizar o cálculo de  $B^i$  usando o método BFGS, ir para o Passo 1.

**Passo 4** Aceitar a sequência  $U^i$  como o mínimo de  $J$  e parar.

### 2.2.2 MPC neural com linearização instantânea (PNLI)

A idéia da linearização instantânea é extrair o modelo linear de uma rede neural não-linear em cada período de amostragem. Assume-se que o modelo de entrada e saída de uma rede neural do sistema a ser controlado é avaliado como:

$$y(t) = g[\varphi(t)] \quad (7)$$

onde o vetor de regressão  $\varphi(t)$  é dado por:

$$\varphi(t) = [y(t-1), \dots, y(t-n), u(t-d), \dots, u(t-d-m)] \quad (8)$$

O princípio da linearização instantânea é o seguinte (mais detalhes podem ser encontrados em [M.Nørgaard et al. \(2000\)](#)):

- i) Interpretar o vetor de regressão como um vetor definindo os estados do sistema e o tempo  $t = \tau$ . Linearizar  $g$  em torno do estado corrente  $\varphi(t)$  para obter o modelo aproximado:

$$\tilde{y}(t) = -a_1 \tilde{y}(t-1) - \dots - a_n \tilde{y}(t-n) + b_0 \tilde{u}(t-d) + \dots + b_m \tilde{u}(t-d-m), \quad (9)$$

onde:

$$a_i = -\frac{\partial g[\varphi(t)]}{\partial y(t-i)} \Big|_{\varphi(t)=\varphi(\tau)} \quad b_i = -\frac{\partial g[\varphi(t)]}{\partial u(t-d-i)} \Big|_{\varphi(t)=\varphi(\tau)} \quad (10)$$

e:

$$\tilde{y}(t-i) = y(t-i) - y(\tau-i) \quad \tilde{u}(t-i) = u(t-1) - u(\tau-1) \quad (11)$$

- ii) Separando a parte da expressão que contém os componentes do vetor de regressão corrente ( $\varphi(t)$ ), o modelo linear aproximado pode ser assim representado:

$$y(t) = [1 - A(q^{-1})]y(t) + q^{-d}B(q^{-1})u(t) + \zeta(\tau), \quad (12)$$

onde o termo  $\zeta(t)$  é dado por:

$$\zeta(t) = y(\tau) + a_1y(\tau-1) + \dots + a_ny(\tau-n) - b_0u(\tau-d) - \dots - b_mu(\tau-d-m) \quad (13)$$

Os coeficientes  $a_i$  e  $b_i$  são os coeficientes dos polinômios  $A(q^{-1})$  e  $B(q^{-1})$  do modelo CARIMA.

Após obtido o modelo CARIMA, usa-se o algoritmo do controlador linear GPC para determinar a ação de controle, conforme a metodologia apresentada em [Camacho & Bordons \(2004\)](#).

### 3. Modelagem da Malha de Controle do Nível do Molde

O controle do nível de aço no molde é uma das malhas mais importantes em processo de lingotamento contínuo, influenciando diretamente na qualidade do aço produzido e, conseqüentemente, em seu valor comercial. O sistema utilizado atualmente na usina da Arcelor Mittal Tubarão é constituído de três partes principais ilustradas na Figura 1, a saber: unidade de medição, unidade de controle, unidade de atuação hidráulica.

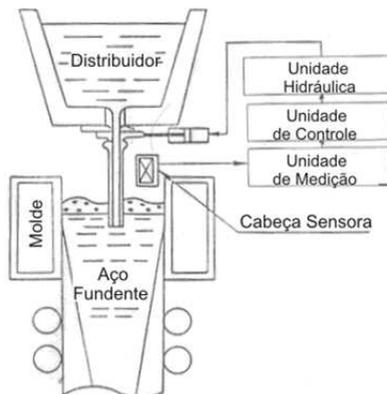


Figura 1. Configuração básica do sistema de controle.

O funcionamento consiste na colocação de um sensor acima da superfície do aço líquido recoberto por pó fluxante. O sinal elétrico captado

pelo sensor é tratado na unidade amplificadora, que envia o sinal para a unidade de controle. Nesta última unidade estão contidas as rotinas de controle e alarmes, através de um algoritmo específico para controle do nível. A unidade de controle avalia o comportamento do nível medido e envia as devidas correções ao sistema de atuação hidráulica, que posiciona um cilindro acoplado a uma válvula gaveta, utilizada para regular a vazão de aço do distribuidor para a válvula submersa (esta válvula localiza-se na saída da válvula gaveta, e é responsável pelo transporte do aço para o molde). Com a posição da válvula gaveta (mais precisamente, sua placa central), estará definida a vazão de entrada de aço na válvula submersa e, conseqüentemente, no molde. Este, em uma situação de regime permanente, deve contrapor ao volume de aço retirado do molde para continuidade do processo e formação das placas na saída da máquina.

A válvula gaveta é um dispositivo formado por três placas sobrepostas que possuem um orifício idêntico em cada uma destas placas, onde a placa central é provida de movimento horizontal ( $X_{sg}$ ), fazendo com que a área de passagem de aço, consista na área de intersecção dos orifícios, conforme é ilustrado pela Figura 2. Obtém-se a área efetiva ( $A_s$ ) de passagem de aço através de um cálculo considerando a intersecção de orifícios concêntricos de raio  $R$ . De acordo com Yoo & Kueon (1997), a área efetiva é dada por:

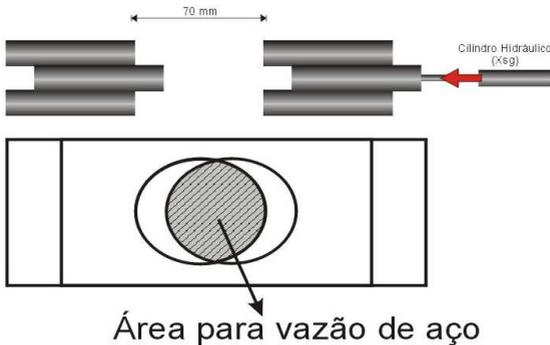


Figura 2. Formato da válvula gaveta do distribuidor.

$$A_s = 2 \left[ R^2 \cos^{-1} \left( \frac{R - \frac{X_{sg}(t)}{2}}{R} \right) - \left( R - \frac{X_{sg}(t)}{2} \right) \sqrt{RX_{sg} - \left( \frac{X_{sg}(t)}{2} \right)^2} \right] \quad (14)$$

A válvula gaveta está localizada logo após a saída distribuidor e o transporte do fluxo do aço da válvula gaveta para o molde é feita através

da válvula submersa. Considerando o comprimento da válvula submersa de  $L = 600\text{mm}$  e a velocidade do fluxo de aço na saída da válvula gaveta de  $V_n = \sqrt{2 \cdot g \cdot h}$ , o atraso de transporte será dado por  $T_v = 0,11\text{s}$ . Assim, a equação dinâmica do nível do molde obtida em [Yoo & Kueon \(1997\)](#), é dada por:

$$\frac{dH(t)}{dt} = \frac{1}{A_m} [A_s \cdot V_n(t - T_v) - A_m \cdot V_{cs}(t)] \quad (15)$$

onde:  $H$  é a altura do nível do molde;  $A_m$  é a área do molde;  $g$  é a gravidade;  $h$  é a altura do fluido entre os níveis de aço do distribuidor e do molde e  $V_{cs}$  velocidade de lingotamento ou extração da placa.

A partir da Equação 15 chega-se à seguinte função de transferência:

$$H = \frac{1}{A_m s} \left[ A_s e^{-sT_v} \sqrt{2gh} - A_m V_{cs} \right] \quad (16)$$

caracterizando-se, portanto, como uma malha integradora (não estável em malha aberta).

Para a representação matemática dos demais blocos que compõem a malha de controle, foram realizados experimentos na planta real, afim de identificar o modelo conforme mostra [Sanchotene \(2009\)](#). A função de transferência (Equação 17) representa o controlador da malha de posição da válvula gaveta (malha escrava) e o sistema hidráulico responsável por sua movimentação física.

$$G_p(s) = \frac{0.25}{0.5s + 1} \quad (17)$$

A função de transferência (Equação 18) representa o bloco referente ao sensor de nível do molde.

$$G_S(s) = \frac{e^{-0,02s}}{0.4s + 1} \quad (18)$$

Desta maneira, observa-se na Figura 3 o sistema não-linear completo simulado no Matlab para representação do processo real de controle do nível do molde.

### 3.1 Os distúrbios presentes no controle do nível do molde

Em uma malha real de controle de nível de molde verifica-se a ocorrência de uma série de efeitos inerentes ou não ao processo metalúrgico de produção de placas que afetam de forma direta e indireta o regime estável do nível de aço na superfície do molde, podendo ser interpretados como distúrbios a serem controlados/reduzidos pelo sistema de controle. O aparecimento destes efeitos (ou distúrbios) e a conseqüente variação demasiada do nível de aço no molde, tem como conseqüência final o surgimento de defeitos

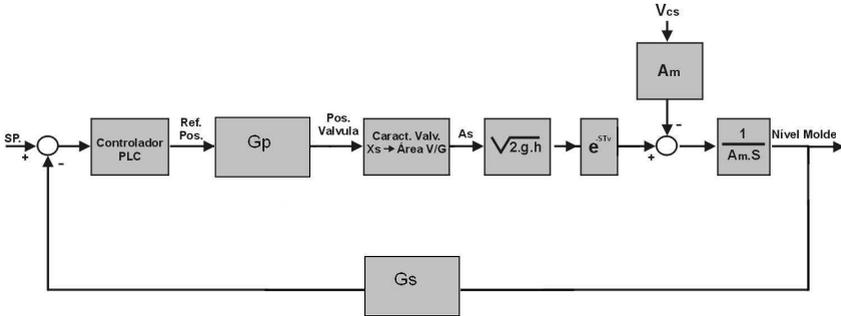


Figura 3. Diagrama de blocos da malha de controle de nível do molde.

nas placas produzidas pela máquina de lingotamento, além dos riscos de transbordamento de aço para fora do molde e *break-outs*.

A seguir são descritos os efeitos que mais interferem na estabilidade de uma malha de controle de nível de molde, seus comportamentos e as respectivas propostas de representação destes, objetivando retratar com fidelidade o processo a ser controlado.

### 3.1.1 Distúrbio de *clogging*

O acúmulo progressivo de deposição de alumina ou outros produtos nas paredes da válvula submersa e/ou da própria válvula gaveta é conhecido como efeito *clogging*. Como este acúmulo é progressivo, há a redução também progressiva da área efetiva de passagem de aço, sendo necessário a compensação pelo controlador da abertura da válvula gaveta. Em adição a este efeito, há também o desprendimento abrupto da alumina acumulada (*unclogging*), liberando o fluxo e provocando a necessidade de ação rápida do controlador para retornar o processo às condições anteriores.

O comportamento do efeito *clogging* pode ser perfeitamente simulado em Matlab, incluindo este na malha da válvula gaveta. Com a observação da posição da válvula gaveta durante o efeito *clogging*, pode-se representar o movimento de abertura da válvula por uma função aleatória crescente no tempo à medida que aumenta a obstrução da válvula. Após terminar o efeito do entupimento, a válvula atinge a abertura máxima, permanecendo até a ocorrência de *unclogging* (desentupimento) alguns segundos depois. Assim, a simulação da diminuição da área de passagem do aço pela válvula gaveta e/ou submersa, pode ser realizada multiplicando a área da válvula gaveta ( $A_s$ ) sem obstrução pela função de redução da área ( $A_{clg}$ ) que é dada pela expressão:

$$A_{clog}(t) = \begin{cases} 1 - \beta_1(t - T_1), & T_1 \leq t < T_2 \\ f, & T_2 \leq t < T_3 \\ f + \beta_2(t - T_3), & T_3 \leq t < T_4 \end{cases}$$

onde  $T_1$  é o instante em que inicia o entupimento da válvula gaveta,  $T_2$  é o instante em que ocorre o entupimento máximo,  $T_3$  e  $T_4$  são os instantes em que inicia e termina o processo do desentupimento, respectivamente, e  $f$  é o fator de entupimento máximo, tal que  $0 < f < 1$  e  $f = 1 - \beta_1(T_2 - T_1) = 1 - \beta_2(T_4 - T_3)$ .

No processo de *clogging* real, por observação, é prudente afirmar que os instantes de  $T_1 < T_2 < T_3 < T_4$  são aleatórios e encontram-se entre os intervalos:  $20s < T_2 - T_1 < 120s$  e  $10s < T_3 - T_2 < 60s$ ,  $0 < T_4 - T_3 < 5s$ .

Com o objetivo de simular o entupimento da válvula submersa e/ou da válvula gaveta, identifica-se na Figura 3 que se deve incluir um ganho variável dado por  $K = A_{clog} * A_s$  para representar o efeito *clogging* no ponto onde está representada a área de passagem do aço, conforme ilustra a Figura 4.

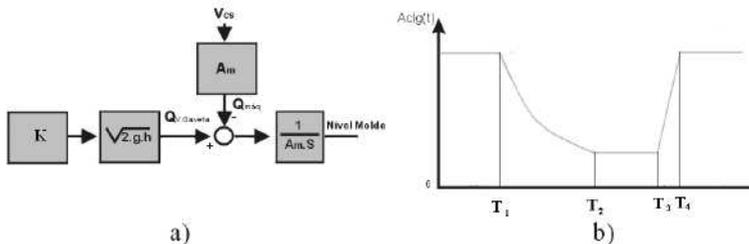


Figura 4. a) Diagrama de blocos com a inclusão do efeito de *clogging* ( $K = A_s \times A_{clog}$ ); b) Representação gráfica da função de redução da área  $A_{clog}$ .

### 3.1.2 Distúrbio de *bulging*

Conforme comentado anteriormente, após a passagem pelo molde, a placa de aço no interior do veio (ou seja, no interior da máquina de lingotamento) possui uma camada de “pele” e, em seu interior, ainda há aço líquido que será gradativamente resfriado pela adição de água na superfície até que esteja totalmente solidificada em seu interior, antes de sair da máquina.

Esta placa de aço, ainda no estado líquido, exerce, portanto, uma força ferrostática que tende a empurrar o aço líquido para fora do veio, movimento impedido pelos rolos que compõem os segmentos. Como estes rolos possuem um espaçamento  $D$  entre eles, a pele de aço formada cede um pouco, de forma que o aço ocupe parte deste espaçamento. Ao passar

pelos rolos, ocorre a diminuição do espaço e conseqüente laminação deste aço sólido/líquido, provocando o rearranjo de aço líquido no interior da máquina.

Após passar por entre os rolos, o aço expande-se novamente até o próximo conjunto (caso ainda não tenha sido formada pele espessa o suficiente para se contrapor à pressão ferrostática). Este fenômeno de retração e expansão ocorre simultaneamente por toda a máquina onde há presença de aço líquido no interior, chegando até o molde. Como a máquina de lingotamento possui uma velocidade de extração da placa em seu interior, denominada  $V_{cs}$ , verifica-se, como efeito decorrente da expansão e retração interna da placa no interior da máquina, o surgimento de ondas senoidais na superfície do molde.

Estas ondas são conseqüência, portando, da variação de volume de aço no interior do veio, função do espaçamento entre os respectivos rolos de onde ocorre o fenômeno e da velocidade de extração. A frequência de oscilação pode ser determinada com boa aproximação por:

$$f_{oscilao}(Hz) = \frac{V_{cs}(m/min)}{D(m) * 60} \quad (19)$$

O aparecimento do fenômeno de *bulging* ocorre principalmente na região de desdobramento do aço no interior da máquina, próximo ao molde, onde há maior volume de aço líquido no interior da placa e, obviamente, menor espessura de “pele” de aço solidificado. A Equação 19 indica uma relação direta do fenômeno com aspectos construtivos da máquina: capacidade de refrigeração e de velocidade, distanciamento de rolos definidos em projeto e curvatura total de lingotamento.

Particularmente, dependendo da região onde há a ocorrência de *bulging* e do espaçamento dos rolos que compõem esta região, pode ocorrer o aparecimento de frequências de oscilação diferentes e simultâneas na superfície do molde.

A amplitude deste fenômeno está diretamente relacionada com a capacidade de refrigeração e tipo de aço lingotado (quanto menor a quantidade de carbono em sua composição em conjunto com a adição de ligas, maior é a maleabilidade do material e amplitude do fenômeno), dentre outros fatores de menor contribuição.

Para a simulação do efeito *bulging* no processo identificado, procede-se da mesma forma que o realizado no item anterior. É, então, necessário definir o ponto correto de inserção do fenômeno no diagrama de blocos do processo identificado (Figura 3). Conforme descrito anteriormente, entende-se o efeito *bulging* como uma variação de volume no molde. A Figura 5 representa um possível ponto de inserção do efeito no modelo obtido.

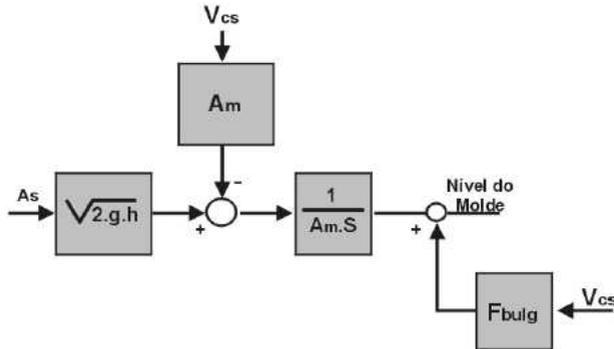


Figura 5. Ponto de inclusão do efeito *bulging* no processo.

### 3.2 O modelo neural do nível do molde sem distúrbios

Com o objetivo de desenvolver um modelo do processo para fins de implementação da estratégia de controle avançada, realiza-se nesta seção a identificação do sistema utilizando-se as redes neurais artificiais. A Figura 6 mostra o conjunto de entrada-saída (*input-output*) utilizado na identificação. Estes dados foram gerados a partir de uma perturbação aleatória imposta à variável de entrada, dentro de certo limite (limite este factível do ponto de vista operacional), onde a abertura máxima da válvula gaveta é de 70 mm. Os dados foram divididos em dois subconjuntos, treinamento e validação. Em seguida os dados (treinamento e teste) foram escalonados de tal modo a apresentarem média zero e variância 1, procedimento este recomendável quando se tem em mente a utilização das redes neurais na correlação de tais dados.

A escolha de um regressor apropriado para descrever a dinâmica de um sistema não é uma tarefa fácil e pressupõe certo empirismo. Uma variável importante na escolha do regressor é a ordem do sistema. Tal ordem pode ser determinada a partir do conhecimento físico do sistema em estudo ou pode ser inferida a partir do conjunto de dados de entrada-saída. Neste último caso, pode-se empregar a metodologia proposta na literatura (Xiangdong & Haruhiko, 1993) para se determinar os valores  $m$  e  $n$  da Equação 5.

Utilizando a função Lipschitz conforme o procedimento apresentado em Xiangdong & Haruhiko (1993), determina-se a ordem do sistema, adotando-se um modelo com  $m = 2$ ,  $n = 2$ . A vantagem deste procedimento é evitar uma determinação iterativa e tediosa da quantidade de entradas a serem utilizadas no regressor.

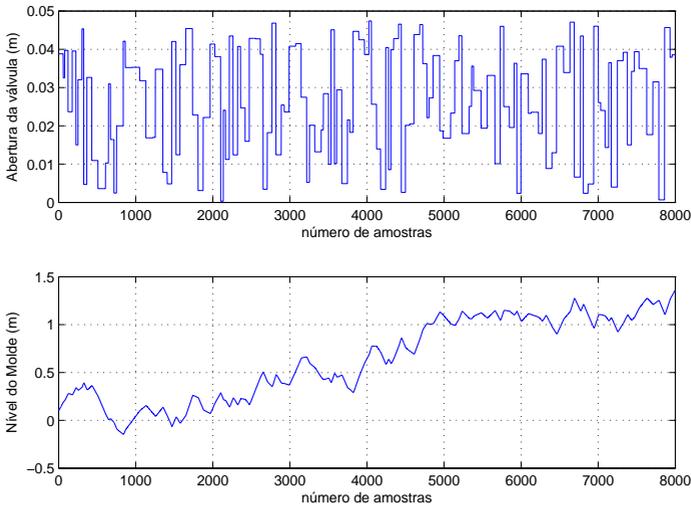


Figura 6. Dados de entrada e saída utilizados na identificação do modelo.

O atraso do sistema (valor de  $d$  na Equação 5) foi escolhido comparando-se os vários modelos com  $m = 2$  e  $n = 2$ , e diferentes valores para o atraso, selecionando o melhor a partir do critério de informação de Akaike (*AIC Akaike's Information Criterion*) e do critério de erros de predição final (*FPE Final Prediction Error*). Através desta seleção, tem-se um modelo com  $m = 2$ ,  $n = 2$  e  $d = 1$ .

Agora identifica-se a rede neural na forma do modelo ARX, rede que é chamada de NNARX. Inicialmente, investigar-se-á o efeito do número de neurônios na camada interna (com função de ativação tangente hiperbólica na camada interna). A rede neural utilizada tem a arquitetura totalmente conectada e função linear na camada de saída. Na Figura 7 mostra-se a saída real do processo e a saída identificada pela rede neural, assim como o erro de predição um passo a frente. A arquitetura da rede utilizada, possui uma camada oculta com 10 neurônios. Como se trata do modelo NNARX, este não possui característica recorrente, sendo que as entradas da rede serão definidas pelas saídas do processo  $y$  em dois intervalos de tempo anteriores  $[y(t-1), y(t-2)]$ , juntamente com o sinal de entrada  $u$  em dois instantes anteriores  $[u(t-1), u(t-2)]$ , caracterizando uma estrutura na forma  $[2, 2, 1]$ .

Observa-se na Figura 7, que a identificação realizada por redes neurais apresenta uma precisão satisfatória para o caso de predição um passo a

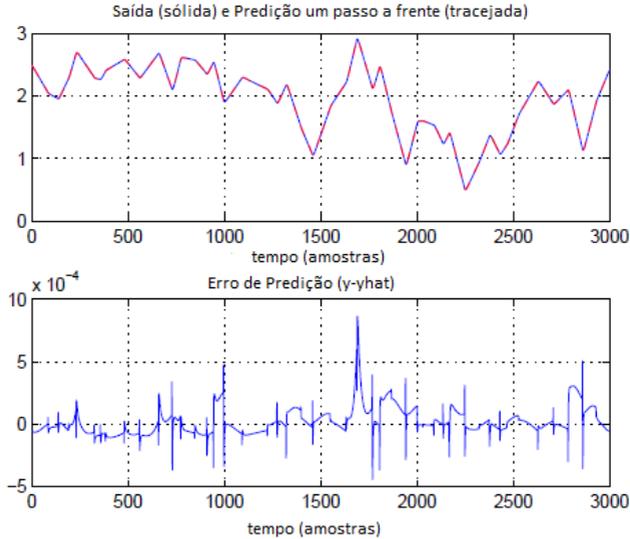


Figura 7. Validação do modelo neural.

frente. Porém, quando a utilização da rede for em controle, o modelo não pode ser julgado por apenas este item.

Uma vez que o modelo obtido por redes neurais será utilizado como modelo de um processo, o qual será usado no cálculo do controlador preditivo, uma característica muito importante que deve ser analisada é a capacidade de predição do modelo. Desta forma, a Figura 8 mostra a saída real do processo juntamente com a saída da rede com predição de 30 passos a frente.

Através da Figura 8 nota-se a precisão que o modelo criado pela rede neural possui em mapear o sistema real.

#### 4. Resultados da Simulação Computacional

Uma vez que o modelo neural já está bem definido, apresenta-se nesta seção a simulação do controle preditivo neural do nível do molde do lingotamento contínuo. Os algoritmos propostos por M.Nørgaard et al. (2000) e discutidos na Seção 2 para o controle preditivo neural foram modificados para incluir a sintonia feita através de algoritmo genético, cuja população é constituída por 50 indivíduos com 30 gerações, e taxas de *crossover* e mutação iguais a 0,8 e 0,1, respectivamente, sugeridas por Srivivas & Patnaik (1994).

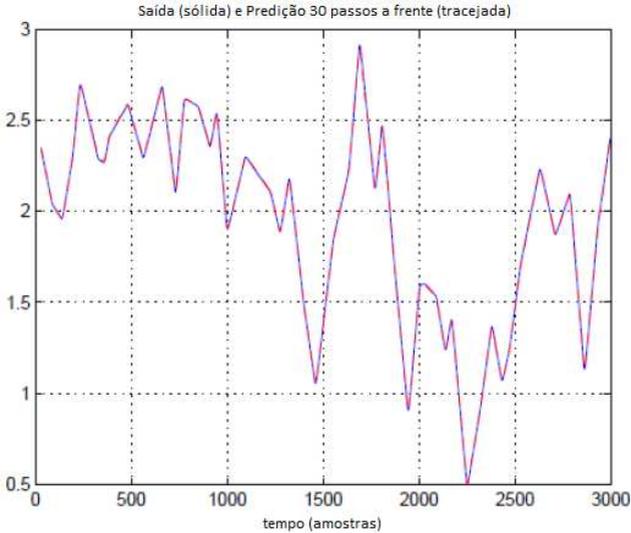


Figura 8. Saída real e saída prevista pela rede neural a 30 passos a frente.

As condições operacionais em que a simulação foi realizada no MatLab são:  $h=1,2$  m (Altura do Distribuidor);  $A_m=0,25$  m<sup>2</sup> (área do molde);  $\bar{X}_{sg}=0,35$  m (abertura da válvula gaveta para uma velocidade constante de 0,02 m/seg). O período de amostragem foi de 0,1 seg.

As restrições no processo são: abertura da válvula gaveta ( $u(t) \leq 0,7$  m) e na sua velocidade de abertura ( $\Delta u(t) \leq 0,0041$  m por período de amostragem) e nos limites físicos do sensor do nível do molde ( $y(t) \leq 0,2$  m). As medidas são dadas em metros para evitar a saturação do modelo neural, caso as unidades adotadas fossem em cm ou mm.

#### 4.1 Controle preditivo neural sem distúrbios

Primeiramente, não se considera a presença de distúrbios tanto na simulação do controlador preditivo neural com linearização instantânea (PNLI), quanto na simulação do controlador preditivo neural com a regra do gradiente (PNRG). Nesta situação, os parâmetros ajustados por AG para a sintonia do controlador PNLI foram:  $N_u=10$ ,  $N_2=10$ ,  $\alpha=0,8843$ ,  $\delta=91,3287$  e  $\lambda=0,0796$ . Os parâmetros ajustados por AG para a sintonia do controlador PNRG foram:  $N_u=1$ ,  $N_2=1$ ,  $\alpha=0,5755$ ,  $\delta=747,6628$  e  $\lambda=645,5345$ . Desta forma, as respostas a diversos degraus do nível do molde lingotamento contínuo, apresentadas para os dois controladores sem a presença de distúrbios, são semelhantes e possuem a forma mostrada na Figura 9.

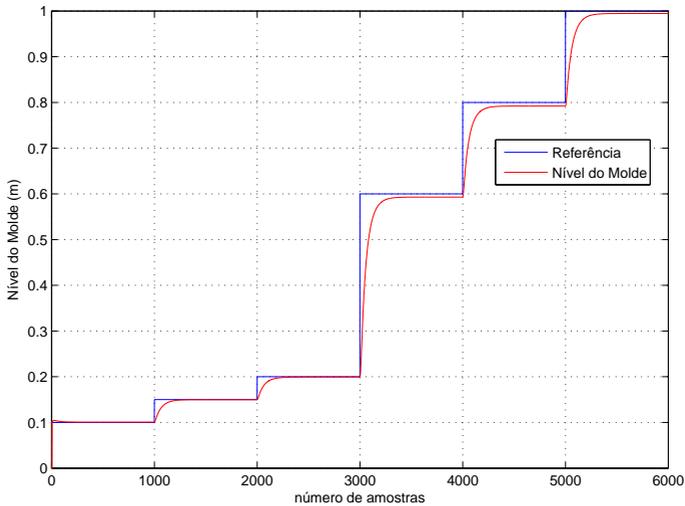


Figura 9. Resposta a diferentes degraus do nível com os controladores PNRG e PNLI.

Observa-se na Figura 9 que o nível segue a referência em toda região do molde, apresentando uma resposta suave e sem oscilações. É importante mencionar que a inversa da função de *fitness* (Equação 4), apresentada na simulação foi de 7,7477 para o PNLI e de 7,754 para o PNRG. Os tempos de processamento de cada simulação foram de 32,67 seg. para o PNLI e de 48,44 seg. para o PNRG. Isto se justifica uma vez que o algoritmo PNRG executa, a cada período de amostragem, um procedimento iterativo para calcular a ação de controle que converge para mínimo da função custo. Já no PNLI a ação de controle é calculada num único passo a cada período de amostragem.

#### 4.2 Controle preditivo neural com distúrbio de *bulging*

O distúrbio de *bulging* pode ser representado por um sinal senoidal com amplitude de 20 mm e uma frequência  $w = (2\pi) (0,077)$  rad/seg, interferindo no nível do molde. Desta forma, a resposta apresentada pelo sistema de nível do lingotamento contínuo com a presença do distúrbio de *bulging* utilizando o PNLI e o PNRG permitiu que as oscilações no nível do molde fossem reduzidas de 20 mm para cerca de 5 mm, o que mostra a eficácia destes controladores para este tipo de distúrbio.

Os parâmetros de sintonia dos controladores PNLI e PNRG foram os mesmos anteriores para o caso sem distúrbio. O tempo de simulação foi de 16,73 seg. para o PNLI e de 25,49 seg. para o PNRG.

### 4.3 Controle preditivo neural com distúrbio de *clogging*

Neste parte é realizada a sintonia do controlador preditivo considerando o pior caso do distúrbio de *clogging* para obter um controlador preditivo robusto. O distúrbio de *clogging* mais severo acontece quando o processo de entupimento da válvula ocorre no menor intervalo de tempo possível ( $T_2 - T_1 = 20$  seg.), permanecendo obstruída com um determinado fator de entupimento  $f$  durante um intervalo de tempo mais longo ( $T_3 - T_2 = 60$  seg.), para em seguida ocorrer um desentupimento abrupto ( $T_4 - T_3 = 0$ ). Nas simulações, o fator de entupimento  $f$  foi incrementado gradativamente de 10% em 10% para se obter o entupimento máximo para o qual o controlador consegue estabilizar o nível.

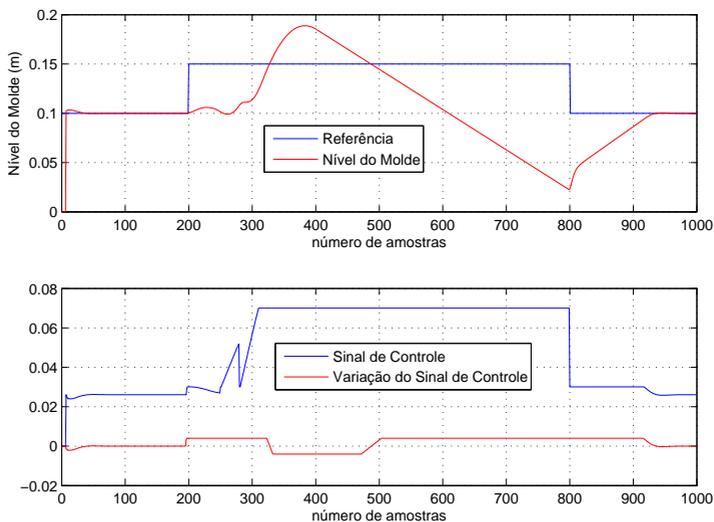
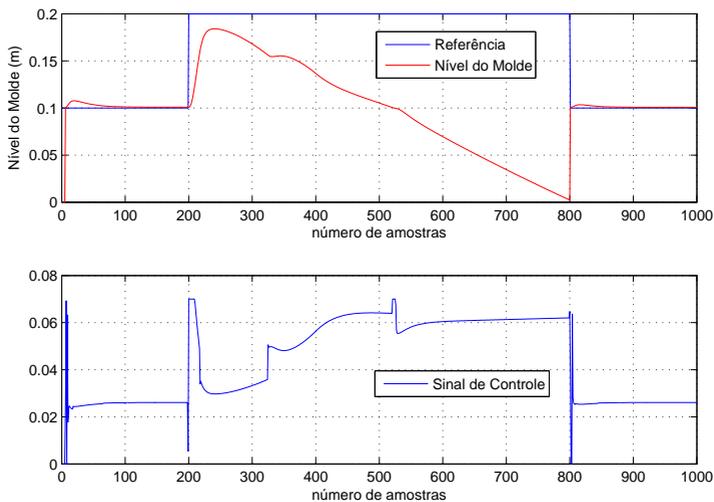
As Figuras 10 e 11, mostram as respostas do nível do molde considerando o distúrbio de *clogging* mais severo, ou seja, com entupimento de até 80% na válvula para o PNLI e de 75% para o PNRG. Tais Figuras mostram também os respectivos sinais de controle e variação do sinal de controle. É importante mencionar que os parâmetros utilizados nos exemplos anteriores (sem distúrbio) não foram capazes de estabilizar o sistema com o entupimento citado. Assim, foi necessário fazer a re-sintonia dos parâmetros usando o AG, de modo a aumentar a robustez dos controladores. Estes parâmetros foram  $N_u=2$ ,  $N_2=4$ ,  $\alpha=0,1342$ ,  $\delta=14,2027$  e  $\lambda=1,5825$  para o PNLI e  $N_u=2$ ,  $N_2=4$ ,  $\alpha=0,1522$ ,  $\delta=825,8170$  e  $\lambda=538,3424$  para o PNRG.

Observa-se nas Figuras 10 e 11 que é aplicado um degrau no instante em que se inicia o processo de entupimento até o instante em que ocorre o desentupimento da válvula. Isto é feito durante o período em que ocorre o *clogging* mais severo, para garantir que o nível do molde suba no início do *clogging* e não atinja valores negativos no final do *clogging*. Na prática, se o nível atingisse valores negativos, o molde se esvaziaria e a produção deveria ser paralisada, implicando em prejuízos operacionais.

Para que o algoritmo do controlador PNRG consiga controlar o nível com distúrbio *clogging*, é necessário reinicializá-lo no momento em que acontece o desentupimento. Isto é devido aos erros numéricos do cálculo da Hessiana aumentarem no instante de desentupimento, causando instabilidade no algoritmo PNRG. O tempo de simulação obtido neste experimento foi de 4,86 seg. para o PNLI e de 17,48 seg. para o PNRG.

## 5. Conclusão

Neste capítulo foi analisado o desempenho do controlador preditivo neural no controle do nível do molde sujeito a distúrbios de *bulging* e *clogging*. Dois

Figura 10. Simulação com controlador PNLI e *clogging* com  $f=0,8$ .Figura 11. Simulação com controlador PNRG e *clogging* com  $f=0,75$ .

algoritmos foram implementados: o primeiro utiliza a regra do gradiente (PNRG) para atualizar a lei de controle e o segundo realiza a linearização da rede (PNLI) para obter o controle através da técnica de controle preditivo linear GPC. Tais controladores foram sintonizados através de um Algoritmo Genético para obter a ação de controle robusta diante dos efeitos dos distúrbios.

A partir dos resultados obtidos por simulação computacional, conclui-se que as estratégias PNLI e PNRG possuem desempenhos semelhantes quando o processo está sem distúrbio ou quando apresenta o distúrbio de *bulging*, reduzindo as oscilações na saída de 20 mm para 5 mm, o que está dentro da faixa tolerável de de 10 mm. No entanto, na presença do distúrbio de *clogging*, foi necessário fazer a re-sintonia dos parâmetros para garantir que o molde não transborde ou se esvazie durante o período em que ocorre este distúrbio. Para o pior caso de *clogging*, observou-se uma pequena superioridade do controlador PNLI em relação ao PNRG, pois o primeiro mantém o nível do aço dentro dos limites físicos do sensor, quando o entupimento da válvula atinge 80%, enquanto que no PNRG o máximo entupimento chega a 75%. A explicação para isto é que os erros numéricos no cálculo da aproximação da inversa da Hessiana aumentam consideravelmente no instante de desentupimento da válvula, sendo necessário reiniciar o algoritmo PNRG após este momento. Outra desvantagem é o tempo de processamento maior do PNRG, que necessita de 8,2 milissegundos para calcular a ação de controle a ser aplicada no processo (no PNLI isto é executado na metade deste tempo no máximo). No entanto, este tempo de processamento é suficiente para realizar o controle em tempo real do nível do molde, já que o período de amostragem utilizado no controlador real é 0,01 segundos.

Outro comentário relevante é que nos controladores PNLI e PNRG, a ação de controle é do tipo derivativa ( $\Delta u(t) = u(t) - u(t - 1)$ ). Isto é devido ao fato do nível do molde apresentar uma elevada instabilidade à ação de controle integrativa dada por  $u(t) = u(t - 1) + \Delta u(t)$ , conforme mostra a análise de estabilidade feita por [Sanchoene \(2009\)](#) para o molde linear.

É oportuno esclarecer que estas conclusões não poderiam ser obtidas se a sintonia do controlador preditivo fosse feita por tentativa e erro, pois para este tipo de processo (instável em malha aberta) esta metodologia não é viável. Para trabalhos futuros pretende-se adaptar filtros à rede neural, tendo em vista que o sinal que entra na rede vindo do sensor de nível possui ruídos provocados pela injeção do gás argônio, presente no processo de lingotamento contínuo do aço.

## Agradecimentos

Os autores agradecem à Arcelor Mittal Tubarão pelo apoio e incentivo a realização desta pesquisa.

## Referências

- de Almeida, G., *Sintonia de Controle Preditivo Generalizado Através de Algoritmo Genético*. Dissertação de mestrado, Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, ES, 2006.
- de Almeida, G.; Salles, J. & Denti Filho, J., Optimal tuning parameters of the dynamic matrix predictive controller with constraints. *Latin American Applied Research*, 39(1):33–40, 2009a.
- de Almeida, G.; Salles, J. & Filho, J., Using genetic algorithms for tuning the parameters of a generalized predictive control. In: *VII Conferência Internacional de Aplicações Industriais (INDUSCON)*. Recife, PE, 2006.
- de Almeida, G.; Salles, J. & Filho, J., Tuning a multivariable predictive control of a hot rolling mill with genetic algorithm. In: *Anais do IX Simpósio Brasileiro de Automação Inteligente (SBAI)*. Brasília, DF, 2009b.
- Bitmead, R.R.; Gevers, M. & Wertz, V., *Adaptive optimal control. The thinking man's GPC*. Englewood Cliffs, USA: Prentice-Hall, 1990.
- Camacho, E.F. & Bordons, C., *Model Predictive Control*. London, UK: Springer-Verlag, 2004.
- Clarke, D.; Mohtadi, C. & Tuffs, P., Generalized predictive control – Part I. The basic algorithm. *Automatica*, 23(2):137–148, 1987.
- Cutler, C. & Ramaker, B., Dynamic matrix control – a computer control algorithm. In: *Proceedings of AIChE National Meeting*. Houston, USA, 1979.
- Dougherty, D. & Cooper, D., Tuning guidelines of a dynamic matrix controller for integrating (non-self-regulating) processes. *Industrial Engineering Chemical Research*, 42(8):1739–1752, 2003.
- Farago, A. & Lugosi, G., Strong universal consistency of neural networks classifiers. *IEEE Transactions on Information Theory*, 39(4):1146–1151, 1993.
- Filali, S. & Wertz, V., Using genetic algorithms to optimize the design parameters of generalized predictive controllers. *International Journal of Systems Science*, 32(4):503–512, 2001.

- Guo, G.; Li, W.; Wang, J. & Liu, M., A CGPC controller in a continuous casting process. In: *Proceedings of 3<sup>rd</sup> World Congress on Intelligent Control and Automation*. Hefei, China, v. 4, p. 2783–2786, 2000.
- Keyser, R. & van Cuawenberghe, A., Extended prediction self-adaptive control. In: *Proceedings of IFAC Symposium on Identification and System Parameter Estimation*. New York, USA, p. 1317–1322, 1985.
- Kurokawa, T.; Kondo, T.; Mita, T.; Liu, K. & Sampei, M., Development of mold level control in continuous casting by  $h_{\infty}$  control theory. In: *Proceedings of 2<sup>nd</sup> IEEE Conference on Control Applications*. Vancouver, Canada, v. 2, p. 865–871, 1993.
- Loureiro, R., *Controle do Nível do Aço de uma Máquina de Lingotamento Contínuo*. Dissertação de mestrado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, 1994.
- Maciejowski, J., *Predictive Control with Constraints*. Englewood Cliffs, USA: Prentice Hall, 2000.
- Mattedi, A.; Almeida, G.; Salles, J. & Sançhotene, F., Controle preditivo do nível do molde do lingotamento contínuo na presença de distúrbios e restrições do processo. In: *Proceedings of IX IEEE/IAS International Conference on Industry Applications (INDUSCON)*. São Paulo, SP, p. 1–6, 2010.
- M.Nørgaard, ; Ravn, O.; Poulsen, N. & Hansen, L., *Neural Networks for Modelling and Control of Dynamic Systems*. London, UK: Springer-Verlag, 2000.
- Qin, S. & Badgwell, T., A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.
- Sançhotene, F., *Modelagem e Controle no Sistema de Nível de Aço no Molde do Lingotamento Contínuo*. Dissertação de mestrado, Departamento de Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, ES, 2009.
- Sançhotene, F. & Salles, J., Estudo de caso do sistema de controle do nível do molde do lingotamento contínuo: Modelagem e aplicação do gpc. In: *Anais do VI seminário Nacional de Controle e Automação (SNCA)*. Salvador, BA, 2009.
- Srivivas, M. & Patnaik, L.M., Genetic algorithms: a survey. *IEEE Computer*, 27(6):17–26, 1994.
- Tao, J. & Wang, N., Fuzzy neuron hybrid control for continuous steel casting. In: *Proceedings of 16<sup>th</sup> IFAC World Congress*. Prague, Czech Republic, v. 16, 2005.

- Xiangdong, H. & Haruhiko, A., A new method for identifying orders of input-output models for nonlinear dynamic systems. In: *Proceedings of American Control Conference*. San Francisco, USA, p. 2520–2523, 1993.
- Yoo, S. & Kueon, Y., Development of mold level controller using sliding mode control in continuous casting processing. In: *Proceedings of 5<sup>th</sup> IEEE Mediterranean Conference on Control and Systems*. Paphos, Cyprus, 1997.

## Notas Biográficas

**Fábio de Borba Sanchotene** é formado em Engenharia Elétrica pela Universidade Federal do Espírito Santo (2001), é Especialista em Automação Industrial pela Universidade de Uberaba (2003) e tem Mestrado em Automação pela Universidade Federal do Espírito Santo (2009). Atua como engenheiro eletricitista na ArcelorMittal desde 2002, sendo responsável pelo desenvolvimento de soluções na área de automação de sistemas de controle envolvendo máquinas de lingotamento e laminadores.

**Gustavo Maia de Almeida** é formado em Engenharia Elétrica pela Universidade Federal de São João Del Rey (2005). Possui mestrado e doutorado em Engenharia Elétrica pela Universidade Federal do Espírito Santo, concluídos em 2006 e 2011 respectivamente. As áreas de interesse em pesquisa são controle preditivo, algoritmos genéticos, redes neurais e lógica *fuzzy*. Atualmente é professor do curso de automação do Instituto Federal de Tecnologia no município de Serra (ES).

**José Leandro Félix Salles** é formado em Engenharia Elétrica pela Universidade Federal do Espírito Santo em 1987, e tem mestrado e doutorado em Engenharia Elétrica pela Universidade Estadual de Campinas (1991 e 1999, respectivamente). É Professor do Departamento de Engenharia Elétrica da UFES desde 1992. Desenvolve pesquisas nas áreas de modelagem, controle e otimização aplicados a sistemas de manufatura, sistemas industriais e sistemas de energia. Possui diversos artigos publicados em congressos e periódicos, tendo orientado alunos de mestrado e doutorado do Programa de Pós-Graduação em Engenharia Elétrica da UFES.

### Algoritmo Evolutivo Híbrido para Escalonamento Integrado na Agroindústria

Ademir Aparecido Constantino\*, Dario Landa-Silva e Wesley Romão

---

**Resumo:** Este capítulo aborda um problema envolvendo o escalonamento integrado de pessoal e caminhões em uma agroindústria brasileira que processa 1 milhão de aves por semana. O escalonamento deve assegurar o constante fornecimento de aves para a fábrica a fim de evitar a ociosidade. O objetivo é minimizar a quantidade de horas extras pagas para os catadores e o tempo de espera dos caminhões no pátio da fábrica. Propõe-se um algoritmo evolutivo combinado com busca local e simulação discreta determinística. Os experimentos foram executados com dados reais e os resultados mostram que o método é efetivo para produzir escalas robustas.

**Palavras-chave:** Escalonamento, Agroindústria, Algoritmos genéticos.

**Abstract:** *This chapter addresses a problem involving integrated personnel and lorry scheduling in a Brazilian farming industry that processes around 1 million poultries per week. The schedule should ensure constant supply of poultries to the factory to avoid idle processing time. The objective is minimizing the extra time paid to squads and minimizing the waiting time of lorries at the factory parking. We propose an evolutionary algorithm combined with local search and deterministic discrete simulation. The experiments were done with real data and the results show that the method is effective to produce robust schedules.*

**Keywords:** *Scheduling, Farming industry, Genetic algorithm.*

---

\*Autor para contato: [aaconstantino@uem.br](mailto:aaconstantino@uem.br)

## 1. Introdução

Escalonamento (*scheduling*) de atividades significa designar um conjunto de eventos ao longo do tempo, atendendo a um conjunto de restrições e, em geral, otimizando alguma função objetivo. Os problemas de escalonamento desempenham um papel importante dentro da computação, especialmente dentro da pesquisa operacional, pelo fato de fomentar a investigação de modelos computacionais e de novas técnicas algorítmicas.

O problema de escalonamento de atividades pode ser encontrado em diversos contextos, seja envolvendo máquinas (simples ou paralela) (Sheen & Liao, 2007; Blazewicz et al., 1994), processos (Sonmez & Gursoy, 2007; Bansal et al., 2009) ou pessoas (Yunes et al., 2000). O escalonamento de pessoal pode ser dividido em dois tipos: escalonamento de pessoal em local fixo e escalonamento de pessoal no setor de transportes (Bodin, 1983). O escalonamento no setor de transportes é um dos mais complexos do ponto de vista computacional (Bodin, 1983). O desafio computacional, portanto, é encontrar uma metodologia que obtenha uma solução satisfatória para o problema em tempo computacional viável.

Como reportado por Bodin (1983), a maioria dos problemas de escalonamento não admitem solução em tempo polinomial e são poucos os casos modelados por programação matemática utilizando algoritmos exatos (Yunes et al., 2000). O escalonamento na agroindústria tratado aqui se diferencia do escalonamento de transporte pelo fato da hora início de cada viagem ser desconhecido. O problema investigado surge no contexto de uma indústria que processa mais de 1 milhão de aves por semana onde pequenas melhorias no escalonamento podem resultar em redução significativa do desperdício, implicando em redução de custos. O escalonamento automatizado deve ser capaz de definir a escala de atividades das equipes de apanha, assegurando o constante fornecimento de suprimento para a indústria, respeitando um conjunto de restrições operacionais e otimizando uma função objetivo. Além disto, informações sobre o tamanho da fila de caminhões ou tempo de espera na fila são parâmetros que devem ser otimizados. Por outro lado, estas informações são complexas para serem integradas em um modelo de programação matemática. Considerando a complexidade de se criar um modelo de programação matemática e a dificuldade de incorporar informações das filas, optou-se por utilizar um algoritmo heurístico.

O problema em tela não tem sido reportado na literatura, exceto uma variação do mesmo apresentado por Hart et al. (1999), também baseado em um caso real no norte da Inglaterra. Neste caso, a empresa não desejava um escalonamento otimizado, mas apenas um sistema que automatizasse o escalonamento, minimizando o número de violações de restrições. Portanto, o caso foi tratado como um problema de satisfação de restrições, o qual foi abordado usando algoritmos genéticos. Porém várias caracteris-

ticas se diferenciam do problema investigado neste estudo. Na Seção 2 o problema será apresentado em detalhes, fazendo uma comparação com o trabalho de [Hart et al. \(1999\)](#).

Dada a complexidade para modelar o problema investigado e a dificuldade de calcular as informações da fila de espera, optou-se por implementar uma metodologia baseada em algoritmos genéticos, integrando-o à programação matemática e à técnica de simulação discreta determinística para cálculo das estatísticas das filas. O algoritmo possui duas fases hierárquicas, sendo que a primeira fase procura minimizar o custo com as equipes de apanha, enquanto que a segunda fase tenta melhorar o resultado focando na minimização do tamanho da fila de espera.

Algoritmos genéticos também tem sido utilizados em diversos problemas de escalonamento, seja de pessoal ([Wren & Wren, 1995](#); [Kotecha et al., 2004](#)), de máquinas ([Zhang et al., 2008](#); [Park, 2001](#)) e de processos ([Contreras et al., 2005](#)).

## 2. Descrição do Problema

O objeto deste estudo é o escalonamento integrado na agroindústria. Este problema está relacionado ao Escalonamento de Pessoal (*Personnel Scheduling*) integrado com escalonamento da produção e designação de veículos de transporte. O problema investigado surge de um caso real observado no setor da agroindústria, cujo insumo processado é oriundo de carga de aves vivas que não podem ser estocadas por muito tempo. A empresa investigada realiza o processamento de mais de um milhão de aves por semana, dividido em três plantas (fábricas) em diferentes localizações. Os caminhões com carga que chegam na fábrica devem aguardar num galpão ventilado, principalmente nos períodos de alta temperatura do verão. O desafio diário da empresa é realizar o escalonamento das equipes de apanha (mais de 100 pessoas no total) para manter as fábricas em constante funcionamento, atendendo às restrições operacionais e trabalhistas, além de minimizar os custos. A Figura 1 resume um exemplo de escalonamento com duas equipes de apanha e mostra a sequência de locais para as equipes e as designações (transportes) das cargas para três fábricas. A ilustração mostra duas sequências individuais, uma para cada equipe, embora em alguns momentos o mesmo local possa ser compartilhado por duas ou mais equipes, não necessariamente no mesmo tempo, pois cada equipe terá sua escala com horários possivelmente diferentes. Nota-se, também, que as escalas dos caminhões com as cargas têm sua origem e destino nas fábricas. As designações dos caminhões são para o transporte de cargas de cada granja para a fábrica. Uma granja pode ter várias cargas que podem ter destinos (fábricas) diferentes.

As aves devem ser coletadas em diferentes granjas, em locais espalhados geograficamente num raio de aproximadamente 150 Km de cada fábrica. A

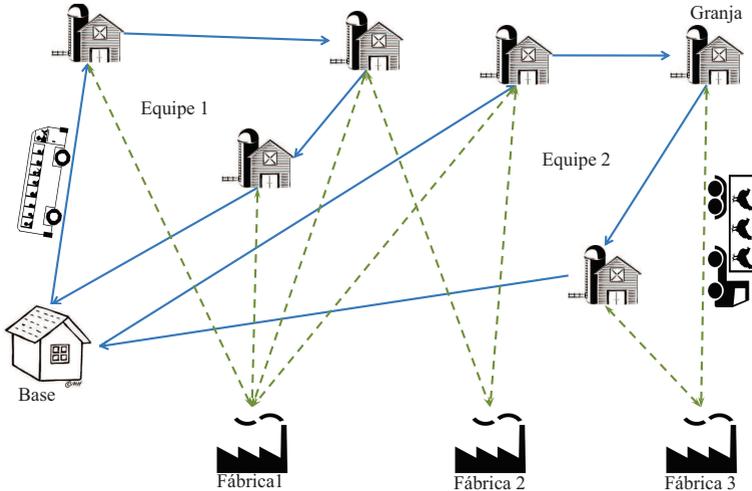


Figura 1. Ilustração do escalonamento de duas equipes de apanha das cargas para três fábricas. As linhas contínuas representam a sequência de locais visitados por cada equipe. As linhas tracejadas representam as designações de cargas das granjas para as fábricas.

cada dia novos locais de coleta das aves são definidos, implicando na necessidade de realizar escalonamentos diários. O pessoal que realiza a apanha das aves é dividido em várias equipes que podem ser designadas para diferentes locais durante o dia. Logo, o tempo de transporte entre os locais de apanha deve ser minimizado. As equipes têm sua base em um local diferente das fábricas. Algumas informações, tais como a sequência de locais de apanha, a hora de início e término das atividades, a hora do carregamento e descarregamento dos caminhões e o destino de cada carregamento (carga) devem ser descobertas com a definição do escalonamento.

Para obter este escalonamento, algumas regras devem ser levadas em consideração, como o acordo com a empresa, as regras trabalhistas e restrições operacionais. Além disto, há uma função objetivo a ser otimizada relacionada com custos operacionais.

Os custos operacionais incluem custos com horas extras para as equipes de apanha, custos com horas ociosas das equipes, custos com transporte e custos relacionados ao tempo de espera do caminhão no pátio da empresa. Considerando o alto custo da ociosidade da indústria, este custo é considerado como uma restrição do problema. Assim, a função objetivo se resume à minimização de custos com as equipes de apanha (horas pagas) e ao tempo de espera na fila (pátio da fábrica).

Para efeito de cálculo do tempo de trabalho pago às equipes, são considerados o tempo de viagem e o tempo gasto durante a apanha e carregamento dos caminhões. Além disto, algumas restrições (regras) devem ser atendidas na elaboração das escalas de trabalho. As regras a serem seguidas são as seguintes:

- O tempo normal de trabalho são 7 horas e 20 minutos (excluindo o tempo de descanso). O tempo que extrapolar é considerado hora extra, incidindo um acréscimo de 50% sobre o valor pago no tempo normal.
- Se o total de horas trabalhadas for superior a 4 horas e não exceder 6 horas, deverá ser adicionado um tempo de descanso de, no mínimo, 15 minutos.
- Se o total de horas trabalhadas for superior a 6 horas, deverá ser adicionado um descanso de no mínimo 1 hora.
- O descanso deve ser de, no máximo, 2 horas, pois acima deste tempo será considerado tempo pago. A hora extra também é contabilizada sobre o tempo mínimo de descanso caso não seja considerado na escala.

Além das restrições operacionais, existem alguns dados gerais que devem ser utilizados pelo algoritmo, tais como: localização de todas as granjas, localização da base das equipes de apanha, localização das fábricas, distância entre todos os locais, velocidade média dos caminhões, velocidade de abate de cada fábrica, hora de início de operação de cada fábrica, hora de término de operação de cada fábrica e capacidade de cada caminhão.

O escalonamento é realizado diariamente e os dados de entrada do algoritmo, específicos para cada dia, devem ser: as granjas onde serão coletadas as aves, e o número de aves de cada granja.

Com base nas informações apresentadas sobre o problema em estudo, a Tabela 1 resume os aspectos em comum e as diferenças do presente trabalho com o de [Hart et al. \(1999\)](#). Observa-se na tabela que o problema em estudo possui mais diferenças do que aspectos em comum. [Hart et al. \(1999\)](#) tem como objetivo minimizar as violações de restrições, enquanto que nesta investigação o objetivo é minimizar custos e tempo na fila, satisfazendo as restrições. O fato de haver uma flexibilidade para construção dos turnos de trabalho, com divisões de tarefas de uma hora de duração, significa que há mais possibilidades de combinações de tarefas, ampliando, assim, a complexidade computacional para resolver o problema. A limitação de espaço no pátio de espera da fábrica e o tempo de espera no pátio são duas características que estão relacionadas e que impõem tanto uma restrição ao problema como um objetivo a ser otimizado. Calcular o tempo de espera

no pátio (fila) é outro aspecto que também introduz um novo desafio para a resolução do problema.

Tabela 1. Comparação com o trabalho de [Hart et al. \(1999\)](#).

	<b>Problema investigado</b>	<b>Hart, Ross &amp; Nelson (1999)</b>
<b>Objetivo</b>	Minimizar os custos operacionais e tempo na fila	Minimizar o número de violações de restrições
<b>Abordagem</b>	Problema de Otimização	Problema de Satisfação de Restrição
<b>Número de fábricas</b>	Três fábricas em locais diferentes e com velocidades diferentes de abate	O número e localização não são informados.
<b>Base das equipes</b>	Uma localização diferente das fábricas	No mesmo local da fábrica
<b>Turno de trabalho</b>	Flexível, com divisões de tarefas (tempo de carregamento) em intervalos de uma hora	Os turnos são divididos em três tipos: manhã, tarde, flutuante
<b>Transporte das aves</b>	Terceirizado. Neste caso as escalas dos motoristas não são de responsabilidade da indústria	Frota própria, mas utiliza serviços de terceiros quando necessário
<b>Pátio de espera na fábrica</b>	Cada fábrica tem um número fixo de vagas em um galpão ventilado	Não consta essa informação
<b>Tempo de espera no pátio da fábrica</b>	Não há um limite de tempo máximo de espera *	Restrições governamentais estabelecem um limite de tempo

\* Não há limite de tempo máximo, porém é desejável que a carga permaneça em espera o mínimo de tempo possível para reduzir o custo operacional de manter a carga no pátio de espera e reduzir a perda de aves.

### 3. Modelagem Matemática

Como resultado da divisão do número de aves de uma granja pela capacidade de transporte de cada veículo, tem-se o que é denominado de “carga”. Cada carga está associada as seguintes informações:

1. Local de origem (granja);
2. Local de destino (fábrica);
3. Hora de início do carregamento;
4. Duração do carregamento;
5. Equipe de apanha responsável;
6. Número de aves.

Uma vez conhecido o conjunto de granjas e suas informações, como localização e número de aves, então o número de cargas, as suas origens e a duração para carregamento são obtidos automaticamente. Portanto, resta descobrir o destino (fábrica), a equipe responsável pelo carregamento e a hora de início do carregamento. Todos estes dados são associados a cada carga. Uma etapa fundamental para minimização dos custos é descobrir a associação das cargas às fábricas. Para isto foi criado um modelo de programação linear inteira binária descrito a seguir:

$$\text{Min} \sum_{g=1}^{ng} \sum_{f=1}^{nf} d_{gf} x_{gf} \quad (1)$$

$$\text{Sujeito a :} \quad \sum_{f=1}^{nf} x_{gf} = 1; \quad g = 1, \dots, ng; \quad (2)$$

$$\sum_{g=1}^{ng} s_g x_{gf} \leq cp_f; \quad f = 1, \dots, nf \quad (3)$$

$$x_{gf} \in \{0, 1\}, \quad g = 1, \dots, ng; \quad f = 1, \dots, nf$$

onde  $ng$  é o número de cargas,  $nf$  é o número de fábricas,  $d_{gf}$  é a distância da granja que originou a carga  $g$  ( $g = 1, \dots, ng$ ) até a fábrica  $f$  ( $f = 1, \dots, nf$ ),  $s_g$  é o número de aves da carga  $g$  e  $cp_f$  é a capacidade de processamento (em número de aves) da fábrica  $f$ . Por fim,  $x_{gf}$  é a variável de decisão, sendo que  $x_{gf} = 1$  se a carga  $g$  for alocada à fábrica  $f$ , sendo  $nf$  o número de fábricas. A restrição 2 garante que apenas uma carga seja alocada a uma única fábrica, enquanto que a restrição 3 garante que a capacidade da fábrica seja respeitada. A função objetivo dada pela Equação 1 procura minimizar a soma total das distâncias relacionadas com o transporte das aves.

Este modelo de programação matemática resolve apenas uma parte do problema global, que diz respeito ao destino das cargas de forma a minimizar custos de transporte. A geração de escala de trabalho por modelos de programação matemática ainda é um desafio para os pesquisadores da área (Bodin, 1983), o que tem justificado o uso de algoritmos heurísticos.

Neste problema em estudo foram identificados dois aspectos que dificultam a formulação de um modelo geral de programação matemática para o problema global:

1. Conceder período de descanso conforme as regras trabalhistas previamente definidas. Neste caso em particular, sempre que possível o descanso deve ser durante o tempo de transporte entre granjas ou da granja para sua base. Esta característica surge neste contexto pelo fato do tempo de transporte ser contabilizado como tempo de

trabalho, fato que não ocorre em escalonamento de pessoal em local fixo como, por exemplo, escalonamento de tele-atendentes.

2. A função objetivo deve considerar o tamanho médio das filas de caminhões nos pátios das fábricas. O uso de simulação tem sido uma alternativa para superar esta dificuldade de introduzir cálculos estatísticos de filas em modelos de programação matemática. Esta abordagem pode ser observada em [Lin et al. \(2000\)](#), [Harrison & Zeevi \(2005\)](#) e [Atlason et al. \(2008\)](#), os quais utilizam a combinação de programação linear e simulação na resolução de problemas de escalonamento no setor de tele-atendimentos.

Por estas dificuldades de modelagem matemática do problema, a implementação de um algoritmo heurístico passa a ser uma alternativa interessante. Neste trabalho é proposto um algoritmo heurístico que utiliza o modelo de designação linear em duas fases: fase de construção da solução e fase de melhoramento (busca local). A formulação do problema de designação (PD), denotada por  $PD([c_{ij}])$ , é a seguinte:

$$z = \text{Min} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} c_{ij} x_{ij} \quad (4)$$

$$\text{Sujeito a : } \sum_{i=1}^{n_1} x_{ij} = 1, \forall j = 1, \dots, n_2 \quad (5)$$

$$\sum_{j=1}^{n_2} x_{ij} = 1, \forall i = 1, \dots, n_1 \quad (6)$$

$$x_{ij} \in \{0, 1\}, \forall i = 1, \dots, n_1, \forall j = 1, \dots, n_2$$

A variável de decisão  $x_{ij}$  é binária  $\{0, 1\}$ , sendo que  $x_{ij} = 1$  se o item  $i$  for associado com o item  $j$ . Caso contrário,  $x_{ij} = 0$ , e  $c_{ij}$  é o custo de associar o item  $i$  ao item  $j$ . Os itens podem ter diferentes significados de acordo com a fase de resolução (construção e melhoramento), apresentadas nas seções seguintes. O valor de  $n_1$  deve ser igual a  $n_2$ . Porém, na fase de construção da solução  $n_1$  representa o número de equipes (reais e fictícias) e  $n_2$  o número de cargas  $ng$ . Considerando que o número de cargas é sempre superior ao número de equipes, são criadas equipes fictícias para tornar a matriz  $[c_{ij}]$  quadrada. Na fase de melhoramento,  $n_1$  e  $n_2$  serão sempre o número máximo de equipes ( $ne_{max}$ ) permitidas para o escalonamento. Embora o algoritmo utilize o número máximo de equipes disponíveis, deve-se observar que a abordagem procura automaticamente minimizar o número de equipes, tendo em vista que o objetivo é minimizar o custo global com pessoal.

## 4. O Algoritmo Proposto

O método proposto é um algoritmo evolutivo que utiliza alguns procedimentos construtivos e de busca local (melhoramento). Nesta seção serão apresentadas a forma de representação da solução, os procedimentos construtivos, o procedimento de busca local e o procedimento geral baseado em algoritmos genéticos.

Uma escala de trabalho para uma equipe é uma sequência de atividades a ser executada ao longo do dia. Porém, o tempo de carregamento de um caminhão é de aproximadamente uma hora. Portanto, a escala de trabalho pode ser discretizada em uma sequência de atividades com duração de uma hora. Cada atividade está relacionada com uma carga definida anteriormente. Para efeito de simplificação do problema, a duração do carregamento de todas as cargas será considerado o intervalo de uma hora. Assim, considere  $t$  o índice de tempo ao longo do dia de trabalho, sendo  $t = 1, \dots, 24$ , onde  $t = 1$  significa o primeiro horário do dia em que alguma equipe inicia o trabalho de carregamento.

### 4.1 Procedimento construtivo

Dado um número máximo de equipes  $ne_{max}$  pré-definido, a fase de construção da solução inicial parte do princípio de que a sequência de atividades (cargas) de cada equipe ainda está vazia e, então, as cargas são designadas sucessivamente até que todas as cargas sejam atribuídas para o número máximo de equipes. A ideia do algoritmo é atribuir as cargas para as equipes considerando que todos os carregamentos iniciam em  $t = 1$ . Assim, em  $t = 1$  todas as equipes recebem a designação de uma carga selecionada aleatoriamente. Para os demais intervalos de tempo  $t$  ( $t > 1$ ) a designação é resolvida utilizando um problema de designação, o qual está associado a uma matriz de custo  $c_{ij}$ , de dimensão  $ng$ , relacionada ao custo de associar uma carga  $j$  a uma equipe  $i$ . Como o número de cargas é sempre maior do que o número de equipes ( $ne_{max}$ ), então a matriz é dividida em dois blocos e assume valores conforme definido na Figura 2.

		Cargas
Equipes		<i>Bloco I</i> $c_{ij} = fa(i, j)$
Equipes Fictícias		<i>Bloco II</i> $c_{ij} = \infty$

Figura 2. Estrutura da matriz de custo para a fase de construção da solução.

Onde os custos do bloco I da matriz são calculados pela seguinte função:

$$fa(i, j) = hp + ph, \quad (7)$$

sendo  $hp$  as horas pagas para a equipe  $i$  ao ser designada ao trabalho de carregamento da carga  $j$ , e  $ph$  uma penalidade caso alguma restrição não seja atendida. Para o cálculo das horas pagas são levadas em considerações as regras apresentadas na Seção 2. Como penalidade foi considerado o limite de 9 horas de trabalho contínuo. O algoritmo do procedimento de construção é apresentado na Figura 3.

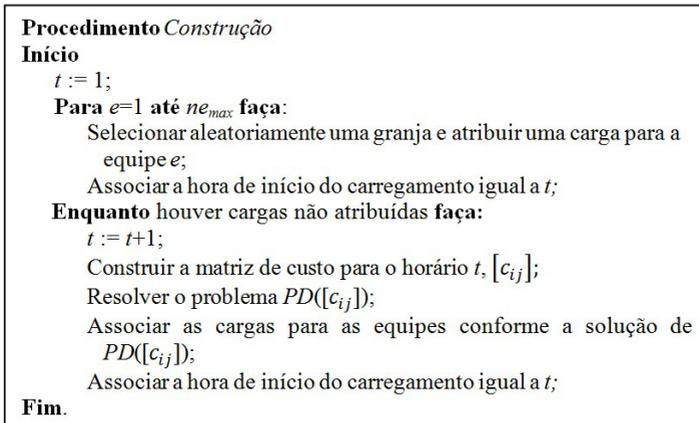


Figura 3. Procedimento de construção.

## 4.2 Busca local baseada em custo

A busca local tem o objetivo de realizar pesquisa em torno de uma solução corrente (algumas vezes chamada de busca na vizinhança) na tentativa de encontrar soluções aprimoradas. Neste trabalho propõe-se dois procedimentos de busca que são baseados em VND (*Variable Neighborhood Descent*), que é uma particularidade do VNS (*Variable Neighborhood Search*) (Hansen & Mladenović, 2001).

Um procedimento de busca local, denominado *k-swap*, consiste em selecionar  $k$  intervalos de tempo consecutivos iniciando no tempo  $t$ ; para cada equipe são agrupadas as atividades associadas neste intervalo gerando os “blocos de atividades” denotados por  $ba(k, t)$ . O *k-swap* realiza a investigação de troca de blocos de atividades destes intervalos entre as escalas das equipes.

A investigação de busca do *k-swap* se dá pela resolução de sucessivos problemas de designação. Cada investigação requer uma matriz de custo  $[c_{ij}]$ , de dimensão  $ne_{max}$ , que representa o custo de associar um bloco de atividades  $i$  com a escala da equipe  $j$ . A Figura 4 ilustra a estrutura da matriz de custo para cada problema de designação utilizado pelo *k-swap*. Nota-se que, diferente da estrutura da matriz do procedimento de construção, agora a matriz de custo trata da relação de custos em associar escalas ou parte das escalas, enquanto que a matriz anterior relacionava o custo de associar cargas às equipes.

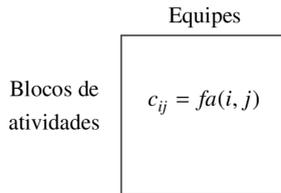


Figura 4. Estrutura da matriz de custo para o *k-swap* que se diferencia da estrutura da matriz do procedimento de construção da solução.

Considere  $ti$  o índice da hora iniciada mais cedo pelas escalas das equipes e  $tf$  o índice da hora finalizada mais tardiamente pelas escalas das equipes. A descrição do algoritmo do *k-swap* é dada pela Figura 5.

```

Procedimento k-Swap
Início
     $z^* = \infty$ ;
    Para  $t = (ti + 1)$  até  $(tf - k - 1)$  faça:
        Construir a matriz de custo  $[c_{ij}]$  para o bloco  $ba(k,t)$ ;
        Resolver o problema  $PD([c_{ij}])$ ;
        Se  $z < z^*$  então
             $z^* := z$ ;
             $t^* := t$ ;
        Associar os blocos para as equipes conforme a solução de  $PD([c_{ij}])$  de  $t^*$ ;
Fim.
    
```

Figura 5. Procedimento de melhoria *k-swap*.

A Figura 6 ilustra um exemplo de *1-swap* para  $t = 4$  sobre um grafo representando as atividades a serem realizadas por 4 equipes num período de 7 horas, para o carregamento de 18 cargas. Os vértices rotulados representam as cargas, enquanto que os vértices vazios representam horários

sem carregamento (descanso ou viagem). As arestas contínuas representam a sequência de atividades previamente definidas, enquanto as arestas tracejadas são possibilidades de trocas que serão investigadas pelo  $k$ -swap.

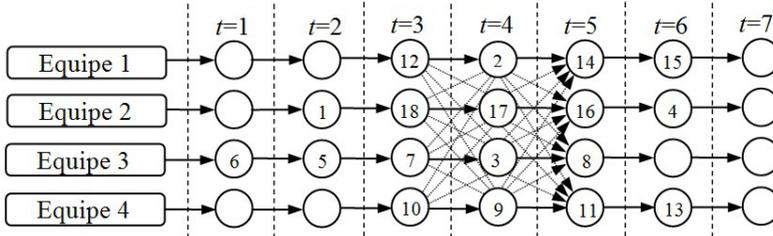


Figura 6. Ilustração de  $1$ -swap considerando somente um intervalo de tempo no horário  $t = 4$ .

A Figura 7 ilustra um exemplo de  $2$ -swap para os intervalos de tempo  $h = 4$  e 5.

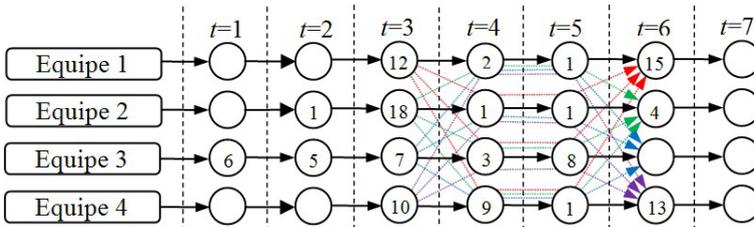


Figura 7. Ilustração de  $2$ -swap considerando dois intervalos de tempo consecutivos,  $t = 4$  e  $t = 5$ .

O segundo procedimento de busca é denominado de PCR (Procedimento de Corte e Recombinação) em que são realizados “cortes” nas escalas entre dois horários de trabalho, separando as sequências de atividades em duas partes. As partes são recombinadas utilizando o PD. A Figura 8 ilustra o procedimento PCR para um corte.

A matriz de custo  $[c_{ij}]$ , neste caso, representa o custo de associar a sequência de atividades  $i$  antes do corte com a sequência  $j$  depois do corte. O algoritmo que descreve PCR é apresentado na Figura 9.

O algoritmo geral de busca local, denominado  $bl$ -VND, é um algoritmo baseado na meta-heurística *Variable Neighbourhood Descent* (VND), que explora o espaço de soluções por sistemáticas mudanças da estrutura de vizinhança (Hansen & Mladenović, 2001). Seja  $R$  o conjunto de vizinhanças,

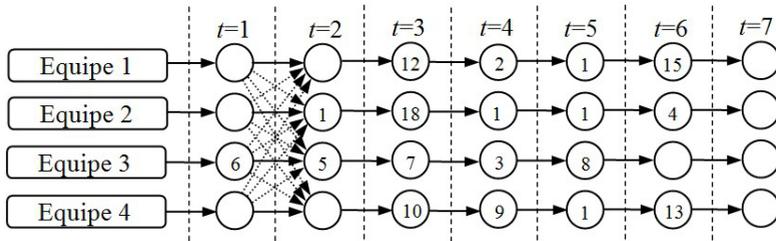


Figura 8. Ilustração do PCR sendo aplicado no corte entre o horário  $t = 1$  e  $t = 2$ .

**Procedimento PCR**

**Início**

$z^* = \infty$ ;

**Para**  $t = t_i$  **até**  $(t_f - 1)$  **faça**:

Construir a matriz de custo  $[c_{ij}]$  para o corte entre  $t$  e  $t+1$ ;

Resolver o problema  $PD([c_{ij}])$ ;

**Se**  $z < z^*$  **então**

$z^* := z$ ;

$t^* := t$ ;

Alocar as seqüências de atividades conforme a solução de  $PD([c_{ij}])$  de  $t^*$ ;

**Fim.**

Figura 9. Procedimento de melhoria PCR.

$N_1, N_2, \dots, N_R$ . Se a solução corrente não é melhorada usando uma particular vizinhança, então uma próxima vizinhança é explorada e assim por diante. Para o algoritmo em questão foram utilizadas  $R = 6$  vizinhanças, sendo  $N_1 = 1\text{-swap}$ ,  $N_2 = 2\text{-swap}$ , ...,  $N_5 = 5\text{-swap}$  e  $N_6 = \text{PCR}$ .

Em cada iteração do *bl-VND* todas as vizinhanças são exploradas e o algoritmo para quando nenhuma melhoria ocorre em uma das iterações. A avaliação de cada solução é derivada da Equação 7 que pode ser descrita como:

$$f1(s) = \sum_{i=1}^{n_{\max}} fa(i, j) \mid j \in SC(i), \quad (8)$$

onde  $s$  é a solução com as escalas das equipes,  $SC(i)$  é o conjunto de atividades da escala da equipe  $i$  e  $fa(i, j)$  é o custo da atividades  $j$  estar na escala da equipe  $i$ . A estrutura geral do algoritmo é apresentada na Figura 10. Quando não ocorrer melhoras na solução corrente depois de

passar por todas as  $R$  vizinhanças, então o algoritmo para (critério de parada).

**Procedimento** *bl-VND*;  
 Inicialização: Selecione um conjunto de estrutura de vizinhança  $N_r$  ( $r = 1, \dots, 6$ ); selecione uma solução inicial  $s$ ;  
**Início**  
**Repetir** até que o critério de parada seja satisfeito:  
 (1)  $r \leftarrow 1$ ;  
 (2) **Repetir** até que  $r = R$ :  
 (a) Busca local: Encontrar uma solução  $s'$  de  $s$  ( $s' \in N_r(s)$ );  
 (b) **Se**  $f1(s') < f1(s)$  **então**  $s \leftarrow s'$  e  $r \leftarrow 1$ ; **senão**  $r \leftarrow r+1$ ;  
**Fim.**

Figura 10. Procedimento geral de busca local utilizando todos os procedimentos de melhoria anteriores.

### 4.3 Busca local baseada em fila

Os procedimentos construtivos e de busca local apresentados têm como foco a minimização dos custos de horas pagas. Portanto, são avaliadas apenas as horas trabalhadas pelas equipes sem considerar o impacto no funcionamento de cada fábrica no que diz respeito ao tempo de espera das cargas no pátio das fábricas e tampouco a hora ociosa das fábricas. Com o intuito de considerar este custo, foi introduzida uma segunda função para avaliar uma solução definida como:

$$f2(s) = tq + po, \quad (9)$$

onde  $tq$  é o tempo médio na fila de espera das fábricas e  $po$  é uma penalidade caso ocorra alguma ociosidade das fábricas, que neste caso foi penalizada a ocorrência de ociosidade da fábrica.

Visando a minimização do tempo de espera no pátio de cada fábrica e a ociosidade das mesmas, foi projetado um segundo algoritmo de busca local, denominado *bl-Queue*, que realiza mudanças na hora de início das escalas de cada equipe. A hora de início da escala para cada equipe é adiantada ou atrasada de tal maneira que o custo  $f2$  seja minimizado. Isto pode ser feito avaliando o impacto de cada escala iniciando no tempo  $t, t = 1, \dots, t_{max}$ , sendo  $t_{max}$  o tempo máximo permitido para que a última carga da escala chegue na fábrica em tempo para descarregamento e abate. Este processo é repetido sequencialmente para cada equipe. O procedimento é apresentado na Figura 11.

Para calcular o valor de  $tq$  foi utilizada uma técnica de simulação discreta conhecida como método das 3-fases (Chwif & Medina, 2007). O

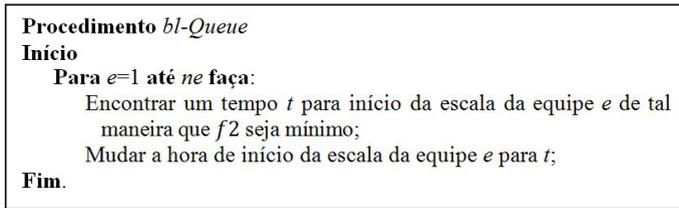


Figura 11. Procedimento de busca local baseado na informação da fila de espera na fábrica.

método considera dois tipos de eventos: condicional e programável. O evento programável é aquele que ocorre no tempo que estava sendo agendado para acontecer, mas o evento condicional é o tipo de evento que só ocorre quando as condições que ele requer forem satisfeitas. Todo evento condicional passa por uma fila de espera na qual permanece até que as condições sejam satisfeitas a fim de que seja executado (por exemplo, se a fábrica já iniciou o funcionamento ou se já é possível fazer o descarregamento de algum caminhão). Uma agenda é utilizada para colocar em ordem de ocorrência os eventos programáveis como também para atualizar o relógio da simulação. O método se resume nas três fases a seguir:

1. Checar o tempo de todos os eventos da agenda e selecionar o que ocorre primeiro. Atualizar o relógio da simulação para o tempo do evento selecionado.
2. O evento selecionado é executado e as entidades (cargas) são movidas para as respectivas filas de espera para a próxima atividade.
3. Pesquisar as entidades que estão na fila de espera e iniciar os eventos que satisfazem as condições do momento. Mover as entidades das filas para a atividade e calcular o tempo que a entidade permaneceu na fila.

Note-se que as próprias cargas, definidas anteriormente, podem ser interpretadas como as entidades que praticaram os eventos da simulação. Uma vez que se sabe a hora que as equipes iniciam suas escalas de trabalho, é possível saber qual a hora que cada carga chegará à fábrica de destino. Esta informação é utilizada para calcular o tempo que cada entidade (carga) permaneceu na fila de espera. Ao final da simulação se obtém as estatísticas das filas, como por exemplo, o tempo de permanência e o tamanho médio das filas.

#### 4.4 O algoritmo geral

Nesta investigação foi proposto um algoritmo evolutivo, baseado em algoritmos genéticos, combinado com busca local. A abordagem por algoritmos

genéticos foi escolhida pela facilidade de tratar problemas complexos onde há dificuldades para a modelagem matemática e definição de um procedimento de busca de soluções. O problema investigado é altamente restrito e o uso de algoritmos genéticos tem sido indicado para estas situações (Podgorelec & Kokol, 1997).

O algoritmo é dividido em duas fases executadas consecutivamente, onde cada fase compreende um algoritmo genético. Porém, cada fase utiliza uma busca local diferente. Na fase 1 o algoritmo procura gerar as escalas de tal maneira que as horas pagas sejam minimizadas. Por outro lado, na fase 2 o algoritmo modifica a hora de início das escalas de cada equipe de tal maneira que o tempo médio na fila de espera da fábrica seja minimizado. Portanto, a primeira fase utiliza uma função de avaliação  $f_1$  (Equação 8) baseada em horas pagas, enquanto que a segunda fase utiliza  $f_1 + f_2$  (Equações 8 e 9, respectivamente). Antes de iniciar a execução das fases 1 e 2, um modelo de programação linear inteira é resolvido para identificar a designação das cargas às fábricas minimizando custos de transporte e respeitando a capacidade de processamento de cada fábrica.

Cada indivíduo  $s$  (solução do problema) é codificado por um cromossomo que representa o conjunto de cargas, implementada em uma lista. O objetivo do algoritmo é encontrar a hora de início de carregamento de cada carga e atribuí-la a uma equipe.

A população inicial é gerada aleatoriamente com auxílio do procedimento Contrução apresentado na Seção 4.1. A população é atualizada com base na estratégia *steady state* (algumas vezes denominada de estacionária ou substituição incremental) para cada indivíduo gerado. O indivíduo substituirá aleatoriamente um membro da população com uma avaliação abaixo da média. Cada indivíduo gerado significa uma iteração do algoritmo. O número máximo de iterações é definido pelo parâmetro *MaxIt*. O uso da abordagem *steady state* tem sido utilizada para resolver muitos problemas de otimização, em particular os problemas de escalonamento (Beasley & Chu, 1996; Jat & Yang, 2009; AlSharafat & AlSharafat, 2010). Segundo Beasley & Chu (1996), um algoritmo genético que utiliza esta abordagem tende a convergir mais rápido do que o método geracional.

A seleção dos indivíduos para reprodução é realizada pela técnica do torneio, selecionando dois indivíduos aleatoriamente e escolhendo o mais apto com base na função de avaliação de cada fase.

O operador de cruzamento seleciona dois indivíduos e os recombina utilizando a técnica do cruzamento uniforme das cargas. As designações das cargas às equipes são realizadas com o procedimento de construção. Os indivíduos selecionados (pais) sempre geram um único indivíduo (filho). Nesta pesquisa não foi utilizado o parâmetro de taxa de cruzamento porque foi observado que o problema investigado é altamente restritivo. Isto significa que nem sempre o cruzamento gera uma solução viável (atendendo as restrições). Portanto, todas as soluções inviáveis são descartadas.

Foi observado nos experimentos que, dependendo da instância, a taxa de soluções inviáveis poderiam chegar até a 40% dos indivíduos gerados. O uso de um operador de cruzamento híbrido com o procedimento de construção tem sido uma alternativa para reduzir esses casos de inviabilidade em problemas altamente restritivos (Burke et al., 1995).

A mutação é aplicada segundo uma taxa a ser definida como parâmetro do algoritmo. A mutação realiza a mudança da hora de início de duas cargas selecionadas aleatoriamente. Este operador genético tem uma função muito importante para introduzir diversidade de soluções, uma vez que a população inicial é criada para horários iniciando em  $t = 1$ .

Na primeira fase do algoritmo os indivíduos são avaliados com base nos custos das horas pagas, enquanto que na segunda fase avalia-se o tamanho das filas nas fábricas utilizando uma técnica de simulação discreta. Assim, após cada reprodução e mutação é aplicado o algoritmo de busca local *bl-VND* na fase 1 e *bl-Queue* na fase 2.

O critério de parada é um parâmetro a ser definido em termos de número de iterações. Quando o critério de parada é alcançado o algoritmo devolve o melhor indivíduo da população. O número de iterações é utilizado como critério de parada para cada fase. Porém, a fase 2 utiliza apenas um percentual  $\Omega$  do número de iterações da fase 1.

Cada fase do algoritmo segue os padrões clássicos dos algoritmos genéticos, mas a população é inicializada na fase 1, enquanto que a fase 2 continua operando sobre a população resultante da fase 1. A Figura 12 mostra uma visão geral do algoritmo proposto.

## 5. Resultados Computacionais

O algoritmo passou por várias experimentações piloto, as quais serviram para calibrar os parâmetros do próprio algoritmo e também para modificar algumas de suas características, de modo a chegar à versão final. Foram realizados extensivos testes computacionais e diferentes parâmetros foram testados de forma empírica. Para os resultados computacionais apresentados nesta seção, os parâmetros finais utilizados são descritos na Tabela 2.

Tabela 2. Parâmetros do algoritmo proposto

Parâmetro	Valor
Tamanho da população	60
Taxa de mutação	3%
Número de iterações – <i>MaxIt</i>	1000
$\Omega$	50%
<i>ph</i>	100
<i>po</i>	10.000

Os experimentos foram realizados num computador com CPU Intel I3, 2,26 GHz, com Windows® 7 64 bits, usando a linguagem Object Pascal com

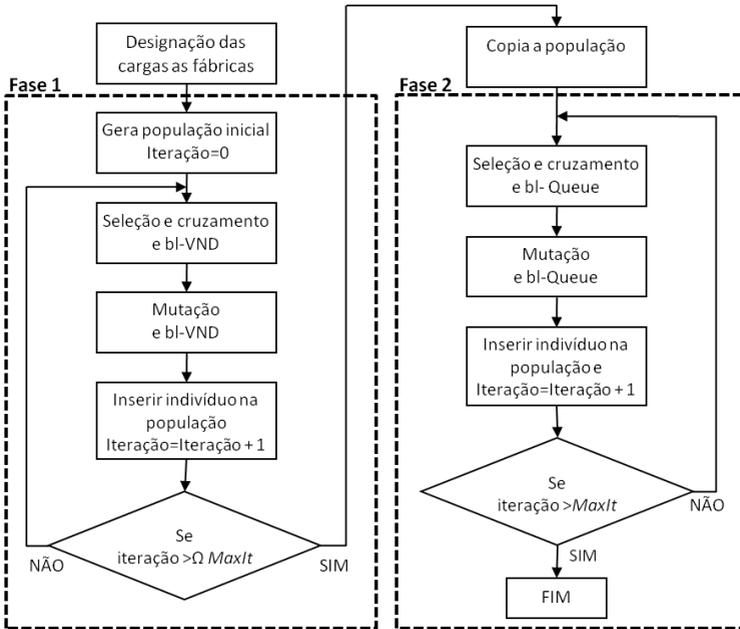


Figura 12. Fluxograma do algoritmo geral, incluindo as duas fases baseadas em algoritmos genéticos.

compilador FreePascal. Foram utilizadas escalas reais de uma empresa, obtidas ao longo de um período de 2 meses de operação, totalizando 45 dias. Envolveram um total de 112 catadores divididos em equipes de 14 pessoas (incluindo o motorista). A tabela 3 compara os valores obtidos pelo algoritmo com as escalas produzidas manualmente pela empresa (coluna Manual).

Os resultados mostram uma redução significativa dos custos operacionais. Os custos com horas pagas são relativos às equipes, o que significa que o custo reduzido deve ser multiplicado pelo número de pessoas envolvidas, neste caso 14 pessoas. A tabela também apresenta o número de escalas inviáveis que também contribui para os custos operacionais. Uma escala típica inviável planejada pela empresa ocorre quando o tempo real de deslocamento entre duas granjas não é contabilizado corretamente. Isto significa que certamente a escala não será executada dentro do tempo que havia sido planejado. Isto acarreta custos operacionais adicionais. Os resultados mostram uma redução de 100% destes casos. Do ponto de vista qualitativo foram observadas algumas eliminações de escalas indesejadas,

Tabela 3. Resultados computacionais com o algoritmo proposto.

	Manual	Algoritmo	Redução	% de redução
Total de horas pagas (horas)	3.290,60	2.938,18	352,42	10,71%
Total de distância percorrida (Km)	315.807,80	295.945,00	19.862,80	6,29%
Total de horas da fábrica parada (horas)	39,43	0,00	39,43	100,00%
Número médio de caminhões esperando no pátio	4,52	2,25	2,27	50,22%
Tempo médio de espera no pátio (minutos)	119,04	4,93	114,11	95,86%
Sobrecarga de caminhões no pátio *	64	0	64	100,00%
Número de escalas inviáveis	39	0	39	100,00%

\*Quantidade de vezes que o limite de caminhões no pátio da fábrica foi ultrapassado.

como, por exemplo, escalas com extensos períodos de trabalho sem um intervalo de descanso.

O total de horas ociosas da fábrica foi reduzido para zero. O custo de uma fábrica parada é geralmente muito alto, podendo chegar ao valor de US\$ 6.000,00 por hora parada.

Além de reduzir o número médio de caminhões no pátio da fábrica em aproximadamente 50%, o tempo médio de espera teve uma redução ainda mais acentuada, superior a 95%. Nota-se, ainda, uma redução de 100% dos casos de excesso de caminhões no pátio da fábrica. Nestes casos de excesso os caminhões ficam em um local impróprio, podendo comprometer a carga e aumentar os custos operacionais.

O tempo total de experimentação foi de aproximadamente 90 minutos para executar todas as instâncias do problema. Algumas instâncias consumiram aproximadamente 10 minutos para obtenção de uma solução.

## 6. Discussão e Conclusões

Neste trabalho foi investigado um algoritmo heurístico híbrido aplicado a um problema de escalonamento de atividades na agroindústria. Os ajustes no algoritmo e calibração dos parâmetros foram resultados de extensivas execuções piloto com dados de casos reais.

Os experimentos foram realizados com dados reais de uma empresa que trabalha com exportação de derivados de aves. Em média, o método possibilitou a redução aproximada dos seguintes aspectos: 11% dos custos

com horas pagas, 7% com custos de transporte, mais de 90% do tempo de espera no pátio do abatedouro e ainda eliminou os casos de ociosidade da fábrica. Além disto, foram observadas outras melhorias nas escalas das equipes do ponto de vista qualitativo. Embora não seja um objetivo explícito, o número de equipes também é minimizado automaticamente uma vez que o algoritmo procura reduzir custos com horas pagas.

Do ponto de vista do estado da arte, este projeto proporcionou a investigação de novos procedimentos aplicados ao escalonamento de atividades extensíveis a outros problemas similares. Algoritmos genéticos tiveram um importante papel neste projeto pela sua flexibilidade e facilidade de trabalhar com a resolução de um problema de difícil modelagem matemática.

Do ponto de vista da aplicabilidade, os experimentos realizados com o algoritmo proposto demonstraram a viabilidade de sua utilização num sistema computacional para automatizar a construção de escala de trabalho em situações reais. Além da agilidade na geração das escalas comparadas com o processo manual, notou-se uma redução significativa em termos quantitativos (custos) e qualitativos. Na incorporação do algoritmo num sistema computacional voltado para um gerente operacional da empresa, possivelmente os valores dos parâmetros do algoritmo deva ficar em segundo plano ou até mesmo ocultos.

Como investigações futuras pretende-se utilizar a abordagem geracional, juntamente com elitismo e utilizar uma taxa de mutação variável, conforme proposto por [Beasley & Chu \(1996\)](#), que aplica uma taxa de mutação mais elevada quando a avaliação (*fitness*) do melhor indivíduo fica próxima do pior indivíduo da população. Pesquisar por novos procedimentos de busca local e novos operadores genéticos também podem ser novas propostas de investigações futuras.

## Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro ao projeto.

## Referências

- AlSharafat, W. & AlSharafat, M., Adaptive steady state genetic algorithm for scheduling university exams. In: *Proceedings of International Conference on Networking and Information Technology (ICNIT)*. p. 70–74, 2010.
- Atlason, J.; Epelman, M. & Henderson, S., Optimizing call center staffing using simulation and analytic center cutting-plane methods. *Management Science*, 54(2):295–309, 2008.
- Bansal, S.; Gowtham, K. & Hota, C., Novel adaptive scheduling algorithm for computational grid. In: *Proceedings of the 3<sup>rd</sup> IEEE international*

- Conference on Internet Multimedia Services Architecture and Applications*. p. 308–312, 2009.
- Beasley, J. & Chu, P., A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2):392–404, 1996.
- Blazewicz, J.; Ecker, K.; Schmidt, G. & Weglarz, J., *Scheduling in Computer and Manufacturing Systems*. 2a edição. New York, USA: Springer-Verlag, 1994.
- Bodin, L., Solving large vehicle routing and scheduling problems in small core. In: *Proceedings of the 1983 Annual Conference on Computers: Extending the Human Resource*. p. 27–37, 1983.
- Burke, E.; Elliman, D. & Weare, R., A hybrid genetic algorithm for highly constrained timetabling problems. In: *Proceedings of the 6<sup>th</sup> International Conference on Genetic Algorithms*. p. 605–610, 1995.
- Chwif, L. & Medina, A., *Modelagem e Simulação de Eventos Discretos: Teoria e Aplicações*. 3a edição. São Paulo, SP: Edição dos Autores, 2007.
- Contreras, A.; Valero, C. & Pinninghoff, J., Applying genetic algorithms for production scheduling and resource allocation. special case: A small size manufacturing company. In: Ali, M. & Esposito, F. (Eds.), *Innovations in Applied Artificial Intelligence*. Berlin, Germany: Springer-Verlag, v. 3533 de *Lecture Notes in Computer Science*, p. 55–70, 2005.
- Hansen, P. & Mladenović, N., Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467, 2001.
- Harrison, J. & Zeevi, A., A method for staffing large call centers based on stochastic fluid models. *Manufacturing & Service Operations Management*, 7(1):20–36, 2005.
- Hart, E.; Ross, P. & Nelson, J., Scheduling chicken catching – an investigation into the success of a genetic algorithm on a real-world scheduling problem. *Annals of Operations Research*, 92(1):363–380, 1999.
- Jat, S. & Yang, S., A guided search genetic algorithm for the university course timetabling problem. In: *Proceedings of the 4<sup>th</sup> Multidisciplinary International Scheduling Conference: Theory and Applications*. Dublin, Ireland, p. 180–191, 2009.
- Kotecha, K.; Sanghani, G. & Gambhava, N., Genetic algorithm for airline crew scheduling problem using cost-based uniform crossover. In: Manandhar, S.; Austin, J.; Desai, U.; Oyanagi, Y. & Talukder, A. (Eds.), *Applied Computing*. Springer Berlin / Heidelberg, v. 3285 de *Lecture Notes in Computer Science*, p. 84–91, 2004.
- Lin, C.; Lai, K. & Hung, S., Development of a workforce management system for a customer hotline service. *Computers and Operations Research*, 27(10):987–1004, 2000.

- Park, Y., A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines. *International Journal of Production Economics*, 73(2):175–188, 2001.
- Podgorelec, V. & Kokol, P., Genetic algorithm based system for patient scheduling in highly constrained situations. *Journal of Medical Systems*, 21(6):417–427, 1997.
- Sheen, G.J. & Liao, L.W., Scheduling machine-dependent jobs to minimize lateness on machines with identical speed under availability constraints. *Computers and Operations Research*, 34(8):2266–2278, 2007.
- Sonmez, O. & Gursoy, A., A novel economic-based scheduling heuristic for computational grids. *International Journal of High Performance Computing Applications*, 21(1):21–29, 2007.
- Wren, A. & Wren, D., A genetic algorithm for public transport driver scheduling. *Computers and Operations Research*, 22(1):101–110, 1995.
- Yunes, T.; Moura, A. & de Souza, C., Solving very large crew scheduling problems to optimality. In: *Proceedings of the 2000 ACM Symposium on Applied Computing*. New York, USA: ACM, v. 1, p. 446–451, 2000.
- Zhang, F.; Cao, X. & Yang, D., Intelligent scheduling of public traffic vehicles based on a hybrid genetic algorithm. *Tsinghua Science & Technology*, 13(5):625–631, 2008.

---

## Notas Biográficas

**Ademir Aparecido Constantino** é graduado em Matemática e tem mestrado e doutorado em Engenharia de Produção na área de Otimização e Simulação, além de Pós-doutorado na Universidade de Nottingham, Inglaterra. Atualmente é professor titular do Departamento de Informática da Universidade Estadual de Maringá.

**Dário Landa-Silva** tem graduação e doutorado em Ciência da Computação. Atualmente é professor na Faculdade de Ciência da Computação na Universidade de Nottingham, Inglaterra.

**Wesley Romão** é graduado em Engenharia Química, tem mestrado em Ciência da Computação e doutorado em Engenharia de Produção. Atualmente é professor adjunto do departamento de Informática da Universidade Estadual de Maringá.

## Heurísticas Evolutivas Híbridas para o Problema de Escalonamento de Projetos com Restrição de Recursos Dinâmicos

André R. Villela da Silva\* e Luiz Satoru Ochi

---

**Resumo:** Este capítulo trata de um Problema de Escalonamento de Projetos utilizando recursos dinâmicos. Tais recursos são importantes para a modelagem de certas situações onde os recursos estáticos não são aplicáveis. Uma nova formulação matemática é proposta. Métodos heurísticos e híbridos são analisados e comparados com resultados da literatura. Os resultados computacionais obtidos mostram que os métodos híbridos são muito eficientes. Em algumas instâncias, estes métodos tiveram um desempenho melhor do que o otimizador CPLEX. Outras características como robustez e capacidade de evitar a convergência prematura também estão presentes nos métodos estudados.

**Palavras-chave:** Algoritmos evolutivos, Métodos híbridos, Formulação matemática, Representação de soluções.

**Abstract:** *This chapter deals with a Project Scheduling Problem using dynamic resources. This kind of resources are important for modeling certain situations where static resources are not applicable. A new mathematical formulation is proposed. Heuristic and hybrid methods are analyzed and compared with results in the literature. The computational results obtained show that hybrid methods are very effective. For some instances, these methods performed better than the CPLEX optimizer. Other features like robustness and capacity to avoid premature convergence are also present in the methods studied.*

**Keywords:** *Evolutionary algorithms, Hybrid methods, Mathematical formulations, Solutions representation.*

---

\* Autor para contato: [avillela@ic.uff.br](mailto:avillela@ic.uff.br)

## 1. O Problema de Escalonamento de Projetos - PEP

Nos Problemas de Escalonamento de Projetos (PEP) dois elementos são de fundamental importância: as tarefas, que constituem cada uma das etapas do projeto a ser executado, e os recursos, que são os insumos necessários para que uma tarefa seja executada. As tarefas estão conectadas entre si através de relações de precedência (do tipo *finish-to-start*) que determinam a ordem em que as tarefas podem ou não ser executadas. Também é muito comum que uma tarefa possa ter mais de uma predecessora. Neste caso, todas as predecessoras precisam ter sido executadas antes da tarefa em questão começar. O objetivo mais comum é fazer com que todas as tarefas do projeto sejam executadas o mais rapidamente possível, respeitando as restrições de precedência e de utilização dos recursos.

Os recursos que são necessários para a execução das tarefas são o outro elemento a ser administrado no PEP. Uma classificação bastante tradicional os divide em dois grupos: recursos renováveis e recursos não-renováveis. Os recursos renováveis são aqueles que, após ser utilizados na execução de uma tarefa do projeto, ficam novamente disponíveis para ser utilizados em outra tarefa ainda não executada. Alguns exemplos de recursos desta classe são as máquinas (escavadeiras, tratores, computadores) e os profissionais (engenheiros, programadores, assistentes). Estes recursos podem ser reutilizados ao final de uma etapa de projeto. Os recursos são classificados como não-renováveis se eles estiverem disponíveis uma única vez durante todo horizonte de tempo no qual deve ser tratado o problema. Uma vez que eles são utilizados (consumidos) não é mais possível contar com eles até o fim do problema. Exemplos mais comuns são combustíveis e dinheiro, entre outros.

Tradicionalmente, os PEPs não supõem a geração e, sim, o consumo dos recursos que são dados de entrada com valores pré-definidos uma vez que ou estes têm caráter não-renovável ou têm sua taxa de renovação bem definida pelo problema, como mostrado em [Valls et al. \(2008\)](#) e [Nonobe & Ibaraki \(2002\)](#). Estes cenários, no entanto, não são capazes de modelar certas situações onde, a partir do término da execução de uma etapa do projeto, esta passa a gerar recursos adicionais. Como forma de ilustrar estas situações, suponha que o projeto em questão seja a expansão comercial de uma companhia. Após a construção de uma nova filial, é razoável supor que ela possa contribuir financeiramente com a matriz por meio do lucro que se espera dela. O recurso que se apresenta mais importante neste caso, sem dúvida, é o retorno financeiro. A quantia, uma vez investida na construção da filial, não fica novamente disponível ao final desta etapa, ao contrário de uma máquina ou um trabalhador. O que ocorre é que após a finalização deste investimento, o mesmo passa a retornar novos recursos financeiros que poderão ser aplicados na execução de outras etapas:

abertura de novas filiais, contratação de pessoal, compra de equipamentos, entre outros.

No modelo que será estudado, o dinheiro é um recurso renovável, mas não como nos modelos tradicionais, onde há uma quantidade máxima disponível a cada instante de tempo. Neste modelo, um recurso é renovável pelo fato de poder ocorrer diminuição e aumento de sua quantidade disponível ao longo do tempo. A produção de recursos (dinheiro, no caso) permanece desde o término da construção da filial até o fim do horizonte de planejamento em questão.

Este capítulo aborda o PEP onde as tarefas consomem recursos ao serem ativadas e, a partir de então, passam a gerar recursos até o final do horizonte de planejamento, cujo tamanho é dado por um parâmetro de entrada  $H$ . O modelo pressupõe, ainda, uma quantidade inicial de recursos que poderão ser gastos nas primeiras ativações. O objetivo deste modelo é chegar ao final do horizonte de planejamento com a maior quantidade possível de recursos.

O recurso abordado neste caso não pode ter uma quantidade máxima definida, já que o objetivo citado perderia o sentido. Este recurso que é consumido e, posteriormente, produzido apresenta variações de quantidade ao longo do horizonte de planejamento que são difíceis de prever. Por isto este tipo de recurso é chamado de Recurso Dinâmico. Desta forma, o problema de escalonamento em questão será denominado Problema de Escalonamento de Projetos com Restrição de Recursos Dinâmicos – PE-PRRD.

O PEPRRD encontra aplicações potenciais em expansões comerciais ou industriais, como no citado exemplo da abertura de filiais que produzem lucro após sua implantação. Também é possível utilizar o PEPRRD em problemas de prestação de serviços ou provimento de infra-estrutura em regiões mais afastadas. Suponha, por exemplo, que se deseja expandir uma rede de fibra ótica para várias cidades do interior do estado ou do país. Não é possível, a princípio, fornecer um ponto de acesso em qualquer localidade por questões físicas, logísticas ou mesmo financeiras. A ampliação da rede deve ocorrer atendendo primeiramente as regiões onde o custo de implantação de um ponto de acesso é pequeno ou onde é possível conseguir maior lucratividade com a prestação de algum tipo de serviço (internet, telefonia, entre outros), desde que esta implantação seja tecnicamente viável, obviamente. Cria-se então uma situação de precedência entre as localidades que diferencia este problema de outros como os das  $p$ -medianas (Mladenović et al., 2007), por exemplo.

O objetivo principal deste trabalho é propor algoritmos e métodos para o PEP estudado – PEPRRD, que ofereçam melhores resultados aproximados ou mesmo encontrem a solução ótima para algumas instâncias da literatura. Embora o foco não esteja na construção de métodos exatos, será

analisada uma nova formulação matemática, na forma de um programa inteiro misto. O ponto central do trabalho fica, portanto, na discussão sobre algoritmos heurísticos e métodos híbridos que incorporem conjuntamente características das abordagens exata e heurística.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 traz a definição do problema e uma revisão da literatura existente. A Seção 3 mostra as formulações matemáticas para o problema. Algoritmos heurísticos são introduzidos na Seção 4. Métodos híbridos são o assunto da Seção 5. Na Seção 6 são apresentados os resultados computacionais. Por fim, a conclusão é apresentada na Seção 7.

## 2. O Problema de Escalonamento de Projetos com Restrição de Recursos Dinâmicos – PEPRRD

Antes de apresentar a definição do problema, é importante deixar claro alguns conceitos utilizados no funcionamento do modelo.

- **Recursos disponíveis:** são os recursos que podem ser aplicados, no instante de tempo  $t$ , na execução de tarefas. Denota-se por  $Q_t$ .
- **Custo de uma tarefa:** é a quantidade de recursos necessários para que uma tarefa possa ser executada. O custo de uma tarefa  $i$  é denotado por  $c_i$ .
- **Lucro de uma tarefa:** é a quantidade não-negativa de recursos que serão produzidos pela tarefa a partir do instante de tempo seguinte a sua ativação. É denotado por  $p_i$  para cada tarefa  $i$  do problema.
- **Lucro acumulado:** é a soma do lucro das tarefas ativadas até um instante de tempo  $t$ . A notação é feita por  $P_t$ .
- **Ativação de uma tarefa:** é a indicação de um tempo  $t$  no qual a tarefa deve ser executada. Neste instante de tempo deve haver recursos disponíveis em quantidade igual ou maior do que o custo da tarefa. Também é necessário que todas as tarefas predecessoras já estejam ativadas até o instante de tempo anterior ( $t - 1$ ).
- **Horizonte de planejamento:** é o conjunto de instantes de tempo nos quais as tarefas podem ser executadas. No PEPRRD, o tempo é discretizado em unidades de 1 até  $H$ , que é um dado de entrada do problema.

O PEPRRD é composto por um grafo acíclico direcionado  $G = (V, A)$ , onde  $V$  é um conjunto de vértices e  $A$  é o conjunto de arcos que unem os vértices. A cada tarefa  $i \in V$ , está associado um custo  $c_i$  e um lucro  $p_i$ , inteiros não-negativos. Inicialmente existe uma quantidade de recursos disponíveis  $Q_0 > 0$  e um lucro acumulado  $P_0 = 0$ . O escalonamento deve ser realizado durante um horizonte de planejamento composto por  $H$

unidades de tempo. O objetivo do problema é maximizar a quantidade de recursos (recursos disponíveis e lucro acumulado) ao final do horizonte de planejamento.

A Figura 1 apresenta um exemplo deste modelo de escalonamento, sendo resolvido por um algoritmo arbitrário. No exemplo, o horizonte de planejamento é composto por três unidades ( $H = 3$ ) e a quantidade inicial de recursos disponíveis  $Q_0 = 4$ . Por questões de simplificação, a quantidade de recursos disponíveis  $Q_t$  e o lucro acumulado  $P_t$  serão indicados por  $Q$  e  $P$ , respectivamente. As tarefas (1,2,3,4,5,6) têm custos (2,3,4,1,2,4) e lucros (1,2,4,2,3,5), respectivamente. As tarefas já ativadas estão em branco, as que estão disponíveis para ativação aparecem em cinza e as que ainda não podem ser ativadas são mostradas em preto.

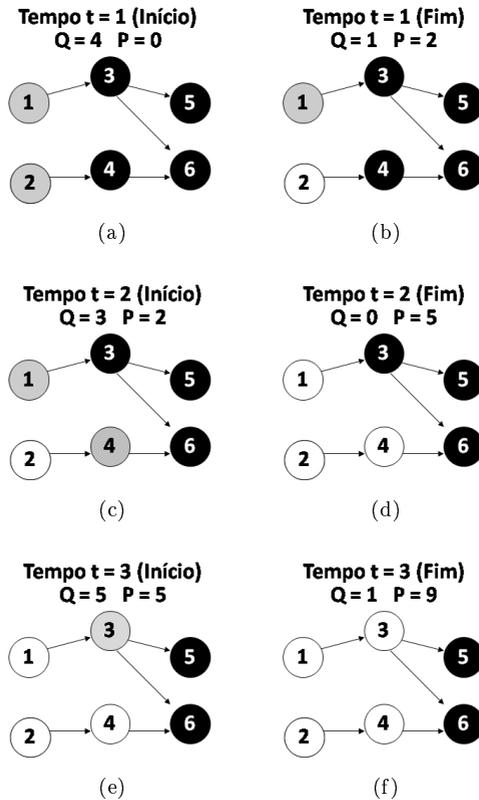


Figura 1. Etapas do exemplo de escalonamento.

No instante de tempo  $t = 1$ , Figura 1(a), as tarefas 1 e 2 estão disponíveis para ativação. Suponha que o algoritmo escolha a tarefa 2 para ser ativada. É necessário consumir 3 unidades de recursos, já que  $c_2 = 3$ . O lucro acumulado que inicialmente era zero passa a conter 2 unidades ( $p_2 = 2$ ). Com o recurso disponível restante não é possível ativar a tarefa 1, logo, o tempo  $t = 1$  deve ser encerrado como na Figura 1(b). Ao se passar para o instante seguinte ( $t = 2$ , Figura 1(c)) este lucro acumulado torna-se recurso disponível, uma vez que ele representa os recursos que estão sendo produzidos pela tarefa já ativada. Com a ativação da tarefa 2 realizada, a tarefa 4 fica apta a ser ativada. A quantidade de recursos disponíveis, agora, permite que ambas as tarefas 1 e 4 sejam ativadas. Esta decisão faz com que sejam subtraídas 3 unidades de recursos e sejam incorporadas mais 3 unidades ao lucro acumulado, como na Figura 1(d). Como não há mais tarefas para serem ativadas neste momento, é necessário passar para o instante de tempo seguinte: o lucro acumulado é somado aos recursos disponíveis e o estado das tarefas é atualizado. No início do tempo  $t = 3$  (Figura 1(e)), o último instante de tempo, a tarefa 3 pode ser ativada já que há recursos suficientes. Atualizados os recursos disponíveis, o lucro acumulado e o estado das tarefas, restou uma unidade de recurso disponível e o lucro acumulado é de nove unidades. O resultado deste escalonamento, portanto, é igual a dez unidades de recurso.

## 2.1 Revisão da literatura

O problema de escalonamento de projetos já é estudado há bastante tempo e apresenta diversos modelos que podem ser classificados de acordo com alguns critérios como o número de projetos existentes em cada instância, as possíveis formas de se executar cada tarefa do projeto, a quantidade e tipos de recursos a serem utilizados, entre outros. Um primeiro trabalho que apresenta estas classificações e suas peculiaridades é apresentado em Blazewicz et al. (1983).

Recentemente, uma abordagem heurística para o Problema de Escalonamento de Projetos com Restrição de Recursos que vem apresentando bons resultados é através da técnica denominada *random-keys* que pode ser encontrada em Mendes et al. (2009) e Snyder & Daskin (2006). Esta técnica trabalha com as tarefas através de prioridades que são atribuídas a elas ao invés de manipulá-las diretamente. Desta forma, é possível explorar melhor o espaço de soluções do problema de forma bastante eficiente.

Em Silva & Ochi (2006) é proposto o PEPRRD e apresentados os primeiros algoritmos para resolvê-los. Naquele trabalho, uma heurística construtiva chamada de ADDR gera, a cada instante de tempo, uma lista contendo todas as tarefas disponíveis neste instante. A lista é ordenada de acordo com a relação custo/lucro. Um elemento é escolhido aleatoriamente para ativação, se ainda houver recursos suficientes para tanto. A quantidade de recursos disponíveis e o lucro acumulado são atualizados;

o algoritmo prossegue tentando ativar mais tarefas até que não haja mais recursos ou tarefas disponíveis. Quando isto ocorre, passa-se para o próximo instante de tempo. Estes procedimentos são repetidos até se analisar o último instante de tempo.

Ainda neste primeiro trabalho foram apresentadas duas buscas locais: LS1 e LS2. Elas recebem um escalonamento já realizado e tentam localizar e remover tarefas ou grupos delas que foram ativadas em vão, ou seja, tarefas não-lucrativas que foram ativadas por causa de possíveis sucessoras mais lucrativas. O algoritmo construtivo e as buscas locais serviram de base para a elaboração de dois algoritmos evolutivos chamados de EA1 e EA2. Ambos operam sobre uma população de tamanho fixo. Os pais são escolhidos de forma a favorecer indivíduos com maior aptidão. Cada tarefa do filho gerado será ativada de acordo com o menor tempo em que ela foi ativada em cada um dos pais. Apenas os melhores indivíduos (entre pais e filhos) permanecem para a próxima geração.

A forma de representação da solução utilizada é a representação direta. Nela, uma solução é dada por um vetor de  $n$  inteiros não-negativos, onde  $n$  é o número de tarefas da instância. Cada valor indica o tempo de ativação de uma tarefa e o valor zero indica que a tarefa não foi ativada.

Outra busca local (LS3) foi empregada em [Silva & Ochi \(2007\)](#), como parte de um novo algoritmo evolutivo (EA3). Esta busca fixa uma parte da solução e reconstrói o restante com um critério puramente guloso. Desta vez, além dos operadores evolutivos básicos, foram propostos outros mecanismos para tentar evitar a convergência prematura. Os mecanismos de intensificação e diversificação mais específicos começam a agir quando a população passa algumas gerações sem aprimorar a melhor solução encontrada. Num primeiro momento, a melhor solução encontrada serve de semente para a geração de indivíduos bastante semelhantes, através da LS3. Esta fase funciona como uma intensificação ao redor desta solução. Se melhorias não ocorrerem, porém, e mais algumas gerações se sucederem sem aprimoramentos na melhor solução, toda a população será descartada e uma nova população será criada a partir do algoritmo construtivo. O operador evolutivo de cruzamento e a seleção natural são idênticos aos propostos em [Silva & Ochi \(2006\)](#). Uma dificuldade que não foi bem superada encontra-se na geração dos filhos. O algoritmo de cruzamento adotado pelo EA3 – o mesmo dos primeiros algoritmos evolutivos – gasta muito tempo analisando a viabilidade dos filhos que são gerados, uma vez que a escolha dos menores tempos de ativação para as tarefas, como é feita, não assegura que os filhos gerados sejam soluções válidas, sem a execução de algoritmos de correção.

Infelizmente, existem poucos trabalhos referentes ao PEPRRD. Não é do conhecimento destes autores a existência de outras abordagens para o problema em questão que permita uma comparação mais detalhada. Vale lembrar que os PEPs tradicionais não podem ser facilmente adaptados para

o modelo estudado aqui, pois eles não admitem a geração de recursos, o que vem a ser um componente fundamental do PEPRRD.

### 3. Formulações Matemáticas Para o PEPRRD

Em [Silva & Ochi \(2006\)](#), foi proposta uma modelagem matemática (F1), na forma de um programa inteiro misto, para o PEPRRD. A formulação foi muito útil para que instâncias pequenas fossem resolvidas até a sua otimalidade e os resultados fossem comparados com os métodos heurísticos apresentados naquele trabalho. Instâncias de maior porte ainda demoravam muitas horas, mesmo utilizando versões recentes de otimizadores.

No entanto, um ponto negativo na formulação original é a possibilidade das variáveis de uma tarefa apresentarem valores que podem aumentar ou diminuir de um instante de tempo para outro (quando uma tarefa é ativada o valor da variável daquele instante passa a ser igual a um, depois este valor volta a ser zero nas variáveis dos tempos seguintes). Para determinar se a tarefa está ativada é preciso fazer um somatório envolvendo todas as variáveis relativas a tempos anteriores.

Tentando resolver esta situação, vale a pena recordar que o PEPRRD pressupõe que uma tarefa permanecerá ativada até o final do horizonte de planejamento. Desta forma, pode-se mudar a semântica da variável binária para que ela indique não somente *quando* uma tarefa foi ativada, mas *se* ela já foi ativada ou não. Usando este raciocínio, foi desenvolvida uma nova formulação (F2) que pretende manter o estado de ativação de uma tarefa em todas as variáveis binárias a partir de um momento escolhido para tanto. A formulação F2 é mostrada a seguir.

$$(F2) \quad \text{Max } (Q_H + P_H) \quad (1)$$

Sujeito à

$$y_{it} \leq y_{it+1} \quad \forall i = 1, \dots, n \quad \forall t = 1, \dots, H - 1 \quad (2)$$

$$y_{it} \leq y_{jt-1} \quad \forall i = 1, \dots, n \quad \forall t = 1, \dots, H \quad \forall j \in \text{Pred}(i) \quad (3)$$

$$Q_t = Q_{t-1} + P_{t-1} - \sum_{i=1}^n c_i (y_{it} - y_{it-1}) \quad \forall t = 1, \dots, H \quad (4)$$

$$P_t = \sum_{i=1}^n p_i y_{it} \quad \forall t = 0, \dots, H \quad (5)$$

$$y_{i0} = 0 \quad \forall i = 1, \dots, n \quad (6)$$

$$y_{it} \in \{0, 1\} \quad \forall i = 1, \dots, n \quad \forall t = 1, \dots, H \quad (7)$$

$$Q_t, P_t \in N \quad \forall t = 0, \dots, H \quad (8)$$

A variável binária  $y_{it}$  indica se a tarefa  $i$  está ativada no tempo  $t$  (valor 1) ou não (valor 0). A partir do momento em que a tarefa é ativada, as

variáveis binárias referentes aos tempos seguintes também serão fixadas em 1, através das restrições (2). As restrições (3) garantem que uma tarefa  $i$  só poderá ser ativada caso todas as predecessoras  $j$  já estejam ativadas anteriormente. As restrições (4) mostram como é definida a quantidade recursos ao final de cada instante de tempo  $t$ . O cálculo do lucro acumulado é feito pelas restrições (5), indicando que este valor é dado pela soma dos lucros de todas as tarefas já ativadas até o momento. As restrições (6) indicam que no início do problema nenhuma tarefa está ativada. As restrições (7) e (8) definem o domínio das variáveis.

A grande vantagem desta formulação é que ela é constituída na sua quase totalidade de restrições que são aritmeticamente muito simples e deixam a matriz de coeficientes muito esparsa. Esta característica permite que o otimizador descarte rapidamente as restrições redundantes e economize bastante memória. Além disto, a busca por uma base ótima no método Simplex pode ser mais eficiente.

A utilização de uma formulação matemática e de um otimizador constitui uma das possíveis abordagens para se resolver um problema computacional, especialmente quando se trata de um problema de grande dificuldade como o que está sendo estudado. Se for dado tempo suficiente, o otimizador conseguirá encontrar pelo menos uma solução que tenha o valor ótimo. Na maioria dos casos, porém, não é possível esperar dias ou semanas para se obter a solução, mesmo que a formulação produzida seja a mais eficiente possível. Uma estratégia para não se descartar completamente desta abordagem é construir algoritmos chamados de híbridos. Estes algoritmos terão componentes provenientes da abordagem exata (geralmente por formulação matemática) e da abordagem heurística. Algoritmos híbridos para o PEPRRD serão apresentados posteriormente neste capítulo, mas é possível adiantar que uma boa formulação matemática, que possa ser executada de forma bastante eficiente pelo otimizador, é crucial para que os algoritmos híbridos tenham bom desempenho.

#### 4. Métodos heurísticos para o PEPRRD

Como o problema tratado neste trabalho é considerado difícil de ser resolvido em tempo computacional razoável (NP-árduo), é justificada a abordagem por métodos heurísticos. Tais métodos devem ter o compromisso de apresentar soluções de boa qualidade em um curto espaço de tempo, mesmo que, para isto, tenham que sacrificar a garantia da otimalidade.

Os Algoritmos Evolutivos são desdobramentos dos Algoritmos Genéticos clássicos propostos em Holland (1975), que apresentam soluções codificadas por valores binários e operadores genéticos simples. Embora o objetivo de manter bons padrões (genes) em várias soluções simultâneas (indivíduos) seja o mesmo, os Algoritmos Evolutivos introduzem novos operadores de recombinação de soluções, buscas locais e refinamentos para

fazer com que o aproveitamento dos padrões seja potencializado. Outras meta-heurísticas evolutivas também utilizam princípios semelhantes (Damak et al., 2009).

Normalmente os Algoritmos Genéticos ou Evolutivos operam sobre dezenas ou centenas de indivíduos simultaneamente, o que pode trazer dificuldades se os operadores não forem eficientes o bastante ou permitirem a geração de soluções infactíveis que precisem de correção para que possam ser consideradas soluções viáveis para o problema.

Os Algoritmos Evolutivos para o PEPRRD apresentam um maneira de representar a solução chamada de Representação Direta (Silva & Ochi, 2007). Nesta representação, cada elemento  $i$  de um vetor indica o tempo em que a  $i$ -ésima tarefa deve ser ativada. Embora esta representação seja bastante objetiva, a principal dificuldade em se trabalhar com ela é que alterações nos seus valores frequentemente resultam em soluções infactíveis. Os cálculos necessários para que estas infactibilidades não existam são computacionalmente tão custosos que também não valem a pena ser executados rotineiramente.

Uma das propostas deste trabalho é empregar uma nova forma de representação chamada de Representação Indireta, pois agora, a cada tarefa, será dada uma prioridade de ativação que deve ser administrada por um algoritmo de escalonamento diferente. As tarefas com maior prioridade deverão ser ativadas primeiro, desde que respeitem as restrições de precedência e de recursos disponíveis. Vale destacar que, para esta representação, quaisquer valores reais (ou mesmo inteiros) podem gerar uma solução viável, desde que o algoritmo de escalonamento respeite as restrições do problema. Neste contexto, o novo algoritmo de escalonamento verifica, a cada instante de tempo, quais tarefas estão disponíveis e as ordena de acordo com a prioridade. Após analisar todas as tarefas disponíveis e realizar as ativações possíveis, o algoritmo passa para a unidade de tempo seguinte.

#### 4.1 Algoritmo evolutivo com prioridades – EA\_priority

A principal característica deste algoritmo, proposto em Silva et al. (2008), é o uso de prioridades na representação de uma solução. Como a representação da solução é diferente daquela utilizada nos trabalhos já citados, assim também deverá ser a forma de se operar sobre ela. Portanto, novos operadores evolutivos precisam ser definidos para que o potencial da representação seja alcançado. A seguir serão apresentados os operadores utilizados. Cada parâmetro, tais como o tamanho da população inicial, a chance de mutação e o número de filhos gerados, foram definidos após testes preliminares com vários valores.

Todo algoritmo evolutivo precisa gerar uma população inicial para que os demais operadores possam entrar em ação. Tarefas que devem ser preferencialmente ativadas devem possuir elevado lucro e/ou baixo custo. As-

sim, a prioridade de ativação de cada tarefa será dada pela razão de seu lucro pelo seu custo acrescido de um pequeno valor aleatório. Após serem gerados todos os indivíduos, a população é ordenada pela aptidão de seus membros de forma decrescente.

A recombinação procura escolher e combinar dois indivíduos já existentes de forma que sejam gerados filhos com características semelhantes a um pai ou a ambos. Para que as soluções geradas sejam interessantes ao problema, é bom que os pais não sejam idênticos ou apresentem grau de semelhança muito elevado. Por outro lado, pais com qualidade superior tendem a gerar filhos melhores. Uma proposta para atender a estes requisitos é dividir a população em classes, por exemplo: classe A – composta pelos 20% melhores indivíduos, classe C – composta pelos 20% piores indivíduos, e classe B – composta pelos demais indivíduos. Um pai é sempre escolhido da classe A e outro sempre da classe B, para tentar evitar que pais muito semelhantes sejam escolhidos, mas que tenham boa qualidade. Na prática, só esta divisão não garante que os pais escolhidos formem o melhor par possível, mas testes preliminares mostraram que, dentre vários esquemas de escolha para os pais, esta divisão da população foi a que apresentou a melhor capacidade de gerar filhos que contribuam para o desenvolvimento das gerações futuras. Portanto, o critério adotado para seleção dos pais deve-se a resultados empíricos mais do que a características teóricas. Outros algoritmos heurísticos que vierem a ser desenvolvidos podem ter desempenho melhor com outros esquemas de seleção de pais.

A combinação dos pais procura gerar uma configuração de prioridades analisando os valores presentes em cada um dos pais. Um critério simples e bastante razoável é aplicar a média aritmética das prioridades dos pais. No operador de recombinação proposto, dois filhos são gerados a cada recombinação. É necessário que se estabeleça alguma forma de diferenciá-los para que não se obtenha indivíduos idênticos. A adição de um pequeno valor aleatório pode ajudar a resolver este problema. Um total de  $N$  filhos é produzido a cada geração.

A seleção natural tem por objetivo definir quais indivíduos poderão continuar existindo na geração seguinte e quais deverão ser eliminados. O critério elitista, no qual os indivíduos que possuem os melhores valores de aptidão são privilegiados, apresentou resultados satisfatórios nos testes preliminares, sendo a escolha dos indivíduos feita de forma determinística. Assim, apenas os  $N$  melhores indivíduos, dentre pais e filhos, permanecerão na geração seguinte.

O operador de mutação consiste em modificar um indivíduo já existente, para que ele ou melhore de qualidade ou transmita seus genes mutantes a outros indivíduos por meio da recombinação. Uma forma de permitir que mutações mais intensas ocorram conforme as gerações se sucedam é atrelar a probabilidade de mutação ao número da geração em questão. Assim, nas primeiras gerações a mutação será pequena e nas últimas gerações

será bem maior. Cada indivíduo terá uma chance de 5% de sofrer mutação a cada geração.

Embora não seja um operador propriamente dito, o critério de parada é um componente importante dos Algoritmos Evolutivos. Um bom critério de parada permite que o algoritmo tenha condições de aprimorar as soluções encontradas sem que seja consumido tempo demais. Possíveis critérios de parada incluem o número de gerações sem melhoria da população, o nível de similaridade entre os indivíduos, tempo computacional máximo ou mesmo algum outro dado derivado do comportamento do algoritmo. Nesta primeira versão de algoritmo evolutivo, no entanto, é utilizado o critério mais trivial que é o número máximo de gerações. Posteriormente, será apresentado outro critério derivado do desempenho obtido pelo algoritmo ao longo de sua execução.

## 4.2 Algoritmo evolutivo com reconstruções - EA\_ages

O Algoritmo Evolutivo EA\_priority apresentou grandes melhorias em relação aos evolutivos anteriores, em especial àqueles propostos em [Silva & Ochi \(2006\)](#). No entanto, o limite de gerações (número de iterações) foi bastante restritivo em algumas instâncias de médio e grande porte. Ao se expandir este limite, recaía-se em outro problema que era a estagnação da população de indivíduos. Uma população é considerada estagnada quando, por exemplo, não consegue gerar indivíduos melhores após algumas gerações consecutivas. A estagnação é um problema comum aos Algoritmos Evolutivos, embora bastante indesejável. Uma saída é incorporar novos mecanismos que permitam aos operadores já existentes uma prorrogação de sua capacidade de gerar indivíduos que não cause a estagnação da população. A seguir são explicados estes mecanismos, partindo-se do princípio que a nova versão do algoritmo evolutivo tem por base o algoritmo EA\_priority.

Como a relação entre o custo e o lucro de uma tarefa pode ter o mesmo valor para diversas combinações (por exemplo: 2/5, 4/10, 6/15), ao elevar o lucro  $p_i$  ao quadrado, espera-se privilegiar as tarefas que, tendo a mesma relação custo-benefício, produzem mais recursos. Também utiliza-se o número de sucessoras para privilegiar tarefas que sejam pré-requisito para outras. Este número de sucessoras é multiplicado pela razão descrita.

Outra forma de tentar melhorar a qualidade de uma solução é aplicar sobre ela algum método de busca local. Algoritmo Evolutivo EA\_ages trabalha com uma relação de vizinhança baseada na troca de duas prioridades. Em outras palavras, uma solução  $S'$  é considerada vizinha de  $S$  se e somente se  $n - 2$  tarefas tiverem as mesmas prioridades e as outras duas tarefas tiverem prioridades trocadas, onde  $n$  é o número total de tarefas de  $S$ . Esta busca local é chamada de LS\_swap.

Visando um gasto menor de tempo, alguns critérios foram adotados para que nem todos os pares de tarefas fossem testados para troca. São

testadas apenas as tarefas que possuem o mesmo nível topológico, mas que tiverem sido ativadas em tempos distintos. Estas tarefas são particularmente interessantes pois poderiam ter sido ativadas no mesmo instante mas não o foram. A troca das prioridades destas tarefas permite que novas escolhas, que não foram contempladas pelo escalonamento original, sejam feitas durante o re-escalonamento que deverá acontecer. Ela será aplicada em duas situações: (i) quando uma nova melhor solução for encontrada, para que ela seja explorada ao máximo; (ii) quando a população não conseguir produzir melhores resultados por 30 gerações seguidas. Neste último caso, a população é considerada estagnada e a busca local será aplicada aos melhores indivíduos, tentando fazer com que estes melhorem pelo menos um pouco suas aptidões. Testes preliminares mostraram que a aplicação apenas na classe A (20% melhores indivíduos) apresenta, em geral, uma boa relação de tempo consumido e melhorias obtidas.

O que se pôde perceber nos testes preliminares é que o número fixo de gerações funciona bem nas instâncias onde rapidamente se produz uma solução próxima ao ótimo global. Nas outras instâncias, este limite fixo pode acabar por encerrar o algoritmo sem que ele consiga esgotar toda sua capacidade de melhoria das soluções correntes. No EA\_ages, o limite de gerações é o mesmo usado na versão anterior, mas conforme o algoritmo consegue gerar soluções melhoradas, o limite é estendido por meio de  $\beta$  gerações extras que são adicionadas a ele. Inicialmente,  $\beta = 4$ ; a cada 20 gerações,  $\beta$  aumenta uma unidade.

Se a população ainda assim permanecer estagnada, não há indícios de que manter a execução normal do algoritmo produzirá resultados melhores. A proposta feita em [Silva & Ochi \(2009\)](#) foi uma mudança mais radical na população estagnada: eliminar todos os indivíduos e reconstruir a população a partir do melhor indivíduo obtido até o momento. O objetivo é produzir uma população mais semelhante a este indivíduo como forma de intensificar as buscas na vizinhança do que se conhece de melhor. A criação de um novo indivíduo similar se dá com a adição de um pequeno valor aleatório a cada prioridade daquele indivíduo tido como semente. Assim, tem-se uma nova população de indivíduos distintos, porém com qualidade próxima ao melhor conjunto de genes já produzido. Esta etapa é chamada de Reconstrução da População.

Esta nova população é tida uma população inicial e o algoritmo procede como se tivesse reiniciado sua execução. O número de gerações restantes, porém, será preservado. O conjunto de gerações entre uma reconstrução e outra será chamada de Era (*Age*, em inglês). As Eras terminam quando forem computadas 150 gerações consecutivas sem mudanças na melhor solução encontrada. A Figura 2 traz um esquema de como os operadores deste algoritmo são executados. O tamanho da população ( $N$ ) no EA3 foi fixado em 20 e em ambos, EA\_priority e EA\_ages, em 100. A probabilidade de mutação de cada indivíduo é de 20% no primeiro e de 5% nos dois

últimos. Todos os parâmetros foram estabelecidos após várias baterias de testes preliminares com cada algoritmo. A seleção dos pais, a recombinação, a mutação e a seleção natural são semelhantes àquelas apresentadas na versão EA\_priority.

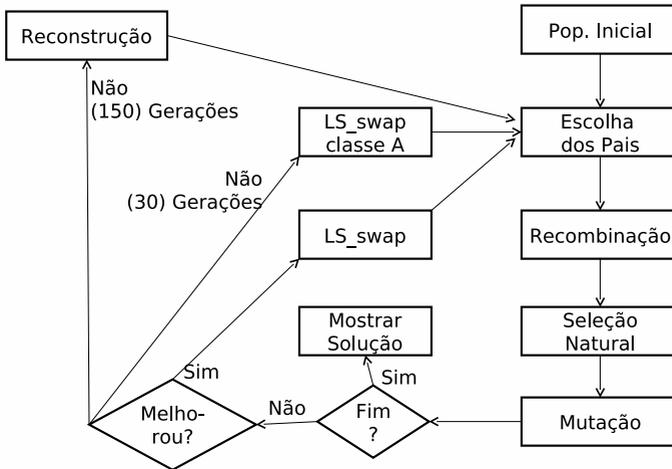


Figura 2. Esquema estrutural do algoritmo EA\_ages.

### 4.3 Versão paralela do EA\_ages

A proposta de uma versão paralela consiste na utilização de processadores com múltiplos núcleos e memória compartilhada, pois praticamente todos os processadores vendidos atualmente têm capacidade de executar simultaneamente dois ou mais processos sem que haja a necessidade de troca de mensagens via rede. O objetivo do emprego desta adaptação foi o de reduzir o tempo computacional gasto para se gerar uma solução de mesma qualidade. O ambiente de testes, descrito na Seção 6, tem capacidade para executar até 4 processos simultaneamente, o que permite que a versão paralela do algoritmo EA\_ages tenha sua carga de trabalho dividida até este grau, sem perda de desempenho. Vale lembrar que a versão sequencial do EA\_ages descrita na seção anterior utiliza apenas um núcleo de processamento, o equivalente a apenas 25% da capacidade total da máquina – outro motivo para o uso de versões paralelas.

A paralelização foi implementada por meio de alguns dos operadores evolutivos utilizados. A seleção dos pais, a recombinação, a mutação, a

aplicação da busca *LS\_swap* quando a população estagna e a reconstrução da população foram paralelizadas e tiveram suas cargas de trabalho divididas por 2 ou 4 núcleos, nas versões com 2 ou 4 *threads* respectivamente. A ordenação da população, a aplicação da *LS\_swap* quando um melhor indivíduo é encontrado e a seleção natural permanecem sequenciais. No caso das etapas paralelizadas, metade ou um quarto da população é entregue a cada núcleo para que possam ser aplicados os devidos operadores. No caso da recombinação, metade ou um quarto dos filhos é gerado por cada núcleo.

## 5. Algoritmos Híbridos Para o PEPRRD

O primeiro dos algoritmos híbridos apresenta uma combinação entre a heurística *EA\_ages* e a formulação matemática F2, proposta na Seção 3 e executada através do *software* CPLEX<sup>1</sup>. O objetivo deste algoritmo é utilizar o ponto forte de uma das abordagens para suprir a deficiência da outra. No caso da execução via CPLEX, um dos grandes problemas é gerar a primeira solução factível. Entretanto, isto deve ser uma tarefa fácil para os métodos considerados heurísticos. No esquema proposto, após a geração da população inicial, é feita uma chamada para que o CPLEX inicie sua execução. Além disto, é informada a melhor solução da população inicial para que o CPLEX tenha, desde o seu início, um limite primal já estabelecido.

Por limite primal entende-se um valor geralmente obtido por uma solução factível, que é próximo (ou igual) ao valor ótimo. O limite dual é um outro valor que indica até onde o valor da solução ótima pode alcançar. Este valor pode ser obtido através de alguma situação de melhor caso que normalmente não pode acontecer na prática, servindo apenas como parâmetro para saber se uma solução está longe de um provável valor ótimo. Quando o limite primal e dual têm o mesmo valor, este será o valor ótimo do problema. Em problemas de maximização, o valor ótimo é sempre maior ou igual ao limite primal e menor ou igual ao limite dual. O otimizador utilizado fornece e atualiza estes limites conforme vai executando a formulação matemática empregada.

O esquema híbrido proposto prevê ainda que, ao encontrar uma nova solução incumbente, isto é, uma nova melhor solução, o CPLEX deve repassá-la ao algoritmo evolutivo, que já roda paralelamente. A solução é inserida na população corrente na forma de um indivíduo extra. Espera-se que este indivíduo possa transferir parte de sua carga genética a seus sucessores e que estes tenham uma aptidão melhorada. A codificação do indivíduo é algo bastante abstrato pois, para o CPLEX, não existe a questão da prioridade entre as tarefas. Alguns mecanismos foram testados e um dos que apresentaram resultado bem satisfatório foi o de multiplicar por uma

---

<sup>1</sup> <http://www.ilog.com/products/cplex>

constante positiva  $K$  a diferença  $H - S_i$ , onde  $S_i$  é o tempo de ativação da tarefa  $i$  na solução encontrada pelo CPLEX. Se uma tarefa não estiver ativada nesta solução, seu valor  $S_i$  será considerado igual a  $H + 1$ , o que resulta em uma prioridade negativa, ou seja, muito baixa.

É importante notar que a passagem de um limite primal melhor para o CPLEX tem dois efeitos positivos: (i) subproblemas que tiverem um valor da relaxação linear pior do que o da solução passada serão automaticamente eliminados; (ii) permite-se que o CPLEX utilize suas técnicas de polimento e inferência de cortes sobre uma solução mais próxima do valor ótimo. Portanto, é provável que sejam obtidos resultados melhores por meio destas técnicas.

O critério de parada deste algoritmo híbrido será o término da otimização realizada pelo CPLEX. Logo, este algoritmo pode ser considerado como um método exato, já que, se for dado a ele tempo suficiente, sempre fornecerá uma ou mais soluções ótimas. Infelizmente na prática, não é viável permitir que ele execute até que prove a otimalidade da solução nas instâncias de maior porte – neste(s) caso(s) será dado um limite para a execução do método híbrido.

### 5.1 CPLEX como busca local

Uma forma muito interessante de utilizar o otimizador CPLEX é realizar uma prefixação do valor de variáveis que estejam relacionadas a uma solução gerada heurísticamente. O algoritmo *Local Branching* de Fischetti & Lodi (2003) é um exemplo desta técnica. Para o PEPRRD, foi desenvolvido um algoritmo que utiliza novamente a heurística EA\_ages, porém em sua versão paralela. Toda vez que esta heurística encontrar uma nova melhor solução  $S$ , será criada uma formulação que levará em conta o tempo de ativação de cada tarefa em  $S$  para que o CPLEX possa examinar a vizinhança desta solução. Aqui, uma solução  $S'$  será considerada vizinha de  $S$  se  $S_i - Z \leq S'_i \leq S_i + Z \quad \forall i = 1, \dots, n$ , onde  $Z$  é outra constante inteira positiva e  $S_i$  ( $S'_i$ ) é o tempo de ativação da tarefa  $i$  na solução  $S$  ( $S'$ ).

Quando o CPLEX realizar a busca local, provavelmente encontrará soluções piores e melhores que  $S$ . Para acelerar a execução da busca local, é informado ao CPLEX o valor da solução  $S$  que serve de limite primal para o algoritmo. Assim, nenhuma solução de qualidade inferior é considerada. Infelizmente, a busca local pode demorar muito tempo, principalmente em instâncias com muitas tarefas. Como o objetivo da busca local não é necessariamente encontrar a solução ótima global do problema, mas apenas buscar na vizinhança de uma solução, é estabelecido um limite de tempo para que a busca seja executada. Ao final, a melhor solução encontrada é devolvida ao EA\_ages que a utiliza como um indivíduo de sua população.

Se outro melhor indivíduo for gerado, o CPLEX é novamente acionado para tentar aprimorar esta nova solução. O operador que realiza a busca local LS\_swap será substituído pela aplicação desta busca local, chamada

LS\_cplex, em um indivíduo escolhido aleatoriamente dentro desta população. Isto é feito para que a LS\_cplex não seja aplicada sempre sobre o mesmo indivíduo, nem que seja aplicada tantas vezes que torne o algoritmo muito custoso computacionalmente.

## 6. Resultados Computacionais

Esta seção traz resultados dos principais testes realizados com as heurísticas e com os métodos exatos discutidos nas sessões anteriores. Os testes foram realizados em computadores Intel Quad-core 9550, com 8 Gigabytes de memória RAM. A versão do CPLEX utilizada foi a 11.2 paralela (com até 4 *threads*). O código dos métodos heurísticos e híbridos foi escrito em linguagem C.

Todas as instâncias utilizadas fazem parte do Projeto Labic<sup>2</sup>, do IC/UFF. São as únicas instâncias existentes para o PEPRRD, já que as instâncias para outros problemas de escalonamento de projetos não contêm uma característica fundamental que é o lucro produzido pelas tarefas.

A topologia dos grafos foi elaborada para que 10% das tarefas não tenham predecessor algum e as demais tenham de 1 a 5 predecessores aleatoriamente escolhidos. O custo e o lucro das tarefas foram aleatoriamente escolhidos dentro dos intervalos [1; 50] e [1; 10], respectivamente. Custos maiores e/ou lucros menores resultam numa grande diminuição do número de tarefas ativadas, tornando as instâncias mais fáceis. Por outro lado, custos menores e/ou lucros maiores permitem que todas as tarefas sempre valham a pena ser ativadas, diminuindo a dificuldade da mesma forma. A quantidade inicial de recursos  $Q_0$  foi aleatoriamente escolhida no intervalo [ $MinCusto$ ; 50], onde  $MinCusto$  é o menor custo entre as tarefas sem precedência. O horizonte de planejamento foi estabelecido com tamanho igual à raiz quadrada do número de tarefas.

Os principais fatores que determinam a dificuldade de uma instância usualmente são o número de tarefas e o tamanho do horizonte de planejamento. Uma instância com 5000 tarefas, por exemplo, tende a ser mais difícil do que outra com 2000 tarefas. Uma instância com 20 unidades de tempo também tende a ser mais difícil do que outra com apenas 8 unidades. A distribuição de custos e lucros, como já mencionado, como também a topologia do grafo podem influenciar na dificuldade da instância, porém de forma menos direta.

A Tabela 1 apresenta comparações dos limites primais e duais gerados pela formulação F2. Todos os parâmetros do CPLEX foram deixados com seus valores padrões e foi estabelecido um tempo limite de 50000 segundos para cada execução. Para as instâncias de menor porte (menos de 300 tarefas), o ótimo foi encontrado e provado dentro do limite de tempo. A

---

<sup>2</sup> <http://www.ic.uff.br/lablc>

quantidade de tarefas de cada instância é mostrada no número que antecede à letra “a”, no nome da instância.

Tabela 1. Limites primais e duais obtidos pela formulação F2.

Instância	Limite dual	Limite primal
350a	2571	2571,0
400a	7271	7271,0
450a	9705	9705,0
500a	14336	14336,0
550a	11087	11136,5
600a	10157	10199,6
650a	16332	16355,8
700a	31820	31836,2
750a	36216	36296,9
800a	38351	38729,8
850a	46840	46898,0
900a	38733	38834,8
950a	67903	67927,4
1000a	71864	71911,9

Um marco importante no desenvolvimento de métodos heurísticos foi a mudança na maneira de representar a solução. Os algoritmos evolutivos propostos em [Silva et al. \(2008\)](#) e [Silva & Ochi \(2009\)](#), respectivamente EA\_priority e EA\_ages, apresentaram grandes melhorias em relação aos algoritmos evolutivos previamente propostos. A Tabela 2 indica a média (de 30 execuções) dos resultados obtidos por aqueles evolutivos que utilizam a representação indireta (por prioridades) e a representação direta (EA3, proposto em [Silva & Ochi \(2007\)](#)). O critério de parada utilizado foi o número máximo de gerações.

Tabela 2. Média dos resultados dos algoritmos evolutivos.

Instância	EA3	EA_priority	EA_ages
100a	<b>304,0</b>	303,2	<b>304,0</b>
200a	613,1	<b>636,0</b>	632,1
300a	1734,3	1958,5	<b>2004,2</b>
400a	5500,6	5962,2	<b>6908,5</b>
500a	11386,3	11954,3	<b>13523,7</b>
600a	7097,8	8616,0	<b>9358,0</b>
700a	23775,5	26217,7	<b>28500,6</b>
800a	31642,3	32354,4	<b>35001,9</b>
900a	28379,1	29912,6	<b>34355,7</b>
1000a	60751,9	63512,8	<b>66399,7</b>

É possível perceber pela Tabela 2 que, em geral, os algoritmos evolutivos que utilizam a representação indireta são bem superiores em relação ao EA3. Na instância 600a, por exemplo, tem-se uma melhoria de mais de 30%. Isto se deve muito à forma de representação e ao algoritmo de recombinação utilizados em EA\_priority e EA\_ages. A Figura 3 ilustra situações como esta na instância 550a. Cada algoritmo rodou por 100 gerações e foram anotadas a qualidade da melhor solução encontrada e a aptidão média dos indivíduos de cada geração computada pelos evolutivos EA3, EA\_priority e EA\_ages.

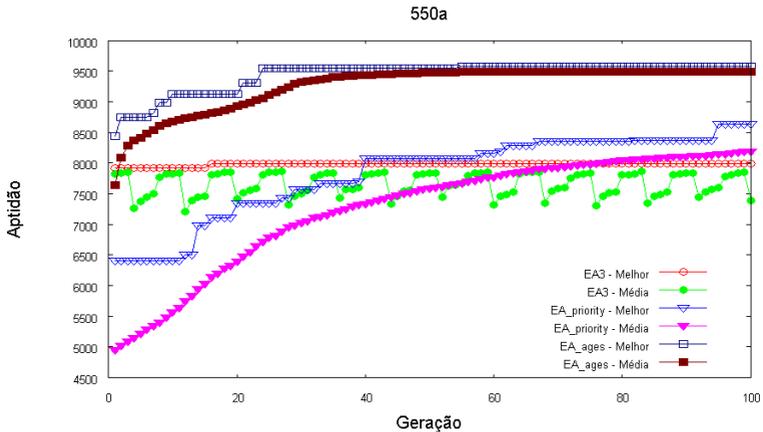


Figura 3. Evolução da melhor solução e da média da população.

O algoritmo EA3 conseguiu evoluir o melhor indivíduo poucas vezes. Sua recombinação, embora conseguisse produzir populações com boa qualidade em relação ao melhor indivíduo, teve dificuldades em estabelecer novos padrões genéticos de qualidade superior. É visível, pelos gráficos, que os mecanismos de intensificação e diversificação do EA3 tentaram cumprir seu papel de promover uma reinicialização da população quando esta foi considerada estagnada, mas sem eficiência suficiente. A recombinação dos algoritmos evolutivos EA\_priority e EA\_ages teve desempenho bem melhor, já que um número bem maior de evoluções foi conseguido. O EA\_ages mostrou um desempenho um pouco melhor, inclusive em relação à média da população que foi sempre bem mais próxima da melhor solução do que a apresentada pelo EA\_priority.

Pelos experimentos computacionais efetuados, ficou claro que a chance de gerar indivíduos melhores do que os pais é bem maior usando a representação indireta e o operador de recombinação correspondente. Mesmo assim, após mais de 200 ou 300 gerações, a população começa a estagnar,

ou seja, a não produzir indivíduos melhores. O algoritmo EA\_ages foi desenvolvido para tentar superar estas dificuldades através de uma série de novos operadores. A aplicação da LS\_swap, que troca pares de prioridades, ajudou a aumentar a qualidade dos indivíduos que eram considerados novos *best*. Mas, sem dúvida, o operador que permitiu adiar o estado de estagnação da população foi a reconstrução da população baseada no melhor indivíduo. Esta reconstrução, como já foi mencionado, é ativada quando a população é considerada estagnada, ou seja, quando não há indícios de que ela possa aprimorar mais a qualidade de seu melhor indivíduo, o que ocorre após 150 gerações sem melhorias.

O primeiro experimento com os métodos híbridos levou em conta o algoritmo EA\_ages + CPLEX e a formulação F2, rodando em algumas instâncias, durante 25000 segundos (metade do tempo dado às formulações nos testes do início desta seção). O método híbrido foi comparado a duas versões do CPLEX. A primeira versão com todos os parâmetros com valor padrão é chamada de DS0, pois utiliza o algoritmo de otimização *dynamic search*, sem ênfases específicas. A segunda versão, chamada de BB4, utiliza o algoritmo de *branch-and-bound* tradicional com ênfase 4, significando que o otimizador gastará mais tempo do que o normal para buscar soluções factíveis a cada etapa da otimização. Esta segunda versão foi testada porque, ao se implementar o método híbrido, o CPLEX automaticamente ajusta os parâmetros para os valores citados, sem que seja possível alterá-los. A comparação do híbrido com o BB4 é, portanto mais justa do que com o DS0, já que os algoritmos e parâmetros internos são idênticos. A versão DS0 foi testada pois é a maneira mais tradicional de se executar a formulação matemática, já que a maioria dos estudos utiliza o otimizador isoladamente sem alterar seus parâmetros. Infelizmente, porém, não existem estudos ou maiores explicações sobre como o *dynamic search* funciona, como ele implementa as escolhas pertinentes ao processo de otimização e quais as diferenças entre ele e o método *branch-and-bound* tradicional.

Três atributos de execução foram analisados: o tempo total consumido, o valor da melhor solução encontrada e o limite dual ao final da execução. A Tabela 3 mostra os limites, enquanto a Tabela 4 mostra os tempos. O símbolo “-” indica que o tempo total consumido pelo algoritmo foi igual ao tempo limite de 25000 segundos ou que o limite dual é igual à melhor solução encontrada. Obviamente, este último fato só ocorre se a tal solução for uma solução ótima. Além das instâncias já testadas, um novo conjunto de instâncias foi incluído no experimento, visando embasar melhor as análises. Em negrito é destacado o melhor limite solução encontrada ou o menor tempo total.

Comparando os resultados das versões DS0 e BB4 pode-se perceber, na maioria das instâncias, que a DS0 produziu soluções melhores. Nas instâncias 350a3, 350a4, 350a7 e 400a2, a BB4 não conseguiu nem terminar

Tabela 3. Limites primais e duais da versão EA\_ages + CPLEX versus CPLEX.

Inst.	Solução			Limite Dual		
	DS0	BB4	Híbrido	DS0	BB4	Híbrido
300a	2073	2073	2073	-	-	-
350a	2571	2571	2571	-	-	-
400a	7271	7271	7271	-	-	-
450a	9705	9705	9705	-	-	-
500a	14336	14336	14336	-	-	-
550a	<b>11081</b>	11054	11054,7	<b>11140</b>	11152	11162
600a	10157	10155	<b>10159,2</b>	10205	10239	<b>10197,9</b>
650a	<b>16332</b>	16308	16331,3	16363	16446	<b>16351</b>
700a	31820	31818	<b>31826,0</b>	31843	31854	-
750a	36216	36220	<b>36225,7</b>	36302	36317	<b>36291</b>
800a	0	0	<b>38641,1</b>	38736	38757	<b>38686</b>
850a	46835	0	<b>46837,2</b>	<b>46912</b>	46968	46920
900a	38723	38650	<b>38787,9</b>	38837	38845	<b>38823</b>
950a	67902	67854	<b>67909,0</b>	67931	67943	<b>67916</b>
1000a	<b>71851</b>	71844	71838,6	<b>71914</b>	71920	71930
300a2	2399	2399	2399,0	-	-	-
300a3	2340	2340	2340,0	-	-	-
300a4	2434	2434	2434,0	-	-	-
300a5	4193	4193	4193,0	-	-	-
300a6	1257	1257	1257,0	-	-	-
300a7	1707	1707	1707,0	-	-	-
300a8	2009	2009	2009,0	-	-	-
300a9	2915	2915	2915,0	-	-	-
300a10	3233	3233	3233,0	-	-	-
350a2	3945	3945	3945,0	-	-	-
350a3	3521	3521	3521,0	-	3525	-
350a4	4126	<b>4128,0</b>	-	4147	-	-
350a5	3614	3614	3614,0	3617	3629	-
350a6	8328	8328	8328,0	-	-	-
350a7	3586	3586	3586,0	-	3594	-
400a2	11254	11254	11254,0	-	11267	-
400a3	5296	5296	5296,0	-	-	-
400a4	10229	10229	10229,0	-	10230	-

a otimização antes do tempo limite. A aparente superioridade da DS0 é de difícil explicação, já que pouco se sabe sobre as técnicas utilizadas por ela.

Tabela 4. Tempo computacional (seg.) da versão EA\_ages + CPLEX versus CPLEX.

Inst.	DS0	BB4	Híbrido
300a	196,3	<b>178,9</b>	2460,4
350a	211,3	408,6	<b>155,0</b>
400a	467,8	1609,2	<b>163,7</b>
450a	9843,2	7570,0	<b>1143,7</b>
500a	6688,7	7228,1	<b>822,6</b>
550a	-	-	-
600a	-	-	-
650a	-	-	-
700a	-	-	<b>15061,2</b>
750a	-	-	-
800a	-	-	-
850a	-	-	-
900a	-	-	-
950a	-	-	-
1000a	-	-	-
300a2	<b>386,9</b>	435,4	1210,0
300a3	358,6	18679,0	<b>90,6</b>
300a4	1732,4	2410,9	<b>771,5</b>
300a5	<b>486,6</b>	1358,3	486,9
300a6	96,1	76,3	<b>38,3</b>
300a7	191,0	<b>95,9</b>	112,7
300a8	480,3	629,8	<b>105,9</b>
300a9	<b>310,4</b>	986,7	446,5
300a10	111,5	93,8	<b>46,6</b>
350a2	3319,7	9459,7	<b>1925,3</b>
350a3	7403,3	-	<b>1198,9</b>
350a4	22771,9	-	<b>18731,5</b>
350a5	-	-	<b>387,7</b>
350a6	<b>770,9</b>	870,1	820,9
350a7	4662,9	-	<b>1642,8</b>
400a2	2895,8	-	<b>464,7</b>
400a3	9745,5	14087,9	<b>3681,8</b>
400a4	<b>1443,2</b>	5944,6	2724,0

Realizando a comparação do DS0 com o híbrido EA\_ages + CPLEX (resultados médios de 10 execuções), pode-se notar que a versão híbrida consegue produzir soluções ainda melhores, na maioria das instâncias. Das

10 instâncias onde não foi possível encontrar o valor ótimo (550a–1000a), em 7 delas a versão híbrida produziu resultados médios melhores e em outras 7 o limite dual foi melhor. Das 22 instâncias onde o tempo limite não foi extrapolado, em 17 delas a versão híbrida foi mais rápida, terminando mais cedo.

A instância 700a é um caso à parte. Nela, o método híbrido conseguiu encontrar o valor ótimo em aproximadamente 15000 segundos, o que é um ótimo resultado se for considerado que, nos primeiros testes, utilizando as duas formulações, isto não foi possível mesmo em 50000 segundos. A explicação para tantos resultados bons se deve às primeiras soluções que são geradas pelo algoritmo evolutivo e aprimoradas por ele e pelo CPLEX. Estas soluções permitem que o limite primal seja rapidamente elevado (em relação a uma execução normal do evolutivo ou do CPLEX), reduzindo o número de nós a serem analisados e possibilitando ao CPLEX executar heurísticas de aprimoramento sobre uma solução de qualidade melhor.

Para realizar os testes com o EA\_ages + LS\_cplex, dois parâmetros foram calibrados: o tempo máximo de execução das buscas locais feitas pelo CPLEX e o valor de  $Z$ . Testes preliminares mostraram que um limite de 100 segundos e um valor  $Z = 4$  foram a melhor configuração. A Tabela 5 mostra os resultados do experimento. O critério de parada foi o tempo limite de 2 horas para as instâncias onde a formulação F2 não encontrou o valor ótimo. Nas demais instâncias, o limite foi o tempo que a formulação F2 levou para terminar a otimização. Os tempos estão presentes na segunda coluna.

Um dos resultados que chamou a atenção foi a robustez do algoritmo, indicada pela pequena diferença entre a melhor e a pior execução (coluna “gap%”). São valores muito pequenos se comparados aos dos outros métodos heurísticos onde a maioria dos valores ficou acima de 2,0%. De fato, a reiterada execução da busca LS\_cplex faz com que o algoritmo produza resultados muito próximos, já que boa parte do espaço de soluções está sendo analisado. Os resultados médios (quinta coluna) também são muito bons, mostrando que a inclusão da LS\_cplex conseguiu melhorar os resultados do evolutivo EA\_ages. A última coluna da Tabela 5, mostra o quanto a solução média do EA3 foi superada em relação ao resultado médio do EA\_ages + CPLEX. Em algumas instâncias as melhorias passaram de 50%, ficando em torno de 30%, na média. Este resultado é bastante positivo, principalmente se for levado em consideração que o EA3 já possui mecanismo para evitar a convergência prematura, que é um dos grandes problemas dos Algoritmos Evolutivos.

Um outro experimento foi realizado para verificar o desempenho dos dois métodos híbridos em um curto espaço de tempo. Foi estabelecido um limite de apenas 300 segundos para que os algoritmos possam atingir o alvo. Neste caso, o resultado médio do algoritmo EA\_ages foi definido como alvo. Várias instâncias de diferentes tamanhos foram aleatoriamente escolhidas,

Tabela 5. Resultados do algoritmo que usa a LS\_cplex.

Inst.	Tempo (s)	EA3	+CPLEX	+LS_cplex	gap (%)	Méd (%)
200a	6,4	604,1	636,0	636,0	0,00	5,3
300a	155,4	1757,8	2072,5	2038,2	0,15	17,9
400a	440,7	5550,2	7271,0	7217,2	0,54	31,0
500a	896,8	11345,2	15335,9	14319,7	0,31	26,4
600a	7200,0	7049,3	10157,6	10107,0	0,29	44,1
700a	7200,0	20982,2	31823,9	31819,6	0,07	51,7
800a	7200,0	30491,9	38637,0	38626,1	0,04	26,7
900a	7200,0	25233,3	38768,6	38766,6	0,11	53,6
1000a	7200,0	60080,5	71843,0	71855,8	0,06	19,6

50 execuções foram feitas para cada uma e a Figura 4 mostra a análise probabilística do experimento para a instância 700a. Nesta análise, cada algoritmo foi executado 100 vezes e os tempos consumidos para que uma solução atingisse o valor alvo foram anotados e ordenados crescentemente. Um ponto  $(t, p)$  no gráfico mostra que a  $p$ -ésima execução mais rápida demorou  $t$  segundos para atingir o alvo.

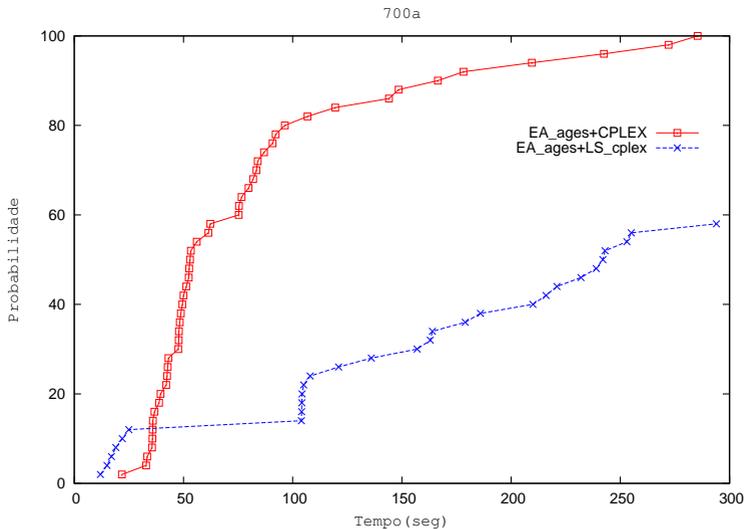


Figura 4. Análise probabilística das versões EA\_ages + CPLEX e EA\_ages + LS\_cplex.

O desempenho do EA\_ages + LS\_cplex não foi muito significativo exatamente porque ele fica dependente da vizinhança da solução a ele passada. Como o experimento foi projetado para testar a capacidade dos algoritmos em prover boas soluções rapidamente, fica claro que, para isto ocorrer, é necessário uma quantidade maior de tempo. As buscas locais até foram capazes de encontrar vizinhos melhores, mas estes eram apenas ligeiramente superiores às soluções originais, o que acabou não sendo suficiente para que o EA\_ages + LS\_cplex tivesse desempenho semelhante ao EA\_ages + CPLEX.

Portanto, analisando-se os resultados obtidos, pode-se dizer que o algoritmo EA\_ages + CPLEX apresenta os melhores resultados dentre os métodos não-determinísticos, tanto em testes com limite de tempo muito pequeno quanto em testes onde o algoritmo pode ser mais longamente executado.

## 7. Conclusões

Este trabalho abordou o Problema de Escalonamento de Projetos com Restrições de Recursos Dinâmicos (PEPRRD), através de duas formulações matemáticas, de métodos heurísticos e métodos híbridos. O PEPRRD se diferencia dos demais problemas de escalonamento de projetos da literatura porque as tarefas não só consomem recursos quando são ativadas, mas também são capazes de gerá-los após suas ativações. O problema encontra aplicações práticas em ambientes comerciais ou industriais nos planos de expansão das empresas.

Um novo esquema de representação das soluções do PEPRRD foi estudado. Este esquema é composto por uma lista de prioridades para cada solução. O algoritmo de escalonamento ativa as tarefas que estiverem disponíveis respeitando a prioridade estabelecida. Desta forma, qualquer lista de prioridade pode dar origem a um escalonamento viável.

O primeiro algoritmo evolutivo analisado, EA\_priority, foi capaz de gerar resultados já bem superiores ao algoritmo EA3, um dos melhores algoritmos heurísticos conhecidos para o PEPRRD. A quantidade de atualizações na melhor solução aumentou consideravelmente. A versão mais complexa, EA\_ages, apresentou características muito interessantes como o critério de parada com gerações extras, a busca local LS\_swap mas, principalmente, a estratégia de reconstrução de população quando esta for considerada estagnada. Todas estas características se mostraram eficientes. As versões paralelas conseguiram obter resultados melhores do que a versão sequencial em tempo um pouco menor.

Os métodos híbridos, que mesclam elementos heurísticos e exatos, foram propostos com a intenção de aproveitar o melhor das duas abordagens. A média dos resultados deste método (EA\_ages + LS\_cplex) foi quase tão

boa quanto a do EA\_ages + CPLEX, com a vantagem de apresentar uma robustez muito maior.

## Agradecimentos

Os autores agradecem ao Conselho Nacional de Pesquisa (CNPq) pelo apoio financeiro ao projeto, sob protocolo 141074/2007-8.

## Referências

- Blazewicz, J.; Lenstra, J. & Rinnooy Kan, A., Scheduling projects to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1):11–24, 1983.
- Damak, N.; Jarboui, B.; Siarry, P. & Loukil, T., Differential evolution for solving multi-mode resource-constrained project scheduling problems. *Computers & Operations Research*, 36(9):2653–2659, 2009.
- Fischetti, M. & Lodi, A., Local branching. *Mathematical Programming*, 98(1-3):23–47, 2003.
- Holland, J., *Adaptation in Natural and Artificial Systems*. Ann Arbor, USA: The University of Michigan Press, 1975.
- Mendes, J.; Gonçalves, J. & Resende, M., A random key based genetic algorithm for the resource constrained project scheduling problems. *Computers and Operations Research*, 36:92–109, 2009.
- Mladenović, N.; Brimberg, J.; Hansen, P. & Pérez, J., The p-median problem: a survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.
- Nonobe, K. & Ibaraki, T., Formulation and tabu search algorithm for the resource constrained project scheduling problem. In: Ribeiro, C. & Hansen, P. (Eds.), *Essays and Surveys in Metaheuristics*. 2002.
- Silva, A. & Ochi, L., A dynamic resource constrained task scheduling problem. In: *Proceedings of Latin-Ibero-American Congress on Operations Research (CLAIO)*. Montevideo, Uruguai, 2006.
- Silva, A. & Ochi, L., A hybrid evolutionary algorithm for the dynamic resource constrained task scheduling problem. In: *Proceedings of the International Workshop on Nature Inspired Distributed Computing (NIDISC'07)*. Long Beach, USA, 2007.
- Silva, A. & Ochi, L., New sequential and parallel algorithm for dynamic resource constrained project scheduling problem. In: *Proceedings of the IEEE International Symposium on Parallel & Distributed Processing*. Piscataway, USA: IEEE Computer Society, p. 1–7, 2009.
- Silva, A.; Ochi, L. & Santos, H., New effective algorithm for dynamic resource constrained project scheduling problem. In: *Proceedings of*

*International Conference on Engineering Optimization (ENGOPT)*.  
Rio de Janeiro, RJ, 2008.

Snyder, L.V. & Daskin, M.S., A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research*, 174(1):38–53, 2006.

Valls, V.; Ballestín, F. & Quintanilla, S., A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 185(2):495–508, 2008.

## Notas Biográficas

**André Renato Villela da Silva** é Doutor em Ciência da Computação pela Universidade Federal Fluminense (Niterói-RJ) e atualmente leciona na Universidade Plínio Leite, também em Niterói.

**Luiz Satoru Ochi** é Doutor em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro (COPPE-SISTEMAS/UFRJ). Atualmente é Professor Titular no Instituto de Computação da UFF (IC-UFF) e pesquisador do CNPq, área de Ciência da Computação, nível 1D.

## Algoritmos Evolucionários na Solução do Problema do Caixeiro Alugador

Marco C. Goldberg\*, Paulo H. Asconavieta da Silva  
e Elizabeth F. G. Goldberg

---

**Resumo:** O Caixeiro Alugador é uma nova variante do Caixeiro Viajante com aplicações na otimização da programação do aluguel de carros e em sistemas de transportes em geral. O presente trabalho define o problema, estabelece sua complexidade computacional e apresenta algoritmos metaheurísticos para sua solução. Um experimento computacional é relatado de forma a avaliar o desempenho de dois tipos de algoritmos evolucionários na solução do problema proposto. Visando sustentar eventuais conclusões frente ao estado da arte, são relatados os resultados alcançados pelos algoritmos e três outros algoritmos metaheurísticos aplicados ao problema. Ao final são apresentadas as conclusões.

**Palavras-chave:** Otimização combinatória, Caixeiro alugador, Metaheurísticas, Otimização de sistemas de transporte.

**Abstract:** *The Car Renter Salesman is a new variant of the Traveling Salesman Problem with applications in the optimization of car renting schedule and general transportation systems. This paper defines the problem, establishes its computational complexity and presents metaheuristic algorithms for its solution. A computational experiment is reported in order to evaluate the performance of two types of evolutionary algorithms for solving the proposed problem. Aiming at supporting eventual conclusions given the state of the art, the results achieved by the algorithms and three other metaheuristic algorithms applied to the problem are reported. At the end conclusions are present.*

**Keywords:** *Combinatorial optimization, Car renter salesman, Metaheuristics, Transportation systems optimization.*

---

\* Autor para contato: marcocgold@gmail.com

## 1. Introdução

O problema do Caixeiro Viajante (PCV) é um dos clássicos problemas da Otimização Combinatória, consistindo em determinar em um grafo ponderado  $G = (N, M)$ , onde  $N = \{1, \dots, n\}$  representa o conjunto de vértices do grafo e  $M = \{1, \dots, m\}$  o conjunto de arestas, um ciclo Hamiltoniano de menor custo. O PCV é *NP*-Difícil (Garey & Johnson, 1979), sendo um dos problemas de otimização combinatória mais intensamente pesquisados na literatura. Ele possui várias e importantes aplicações práticas (Gutin & Punnen, 2002) e diversas variantes. O problema do Caixeiro Alugador ou *Car Renter Salesman Problem* (CaRS), até onde sabem os autores, é uma nova variante do PCV que possui aplicação na Engenharia de Transportes, no segmento de transporte turístico. Este modelo tanto pode representar importantes aplicações na área do aluguel de transporte para turismo (Car, 2008), quanto possui aplicações em manufatura flexível. Por admitir o PCV como um subcaso elementar, o modelo representa uma variante de complexidade seguramente desafiadora. Sob a dimensão da aplicação na Engenharia de Transporte o trabalho realiza também uma revisão do estado da arte dos modelos propostos para a solução dos problemas de transporte turístico.

Como o objetivo principal da pesquisa, todavia, é contribuir para a solução do problema citado. O presente trabalho relata um estudo experimental para a solução do Problema do Caixeiro Alugador envolvendo a elaboração e comparação de desempenho entre algoritmos evolucionários do tipo Memético (Moscato, 1989) e Transgenético (Goldbarg et al., 2009). Tais algoritmos metaheurísticos foram escolhidos em virtude de pertencerem a classes de algoritmos que possuem reconhecido desempenho na solução de problemas envolvendo roteamento Hamiltoniano em grafos, o tipo de problema objeto da pesquisa.

A Seção 2 do presente trabalho define o problema do Caixeiro Alugador e examina as várias condições sob as quais esta variante do caixeiro viajante pode se apresentar. A Seção 3 apresenta os fundamentos da abordagem transgenética. A Seção 4 descreve dois algoritmos evolucionários de solução, um algoritmo Memético e um algoritmo Transgenético. Na Seção 5 o trabalho desenvolve um experimento computacional para comparar o algoritmo transgenético proposto como um algoritmo da literatura. Na Seção 6 são relatadas as conclusões do experimento e do comportamento dos diferentes algoritmos envolvidos.

## 2. O Problema do Caixeiro Alugador

### 2.1 A indústria de aluguel de veículos

Existem hoje mais de 90 companhias de locação de carros com porte econômico significativo no mercado mundial (Car, 2008). A importância do ne-

gócio de aluguel de carros pode ser medida tanto pelo faturamento do setor como através do porte das companhias prestadoras de serviço. Por exemplo, a empresa Hertz possui aproximadamente 8.000 postos de atendimento em cerca de 145 países (Hertz, 2009). Por outro lado, a Enterprise Rent-A-Car, outra importante companhia do setor possui uma frota de aluguel com cerca de 878.000 veículos operando em cerca de 6.900 agências (Enterprise Rent-A-Car, 2009). Por sua vez, a empresa Avis opera em cerca de 3.800 postos de aluguel espalhados pela Europa, África, Oriente Médio e Ásia. Em dezembro de 2007 a Avis contabilizava uma frota de aluguel com cerca de 118.000 veículos (Avis, 2009).

Os números que reportam a movimentação financeira deste setor são impressionantes. O Avis Budget Group faturou 5,1 bilhões de dólares em 2009 (Avis, 2010). A Enterprise Rent-A-Car, uma empresa da Enterprise Holdings, faturou nos Estados Unidos 9,06 bilhões de dólares em 2006. Em 2009 a Enterprise Holdings que possui, hoje, ainda a National Car Rental, Alamo Rent-A-Car and WeCar faturou cerca de 12,1 bilhões de dólares (Conrad & Perlut, 2006). Os números citados representam apenas uma parte do mercado americano que ainda conta com outras grandes redes de aluguel de carros como a Dollar e a Hertz. O mercado mundial em 2012 é estimado em 52,6 bilhões de dólares (Car, 2008).

Todavia, o aluguel dos veículos é apenas uma parte dos custos de transporte associados às atividades de turismo e outras que demandam deslocamentos terrestres para negócios variados. Sobre estes custos se somam pelo menos as despesas com combustível, pedágios e seguros. Presentemente as opções de aluguel se diversificam cada vez mais com a expansão das empresas do setor, justificando-se a busca de esquemas de aluguel que minimizem o custo total desta forma de transporte.

## 2.2 Modelos de otimização combinatória para a indústria de aluguel de carros

Dentre os diversos problemas logísticos deste ramo de atividades, a literatura relata estudos específicos de otimização combinatória no problema de programação de frota (Li & Tao, 2010), planejamento estratégico e tático de frota de veículos (Pachon et al., 2003), previsão de demanda (Edelstein & Melnyk, 1977) e gestão de frota de veículos com restrições de manutenção (Hertz et al., 2009). É também comum o emprego de modelos de otimização em transportes para encaminhar a solução de vários problemas na indústria de aluguel de carros (Yang et al., 2008).

Os trabalhos da literatura focalizam o ponto de vista da locadora de veículos. A otimização dos custos sob a ótica do cliente ainda não foi, do conhecimento dos autores, objeto de pesquisa publicada.

### 2.3 O problema do caixeiro alugador

De uma forma geral, sob o ponto de vista de um usuário de carros alugados, o objetivo é realizar um dado conjunto de deslocamentos minimizando os custos totais da atividade. Adicionalmente, no caso do aluguel turístico, é bastante comum que o alugador inicie e termine seu *tour* terrestre em uma mesma cidade, realizando os deslocamentos mais longos através de outros meios de transporte. No deslocamento, além dos custos relativos ao aluguel, somam-se, pelo menos, os custos relativos ao combustível e pedágio. Considerando um grafo  $G = (N, M, W)$ , onde  $N = \{1, \dots, n\}$  representa o conjunto de vértices do grafo,  $M = \{1, \dots, m\}$  o conjunto de arestas e  $W = \{1, \dots, w\}$  o conjunto das distâncias entre os vértices de  $G$  ou o comprimento das arestas do conjunto  $M$ . O problema presentemente descrito possui as características citadas a seguir.

Estão disponíveis para o aluguel vários tipos de carro, cada qual possuindo características próprias, ou seja, custos de operação específicos. Tais custos englobam o consumo de combustível, eventuais taxas de pedágio e o valor do aluguel. Como as taxas de pedágio dependem, normalmente, tanto do tipo de carro como da extensão do trecho percorrido e o valor do aluguel pode ser associado ao quilômetro rodado, sem perda de generalidade pode-se considerar que todos os custos são contabilizados em função de cada carro em um valor associado ao peso ou comprimento das arestas  $(i, j)$  do grafo  $G$ . O custo total de operação de um dado carro  $k$  ao percorrer uma aresta  $(i, j)$  é representado pela variável  $c_{ij}^k$ .

Um carro alugado em uma dada operadora somente poderá ser devolvido em uma cidade que conta com pelo menos uma agência desta mesma locadora ou agência conveniada. Conseqüentemente, não é permitido ao caixeiro alugar um carro de uma dada operadora para cumprir um trecho de sua rota, se este carro não puder ser devolvido, na última cidade do trecho, em local onde não exista uma agência com capacidade para executar seu recebimento. Sempre que for possível alugar um carro em uma cidade  $i$  e devolvê-lo em uma cidade  $j$  sendo  $i \neq j$ , haverá uma sobretaxa relativa ao retorno do carro à sua cidade de origem. A variável  $d$  representa a despesa pelo retorno do carro  $k$  alugado na cidade  $i$  e entregue na cidade  $j$ ,  $i \neq j$ ,  $d_{ij}^k \geq 0$ ,  $\forall k$ . O *tour* é iniciado e concluído na cidade em que o primeiro carro é alugado, a cidade base do caixeiro alugador. O caso em que o caixeiro realiza seu *tour* com apenas um carro alugado corresponde ao Caixeiro Viajante considerando-se as demais condições de custo associadas apenas ao carro selecionado.

Mesmo carros de características iguais e alugados em uma única rede de locação podem ser contratados sob diferentes custos, dependendo da cidade de locação e da negociação de contrato. Portanto, sem perda de generalidade, a designação de aluguel pode ser então eficazmente controlada por decisões associadas aos carros, abstraindo-se as locadoras. Presentemente o conjunto  $K = \{1, \dots, k\}$ ,  $|K| = k$  é o conjunto dos diferentes carros

que podem participar da solução. Os custos do retorno do carro alugado podem estar estritamente associados ao caminho entre a cidade de entrega e a de origem ou serem decorrentes de cálculo independente.

O objetivo do problema proposto é encontrar o ciclo Hamiltoniano que, partindo de um vértice inicial previamente conhecido, minimize a soma dos custos totais de operação dos carros utilizados no trajeto. Os custos totais do deslocamento são compostos por uma parcela que unifica o aluguel e demais despesas de trajetória de cada carro em um valor associado às arestas, e por uma parcela relativa à devolução de um veículo fora da sua cidade base, calculada para cada carro e para cada par de cidades origem/devolução existentes no ciclo.

A Figura 1 exemplifica, em um grafo completo de seis vértices, uma típica instância do CaRS. No exemplo existem três diferentes carros de aluguel, e os carros estão disponíveis para aluguel e entregues em todas as cidades do grafo. As Figuras 1(a), 1(b) e 1(c) exibem a contabilidade dos custos envolvidos no deslocamento de cada tipo de carro. Observe-se que, diferentemente do ciclo do caixeiro viajante clássico, a solução do CaRS depende da cidade escolhida para o início do *tour*, cidade base do caixeiro. Esse fato decorre da taxa de retorno poder estar vinculada tanto à cidade que inicia o ciclo, quanto ao próprio sentido de percurso deste ciclo. No exemplo nessa cidade é representada pelo vértice F.

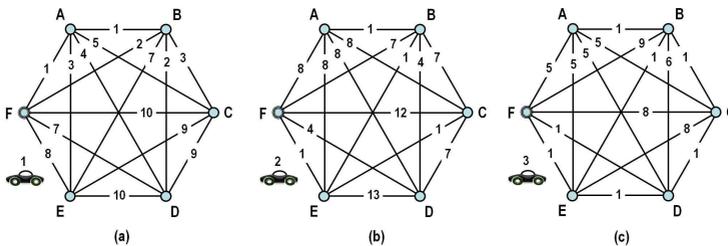


Figura 1. Custos de rotas associadas aos carros (a) 1, (b) 2 e (c) 3.

A Figura 2 mostra, para o exemplo da Figura 1, alguns dos custos do retorno dos carros a suas bases. Os custos de entrega aparecem como números sublinhados junto aos vértices. A Figura 2(a) exibe o grafo de retorno do carro 1 quando alugado no vértice F. A Figura 2(b) exibe o grafo de retorno do carro 2 quando alugado no vértice B e a Figura 2(c) o retorno do carro 3 quando alugado no vértice C. No caso geral são conhecidos os custos de retorno de todos os carros quando alugados em qualquer uma das cidades.

Uma solução do problema exemplificado nas figuras 1 e 2 é exibida na Figura 3. Essa solução considera um caso em que todos os carros disponi-

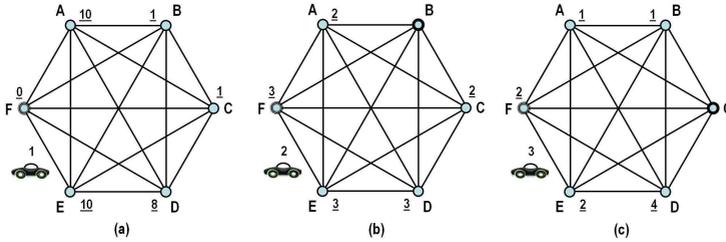


Figura 2. Custos de retorno associados aos carros (a) 1 alugado em F, (b) 2 alugado em B e (c) 3 alugado em C.

veis são alugados e não há carros alugados mais de uma vez. O custo de percorrer o ciclo, segundo o esquema de solução da figura 3, corresponde ao custo do caminho F-A-B para o carro 1, somado ao custo do caminho B-E-C para o carro 2, somado ao custo do caminho C-D-F do carro 3, em um total de 6 unidades. A este valor deve se acrescentar o custo dos retornos dos carros a suas bases. No caso do carro 1, o retorno do vértice B ao vértice F, custa uma unidade. Para o caso do carro 2, o retorno do vértice C ao vértice B custa duas unidades e, para o carro 3, o retorno ao vértice C quando o carro é entregue no vértice F custa duas unidades. Assim, a solução final custa 11 unidades.

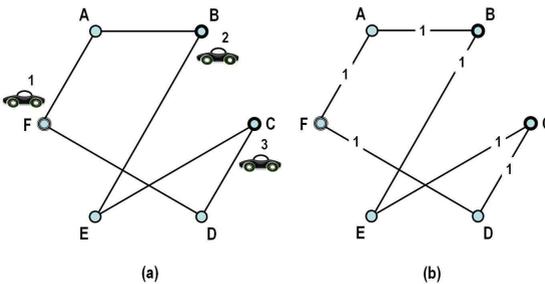


Figura 3. (a) Carros e (b) custos da rota para o exemplo proposto.

### 2.4 Variantes do problema do caixeiro alugador

O Caixeiro Alugador admite várias situações específicas de forma que pode ser classificado quanto a:

1. **Disponibilidade de carros para aluguel:** Em uma situação real não é possível considerar, de forma geral, que qualquer carro pode ser alugado em qualquer cidade. O caso em que é possível alugar todos

os carros em todas as cidades é denominado “total”. Em qualquer outro caso o problema é denominado “parcial”. O presente trabalho aborda o problema “total”.

2. **Alternativas de devolução do carro alugado:** Em uma situação real não é possível considerar, de forma geral, que qualquer carro alugado pode ser devolvido em qualquer cidade. O caso em que todas as cidades podem fazer a recepção de todos os carros é denominado de “irrestrito”. Em qualquer outra situação o problema será denominado de “restrito”. O presente trabalho aborda o problema “irrestrito”.
3. **Integridade do contrato:** Quando o problema não permitir que um mesmo tipo de carro seja alugado mais de uma vez no *tour* do alugador, o problema será denominado “sem repetição”. Nessa hipótese  $k \geq t$ , ou seja, o número de carros disponíveis para o aluguel é sempre igual ou maior que o número de carros alugados pelo caixeiro. O caso “sem repetição” em que todos os carros devam ser obrigatoriamente alugados é denominado *exato*. Nessa hipótese  $k = t$ . O problema é denominado “com repetição” em qualquer outro caso. Este trabalho aborda o problema “sem repetição”.
4. **Cálculo dos custos de devolução do carro alugado:** Os custos de devolução dos carros podem ser constituídos por valores independentes da topologia ou restrições da rede. Neste caso o problema é dito “livre”. No caso em que os custos de devolução de um carro são calculados levando em conta a rota empregada pelo carro para retornar a sua base, o problema é dito “vinculado”. Neste trabalho, quando nenhuma observação for feita em contrário, o problema é considerado “livre”.
5. **Simetria das distâncias entre as cidades:** Quando  $c_{ij}^k = c_{ji}^k$  para  $1 \leq i, j \leq n$ ,  $1 \leq k \leq ncars$ , onde *ncars* representa o número de carros disponíveis para alugar em dado caso, o problema é denominado “simétrico”. Em caso contrário é “não simétrico”. Observe-se que o problema simétrico não implica que os custos finais de operação possam ser considerados “simétricos”.
6. **Existência de ligações no grafo de conexão que modela o problema:** Quando o grafo do problema for “completo” o problema receberá o mesmo nome. O problema será denominado “incompleto” em caso contrário.

## 2.5 A dificuldade de solucionar CaRS

O problema consiste basicamente na determinação de um ciclo Hamiltoniano em um grafo  $G$  através da composição de caminhos desenvolvidos sobre os vértices de  $G$ . Considerando  $T = \{1, \dots, t\}$  o conjunto do índice

de 1 até  $t$  subgrafos  $H_i$  de  $G$ ,  $i \in T$ , e denominando por  $V(H_i)$  os vértices de  $H_i$ , os subgrafos  $H_i$  de um CaRS possuem as seguintes propriedades:

$$\bigcup_{i=1}^t V(H_i) = N \quad (1)$$

$$\left| V(H_j) \cap V(H_i) \right| \leq 1, \quad \forall j, i \in N \quad (2)$$

A restrição 1 determina que a união de todos os caminhos visite todos os vértices do grafo. A restrição 2 implica que dois diferentes subgrafos nunca possuam mais de um vértice em comum, uma condição para evitar a formação de subciclos. Observe-se que as restrições 1 e 2 ainda não são suficientes para garantir o ciclo do caixeiro alugador. É necessário ainda que os  $t$  subgrafos considerados três a três, quatro a quatro, etc, até  $t - 1$  a  $t - 1$ , também não possuam mais do que um vértice em comum.

Por se tratar de caminhos Hamiltonianos, cada caminho desenvolvido por um carro em um dos subgrafos  $H_i$  visita todos os vértices do subgrafo. O caminho do subgrafo  $H_i$  deve ser atribuído a um carro diferente dos carros atribuídos a caminhos tornados vizinhos a este mesmo caminho durante a construção do ciclo Hamiltoniano em  $G$ . Os custos das arestas de cada subgrafo correspondem aos custos de operação do carro que por ele trafega. Além disto, quando  $t \geq 2$  deve-se levar em conta no somatório dos custos totais o valor do retorno de cada carro alugado em uma cidade  $i$  e devolvido na cidade  $j$ , quando  $i \neq j$ .

O problema da determinação de ciclos e caminhos Hamiltonianos é *NP*-completo (Garey & Johnson, 1979). Pelo anteriormente exposto, a dificuldade de solução do CaRS é pelo menos tão grande quanto a do PCV. Todavia, apesar de algumas soluções do PCV também se constituírem em soluções viáveis para o CaRS, este último problema possui um número muito maior de soluções viáveis e incorpora todas as exigências do PCV, semelhantemente ao que acontece com diversas classes de problemas de roteamento de veículos – que sabidamente são problemas mais difíceis de resolver do que o PCV (Ralphs et al., 2003). O PCV é um caso particular do CaRS na situação em que existe apenas um único veículo disponível para aluguel na primeira cidade do *tour* e nenhuma possibilidade de troca deste veículo nas demais cidades. Ou ainda, a situação em que existe somente um veículo para aluguel no problema, e os custos de aluguel do mesmo independem da cidade em que é alugado, sendo as taxas de retorno maiores que zero.

## 2.6 Estado da arte da solução do problema do caixeiro alugador

O CaRS foi descrito recentemente na literatura (Goldberg et al., 2011). Neste trabalho apresenta-se tanto o modelo quanto algoritmos metaheurísticos de solução da classe *Greedy Randomized Adaptive Search* (GRASP)

(Feo & Resende, 1995), do tipo *Variable Neighborhood Search* (VNS) (Mladenović & Hansen, 1997), híbridos GRASP/VNS e um algoritmo memético. São relatados resultados para 40 instâncias disponíveis na literatura, 20 instâncias organizadas em norma Euclidiana e 20 em norma não Euclidiana. O trabalho em pauta desenvolve um experimento computacional que mostra ser o algoritmo memético estatisticamente superior aos demais, tanto no consumo de tempo de processamento, quanto na qualidade da solução alcançada.

### 3. Fundamentação da Abordagem Transgenética

Os Algoritmos Transgenéticos pertencem à classe dos algoritmos que mimetizam o processo biológico de evolução endossimbiótica intracelular. Esse tipo de evolução biológica pode ocorrer quando uma célula é invadida por outras unidades vivas de origem exógena. Se as unidades invasoras forem capazes de viverem no interior da célula hospedeiro sem a prejudicar ou até mesmo produzindo efeitos benéficos, a associação será denominada endossimbionte.

Como o fenômeno da endossimbiose ocorre no interior das células, o contato da unidade viva invasora com o citoplasma da célula hospedeiro pode desencadear complicados mecanismos de transformação genética. Tais mecanismos podem resultar em alterações genéticas (evolução) que favoreçam a *fitness* do indivíduo composto (muitas vezes denominado de “quimera”). O objetivo da evolução artificial, quando mimetizando este paradigma, é permitir constituir um sistema hospedeiro/endossimbiontes capaz de intercambiar informações genéticas até que a quimera otimize sua *fitness*, ou seja, o problema seja solucionado. Esta forma de evolução foi identificada e caracterizada pela Teoria da Evolução Endossimbiótica Serial (*Serial Endosymbiotic Theory* – SET) (Taylor, 1974; Margulis, 2004).

As transformações genéticas naturais que viabilizam fisicamente a troca de informação genética entre hospedeiro e endossimbiontes são denominadas de transferência endossimbiótica de genes (Henze et al., 2002). A transferência endossimbiótica de genes é um caso particular da Transferência Horizontal (ou lateral) de Genes (THG), mecanismo considerado, hoje, fundamental para a evolução por fornecer aos organismos acesso a genes específicos que podem enriquecer ou diversificar o repositório genético das criaturas vivas (Jain et al., 2003). Na transferência endossimbiótica de genes o DNA dos endossimbiontes tende a diminuir enquanto o do hospedeiro tende a ser enriquecido (Wernegreen, 2005).

De uma forma simplificada pode-se dizer que as trocas genéticas entre hospedeiro e endossimbiontes ocorrem através de plasmídios e transposons, os elementos genéticos móveis do fluxo intracelular (Zaneveld et al., 2008). Os plasmídios são partículas genéticas móveis, anéis de DNA, que podem se replicar independente dos cromossomos. Os transposons, ou “genes sal-

titantes” (*jumping genes*), são elementos genéticos que podem mover-se espontaneamente entre diferentes posições de uma molécula de DNA. Os transposons formam suas sequências através de dois mecanismos diferentes de edição do DNA. O primeiro permite cortar e colar trechos de DNA enquanto o segundo executa uma cópia e cola (Bouuaert & Chalmers, 2010). De fato, outros mecanismos também podem promover o transporte horizontal dos elementos móveis, todavia não serão abordados no presente trabalho.

### 3.1 Algoritmos evolucionários associados à transgenética

Até onde os autores conhecem, os algoritmos transgenéticos são a única proposta atualmente disponível na literatura da Ciência da Computação que mimetiza a evolução intracelular endossimbiótica. Todavia compartilha sua idéia cooperativa com a maioria dos tradicionais algoritmos endossimbióticos e o uso de mecanismos que mimetizam a transferência horizontal com diversas variantes de algoritmos genéticos. Apesar dos pontos em comum, observe-se que a transgenética computacional pode dispensar o uso do mecanismo da reprodução e de mutações, características que definem os algoritmos genéticos e suas variantes.

A literatura relata alguns algoritmos evolucionários que reclamam a habilidade de mimetizar a endossimbiose extracelular. Neste tipo de endossimbiose dois organismos complexos vivem um no interior do outro. Um exemplo deste tipo de endossimbiose são certos cnidários alojando algas nos seus tentáculos. Como este caso não trata de um processo de intrusão celular, tanto hospedeiro quanto endossimbionte podem preservar sua completa autonomia, raramente compartilhando material genético por via lateral. Normalmente, os algoritmos que seguem nesta linha de mimetização são classificados como algoritmos evolutivos cooperativos (Kim et al., 2006), onde a endossimbiose é imitada no contexto computacional através da formação de diferentes populações que consistem em soluções parciais do problema investigado. Cada população evolui, separadamente, por meio de um algoritmo genético padrão. Rieffel (2006) apresenta um modelo para a representação da endossimbiose baseada no encapsulamento de soluções parciais aplicável a problemas de montagem em plantas industriais. Nesta proposta o “endossimbionte” é uma solução parcial (peças já montadas) que pode ser composta com outros elementos residentes no hospedeiro (peças ainda fora da solução) visando formar uma solução viável de montagem. Kumar et al. (2006) apresentam o *Endosymbiotic-Psychoclonal Algorithm* (ESPC), onde a evolução endossimbiótica é representada pela união de dois simbiotes (soluções incompletas) para formar soluções completas. Shin et al. (2011) apresentam um algoritmo evolucionário multi-objetivo para Sistemas de Manufatura Flexível. O processo de evolução é dividido em dois níveis. O primeiro nível é composto por simbiotes que são soluções para o processamento de cada peça – soluções parciais do problema. O

segundo nível é composto por endossimbiontes que representam soluções completas do problema. A combinação dos indivíduos do primeiro nível pode gerar indivíduos para o segundo nível. Os indivíduos do primeiro nível podem ser utilizados para avaliar os indivíduos do segundo nível. [Heywood & Lichodziejewski \(2010\)](#) propõem mecanismos endossimbióticos para a representação de modelos hierárquicos.

Observa-se que os algoritmos endossimbióticos anteriormente descritos são todos associados à metáfora dos algoritmos genéticos. De alguma forma promovem a reprodução dos simbiontes/endossimbiontes (frequentemente esta literatura não faz distinção entre os termos). Neste tipo de associação a reprodução dos endossimbiontes, representada pelos operadores de reprodução dos algoritmos genéticos, é justificável em virtude da completa independência física destas unidades vivas.

Certamente a reprodução livre do endossimbionte encontra dificuldades de se tornar aceitável quando a *quimera* evolui para formar um mutualismo obrigatório que beneficia tanto o hospedeiro quanto o endossimbionte.

Já no contexto da mimetização da THG, a literatura relata um farto número de pesquisas em que a evolução clássica dos algoritmos genéticos é enriquecida ou guiada através da utilização de mecanismos de transferência horizontal de genes. A transdução viral é abordada no trabalho de [Kubota et al. \(1996\)](#). A conjugação entre cromossomos é abordada nos trabalhos de [Harvey \(2011\)](#) e [Muhlenbein & Voigt \(1995\)](#). [Perales-Graván & Lahoz-Beltra \(2008\)](#) introduziram um operador genético inspirado na conjugação exibida por uma colônia de bactérias. O operador proposto permite a redundância e a sobreposição de genes no cromossomo. A transformação é abordada por [Simões & Costa \(2001\)](#) que propõem agregar aos cromossomos da população e ao longo do processo evolucionário, fragmentos de DNA de um repositório de informações que é formado ao início do processo evolucionário e atualizado ao longo da evolução. O operador é empregado no interior de um algoritmo genético clássico em substituição ao operador de recombinação.

Algoritmos evolucionários que empregam mecanismos de transposição são propostos por [Simões & Costa \(1999\)](#) para problemas com um objetivo e por [Deb et al. \(2000, 2002\)](#) para problemas com múltiplos objetivos. [Chan et al. \(2005\)](#) simulam o processo de realocação de genes onde cada cromossomo possui pelo menos um conjunto de genes consecutivos que são transponíveis. O comprimento deste conjunto transponível pode ser variável. A alocação destes conjuntos no cromossomo pode ser designada aleatoriamente. Por outro lado, a realocação do conjunto transponível tanto pode ser realizada dentro do próprio cromossomo quanto em outros cromossomos da população. [Kumar et al. \(2007\)](#) apresentam as operações de transposição sob a ótica de macromutações e sugerem a classificação destas operações de forma semelhante às propostas por [Deb et al. \(2002\)](#) e [Chan et al. \(2005\)](#). De maneira similar [Yeung et al. \(2008\)](#) empregam

um operador de transposição que realiza operações de cortar e colar em trechos do DNA dos cromossomos que são empregadas para enriquecer um algoritmo genético aplicado à decisão multicritério no projeto de antenas.

### 3.2 Componentes básicos dos algoritmos transgenéticos – ATs

Os Algoritmos Transgenéticos (ATs) são baseados em três premissas:

1. A evolução ocorre no interior de uma célula hospedeiro que foi invadida ou que absorveu outras unidades vivas. Hospedeiro e invasores possuem seu próprio DNA.
2. A evolução do ecossistema formado hospedeiro/endossimbiontes ocorre de forma guiada e influenciada pelo DNA do hospedeiro.
3. O processo de troca de informações genéticas necessário à evolução é realizado exclusivamente através de mecanismos de transferência horizontal de genes.

Como consequência das premissas anteriores os ATs pressupõem a interação de três contextos:

- Uma população de cromossomos, denominados cromossomos endossimbiontes.
- Um hospedeiro que possui informações capazes de influenciar a evolução da população de cromossomos endossimbiontes.
- Uma população de vetores, ditos vetores transgenéticos, que transportam informação do hospedeiro para os cromossomos endossimbiontes, alterando os códigos dos endossimbiontes e, por consequência, promovendo a variação necessária ao processo de busca.

Como a endossimbiose intracelular, mesmo em estágios iniciais de evolução, praticamente não permite a reprodução do endossimbionte no interior do cromossomo, os algoritmos transgenéticos igualmente independem da reprodução e das mutações delas decorrentes para desenvolverem sua evolução artificial. Por outro lado, como hospedeiro e endossimbiontes podem possuir diferentes DNAs, o processo é capaz de utilizar informações sobre o problema e obtidas *a priori* ao desenvolvimento do processo evolucionário (DNA do hospedeiro) conjuntamente com informações obtidas durante este processo (DNA do hospedeiro e do endossimbionte). As informações *a priori* podem ser obtidas a partir de algum conhecimento prévio sobre o problema tais como, limites inferiores ou superiores, soluções heurísticas, resultados de análise estatística do problema, entre outros. As informações *a posteriori* são constituídas a partir da evolução das soluções iniciais do algoritmo. Informações obtidas *a priori* e *a posteriori* são residentes no hospedeiro.

Os vetores transgenéticos mimetizam os vetores naturais de THG e atuam segundo métodos de transcrição específicos que são definidos conforme cada caso de aplicação. Um vetor transgenético,  $\lambda$ , é uma dupla  $\lambda = (I, \Phi)$ , onde  $I$  é uma informação e  $\Phi$  é o método através do qual o agente modifica, ou manipula, o cromossomo alvo. O método  $\Phi$  é composto de procedimentos, isto é  $\Phi = (p_1, \dots, p_s)$ , onde o conjunto de procedimentos  $p_j, j = 1, \dots, s$ , define a atuação do vetor. Ao manipular um cromossomo ( $Crom$ ) o vetor transgenético provavelmente alterará o código de  $Crom$  e, por consequência, sua adequação. Em uma analogia à terminologia empregada pela microbiologia são definidos vários tipos de vetores transgenéticos, dentre eles os plasmídios, plasmídios recombinados, transposons e vírus. Na aplicação descrita neste trabalho dois vetores transgenéticos são utilizados: plasmídio e transposon, os quais são definidos na Tabela 1 que resume os procedimentos que compõem o método de manipulação destes dois vetores transgenéticos.

Tabela 1. Procedimentos utilizados por plasmídios e transposons.

Procedimento	Caracterização
$p_1$ - Ataque ( $A$ )	Define o critério de avaliação que estabelece quando um cromossomo $Crom$ é suscetível à manipulação de um vetor transgenético, $\lambda$ .
$p_2$ - Transcrição	$A : (Crom, \lambda) \leftarrow \{falso, verdadeiro\}$ Se $A : (Crom, \lambda) = "verdadeiro"$ , o procedimento define como a informação $I$ , transportada pelo vetor, será transferida para o cromossomo.
$p_3$ - Identificação	Identifica posições que serão utilizadas para limitar a operação do vetor

Um vetor  $\lambda$  é dito um plasmídio quando sua cadeia de informação  $I$  é descrita no mesmo formato que os cromossomos endossimbiontes, uma subcadeia de DNA, e seu método utiliza os procedimentos  $p_1$  e  $p_2$ . Os plasmídios obtêm suas cadeias de informação, em princípio, do hospedeiro. Os plasmídios recombinados podem obter suas cadeias em procedimentos heurísticos que combinam informação genética pré-existente com outro tipo de informação. Um vetor  $\lambda$  é dito transposon quando sua informação é uma regra de rearranjo do código dos cromossomos endossimbiontes e seu método utiliza os procedimentos  $p_1, p_2$  e  $p_3$ . Noticia-se a existência de mais um vetor transgenético, o vetor vírus. Presentemente não será descrito em virtude de não ser empregado no algoritmo objeto deste trabalho.

O Algoritmo 1, mostrado a seguir, mostra a arquitetura geral de um AT. O passo 1 cria a população de cromossomos endossimbiontes,  $Pop$ . O passo 2 carrega as informações genéticas a priori -  $IG$  no hospedeiro. Tanto a população de endossimbiontes quanto as informações ditas  $a$

*priori* também podem ser constituídas de forma aleatória, caso não sejam disponíveis. A evolução transgenética é beneficiada com informações *a priori* de boa qualidade. Entretanto, a sua ausência não impede o processo. No passo 3 são criados os vetores transgenéticos que atuarão sobre a população. O procedimento *cria\_vetores\_trans()* define a quantidade e informação transportada pelos vetores. O passo 5 seleciona um subconjunto de cromossomos da população, *SubPop*, que será alvo do ataque dos vetores transgenéticos. No passo 6 o procedimento *manipular\_cromossomos()* implementa a manipulação dos cromossomos de *SubPop* pelos vetores em *TransVet* e atualiza *NovaSubPop* com o resultado obtido. No passo 7 a população corrente é atualizada com os cromossomos manipulados. Se alguma informação julgada significativa é criada durante a manipulação dos cromossomos endossimbiontes, esta informação é guardada no repositório de informações genéticas do hospedeiro no passo 8.

*Algoritmo 1. Meta-código de um algoritmo transgenético*

1. *Pop* ← *iniciar\_população()*
2. *IG* ← *informação\_genética()*
3. **Repita**
4. *TransVet* ← *cria\_vetores\_trans(IG)*
5. *SubPop* ← *seleciona\_cromossomos(Pop)*
6. *NovaSubPop* ← *manipular\_cromossomos(SubPop, TransVet)*
7. *Pop* ← *atualiza\_pop(Pop, NovaSubPop)*
8. *IG* ← *atualiza\_ig(Pop)*
9. até atender critério de parada

## 4. Algoritmos Evolucionários na Solução do CaRS

Esta seção descreve dois algoritmos evolucionários aplicados à solução do CaRS: um algoritmo da classe memética, baseado no trabalho de Goldbarg et al. (2011), e um algoritmo transgenético, baseado nas informações constantes na Seção 3 deste capítulo.

### 4.1 Algoritmo memético

Algoritmos evolucionários baseados em uma mescla entre a evolução genética e a evolução Lamarckiana foram denominados, na computação, de Algoritmos Meméticos. Esta abordagem foi formalizada por Radcliffe & Surry (1994) e consiste, basicamente, em aparelhar um algoritmo genético com mecanismos de busca local. A literatura reporta excelentes resultados na aplicação deste tipo de algoritmo para a solução de problemas de roteamento (Prins, 2009), especialmente em virtude da existência de vizinhanças e algoritmos de busca local que se mostraram tradicionalmente eficientes para a solução destes tipos de modelos.

#### 4.1.1 Elementos básicos do algoritmo memético

O Algoritmo 2 exibe o pseudo-código do algoritmo memético desenvolvido para o CaRS por [Goldberg et al. \(2011\)](#). Os parâmetros utilizados neste trabalho são: o número de gerações ( $nOffspring$ ) igual a 30, o tamanho da população ( $sizePop$ ) igual a 100, a taxa de recombinação ( $txCros$ ) ou número de indivíduos que reproduzem a cada geração igual a 0,6 da população, a taxa de mutação ( $txMuta$ ) igual a 0,4, e taxa de renovação da população ( $txRenw$ ) de 0,15%.

*Algoritmo 2. Algoritmo Memético*

```

1.  main(nameInstance, sizePop, nOffspring, txCros, txMuta, txRenw)
2.  instanceRead(nameInstance)
3.  Pop[] ← generateInitPop(sizePop)
4.  VNDlocalSearchPhase(Pop)
5.  for i ← 1 to nOffspring do
6.    for j ← 1 to sizePop*txCros do
7.      dad, mom ← parentsSelection()
8.      son1, son2 ← Recombination(dad, mom)
9.      son1, son2 ← carsMutation(son1, son2, txMuta)
10.     VNDlocalSearchPhase(son1, son2)
11.     if son1, son2 < Pop[dad], Pop[mom]
12.       Pop[dad] ← son1, Pop[mom] ← son2
13.   generateNewIndividuals(sizePop*txRenw)
14.  return(Pop[0])

```

Os cromossomos são vetores bidimensionais com  $n$  elementos, conforme ilustrado na Figura 4(a), para o caso do exemplo da Figura 3, e na Figura 4(b), para uma instância com  $n = 11$  e 5 carros. A primeira linha do vetor corresponde aos índices dos carros que visitam as cidades. A segunda linha do vetor corresponde à sequência de cidades visitadas no *tour*. O elemento na primeira coluna e segunda linha corresponde à cidade  $i_1$ , e assim por diante. Quando um carro  $C_i$ ,  $i = 1, \dots, k$ , onde  $k$  é o número de carros do problema, viaja da cidade  $i_1$  para a cidade  $i_m$  o valor correspondente ao carro  $C_i$  é escrito na primeira linha do vetor acima da posição da cidade  $i_1$ . A última cidade visitada por um carro não recebe o índice do carro que nela termina seu caminho. Assim, as posições da primeira linha do cromossomo são anotadas pelos carros que iniciam seu caminho nas cidades registradas na respectiva coluna da primeira linha. A cidade 0 é sempre a cidade inicial do *tour* e não é representada no cromossomo como destino final, o que fica subentendido. A Figura 4(b) representa um *tour* que inicia na cidade 0 com o carro 2 passando pelas cidades 6, 4, 3 e 10 até chegar à cidade 7. Na cidade 7 o carro 2 retorna à cidade 0 e o carro 1 é alugado para prosseguir o *tour* até a cidade 9. Na cidade 9 o carro 5

é alugado e assim por diante. O carro 4, alugado na cidade 8 completa o *tour* na cidade 0.

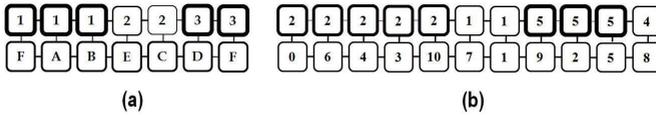


Figura 4. (a) Cromossomo da Figura 3, (b) cromossomo exemplo.

A adequação de cada cromossomo é dada pelo cálculo do inverso do valor da função objetivo. A população inicial do algoritmo é gerada pelo procedimento do vizinho mais próximo (Gutin & Punnen, 2002) adaptada ao CaRS no procedimento *generateInitPop()* que recebe o tamanho da população como parâmetro de entrada, sendo  $nCar$  o número de carros disponíveis. O procedimento determina inicialmente de forma aleatória um número de carros máximo para compor uma solução. A cidade inicial, cidade zero, é retirada da lista de cidades visitáveis. Escolhe-se aleatoriamente um carro  $c$  e uma cidade destino  $j$  para  $c$ . Constrói-se um caminho para  $c$  partindo da cidade inicial do *tour* até a cidade  $j$  através do procedimento do vizinho mais próximo. O procedimento de escolha de um carro diferente e de uma cidade alvo se repete retirando-se as cidades visitadas nos *tours* do conjunto de cidades que podem ser visitadas. Existem duas situações de parada: 1) As cidades a serem visitadas se esgotam no *tour* de um determinado carro sorteado. Neste caso a última cidade do *tour* deste último carro é ligada à cidade 0 e fim. 2) O último carro é selecionado e existem ainda cidades para serem visitadas. Nesta hipótese a heurística do vizinho mais próximo é aplicada a partir do vértice inicial do *tour* até visitar todas as cidades restantes e cair no caso 1. A população inicial do algoritmo é formada então por  $100-k$  soluções obtidas pelo método construtivo, adicionadas a  $k$  soluções obtidas da aplicação do *Concorde TSP solver* (Applegate et al., 2001) para cada carro isolado, onde  $k$  é o número de carros disponíveis.

O número de carros a ser usado na recombinação é sorteado aleatoriamente no intervalo  $[2, nCar]$  em distribuição equiprovável. O operador de recombinação é multiponto. O número de pontos a ser empregado no operador *Recombination(dad,mom)* é escolhido aleatoriamente e com igual probabilidade no intervalo  $[1, 4]$ . Após a reprodução, os filhos gerados são submetidos ao procedimento de busca local VND descrito na Seção 4.1.2. Os pais são selecionados para recombinar pelo método da roleta.

A recombinação é ilustrada na Figura 5 em uma instância com  $n = 11$  e 3 carros. Os cromossomos A e B geram os filhos C e D. A Figura 5 exibe a execução de um operador em 2 pontos. A primeira e terceira partes dos

cromossomos A e B são herdadas pelos cromossomos C e D, respectivamente. Como a recombinação pode gerar inviabilidade no cromossomo, um procedimento de reparação deve recuperar a viabilidade após a designação das rotas e carros. Por exemplo, após a recombinação de rota o cromossomo C fica formado com a sequência de visitas [0 3 1 8 10 1 9 4 5 10 6] que não é viável uma vez que repete as cidades 1 e 10 e deixa de visitar as cidades 2 e 7. Então a rota do cromossomo C é reescrita como [0 3 1 8 10 \* 9 4 5 \* 6] onde os asteriscos marcam a repetição das visitas. Cada asterisco é substituído por uma cidade ausente do *tour*, no caso as cidades 2 e 7. A linha correspondente à designação de carros após a recombinação é [1 1 1 2 3 3 2 2 2 2 3] que igualmente não é viável, uma vez que um carro somente pode ser alugado uma vez. Assim, a segunda linha do cromossomo C é reescrita como [1 1 1 2 3 3 \* \* \* \* 3] e os asteriscos são substituídos pelo carro 3. Caso houvesse dois carros diferentes nas extremidades do conjunto de asteriscos, o escolhido seria o primeiro carro no sentido do desenvolvimento do *tour*. O cromossomo D é tratado de forma similar.

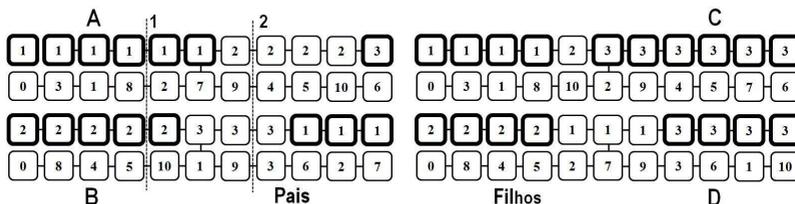


Figura 5. Operador de recombinação.

Os filhos obtidos pelas recombinações são submetidos ao procedimento de mutação. O operador de mutação  $carsMutation(son1, son2, txMuta)$  verifica quais veículos não estão representados na solução. Cada um destes veículos é inserido em um dado segmento da rota conforme será descrito adiante. O tamanho máximo de cada segmento inserido é definido na taxa de mutação. O operador é ilustrado na Figura 6, considerando-se uma instância com  $n = 11$ , e 5 carros, e a taxa de mutação associada a um segmento de tamanho máximo 3 – três cidades. No exemplo, os carros 3 e 4 não são usados na solução do cromossomo A da Figura 6. O operador de mutação insere um dos carros não usados na solução, resultando o cromossomo B. As cidades 10 e 5 são escolhidas de forma aleatória para serem iniciadas pelos carros 3 e 4, respectivamente. Assim, o veículo 5 é substituído pelo veículo 3 na cidade 10, 7 e 1, e o carro 4 substitui o carro 1 nas cidades de 5 a 8, uma vez que é necessário pelos menos duas cidades em um *tour*. A sequência de carros resultante no cromossomo é [2 2 2 5

3 3 3 5 1 4 4]. Após a mutação será necessário aplicar o procedimento de reparação para retomar a viabilidade do cromossomo, uma vez que o carro 5 é alugado duas vezes. Assim, o cromossomo é reescrito com asteriscos resultando em [2 2 2 5 3 3 3 \* 1 4 4], e o asterisco substituído por 3, o carro imediatamente anterior ao problema.

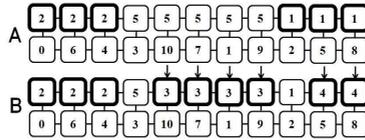


Figura 6. Operador de mutação.

Após as operações de cruzamento e mutação os filhos são submetidos à etapa da evolução Lamarckiana representada pela busca local VND apresentada na Seção 4.1.2.

### 4.1.2 A busca local do algoritmo memético

Uma forma de busca local extremamente eficiente é proposta por [Mladenović & Hansen \(1997\)](#) e denominada na literatura por *Variable Neighborhood Descent* (VND) – ou busca em vizinhança variável descendente. A composição deste tipo de busca local com o poder de diversificação dos algoritmos genéticos é promissora. Basicamente a idéia do VND é explorar sistematicamente diferentes vizinhanças de busca para aumentar o poder de intensificação do algoritmo de busca. No caso do algoritmo desenvolvido, o procedimento VND – *VNDlocalSearchPhase(Pop)* – executa a fase de busca local do algoritmo memético. O procedimento explora três diferentes vizinhanças denominadas *InvertSol*, *Insert&Saving* e *2-Shift*. *InvertSol* é um procedimento que inverte o sentido do percurso da rota, preservando o sentido de menor custo. O procedimento *Insert&Saving* busca inserir um carro na rota de forma que o custo total da rota diminua. Considere  $S$  uma solução que possui pelo menos um carro e que reste pelo menos um carro ainda não alugado em  $S$ . O procedimento *Insert&Saving* possui duas fases. Na primeira escolhe aleatoriamente um carro não designado na rota e examina a possibilidade de inseri-lo em cada uma das cidades da rota, substituindo um carro já existente, ou ocupando um trecho da rota entre dois carros já designados. Se um destes dois casos produzir uma rota de menor custo, então o cromossomo é reescrito segundo a nova configuração encontrada. O procedimento continua até que não existam mais carros a serem considerados para a inserção. Em caso de fracasso na primeira fase, o procedimento passa para a segunda fase. Nela a busca examina a possibilidade de alterar a cidade da troca de carros, sem alterar a sequência da troca de carros. Supondo que um carro  $c_1$  seja alugado na cidade  $x_1$  e

entregue na cidade  $x_2$  e um carro  $c_2$  seja alugado na cidade  $x_2$  e entregue na cidade  $x_3$ , o procedimento testa a possibilidade dos carros  $c_1$  e  $c_2$  serem trocados em todas as cidades entre  $x_1$  e  $x_3$  pertencentes à rota e diferentes de  $x_2$ . O procedimento *2-Shift* examina os caminhos de cada carro, dentro do *tour* geral, realizando os deslocamentos (*shifts*) entre uma cidade da rota e sua vizinha, para todas as cidades da rota, sem que se altere o carro de aluguel. Os caminhos dos carros são examinados em sequência. Cada vez que uma melhoria é alcançada, a solução  $S$  é atualizada.

## 4.2 O algoritmo transgenético

O algoritmo Transgenético proposto utiliza dois tipos de agentes: Plasmídeo e Transposon. O material genético do hospedeiro será constituído, inicialmente, das soluções obtidas do *Concorde TSP Solver* (Applegate et al., 2001). Observar que estas soluções são também adicionadas à população inicial do algoritmo memético. O pseudocódigo do Algoritmo 3 exhibe o procedimento principal do algoritmo transgenético desenvolvido para o Problema do Caixeiro Alugador.

A variável *informação* transporta as informações da cadeia do plasmídeo. O procedimento *loadHostRepository()* realiza a inicialização do material genético do hospedeiro, o qual é formado inicialmente também por uma solução gerada pelo *Concorde* a partir de um grafo cujas arestas constituem os menores custos individuais de cada carro. Além disto,  $nCar + 1$  soluções de melhor aptidão da população de cromossomos completam o material genético do hospedeiro, sendo que ao final de cada geração transgenética este material é atualizado com até  $nCar + 1$  soluções campeãs no procedimento *updateHostRepository()*.

Os parâmetros de entrada do algoritmo são: o número de elementos da população de cromossomos (*tamPop*) que é igual a 100, e o valor máximo de gerações *nIterations* que, sendo igual a 100, define quantas vezes a população de cromossomos é atacada pelos agentes endossimbiontes. A população inicial do algoritmo transgenético é constituída da mesma forma que a população inicial do algoritmo memético através do procedimento *generateInitPop()* já descrito anteriormente. No algoritmo transgenético proposto são utilizados tanto plasmídios quanto transposons. Os agentes plasmídios atuam na transferência de uma cadeia de informações de comprimento aleatório retirada de um dos componentes do material do hospedeiro, também escolhido em sorteio equiprovável. O plasmídeo forma sua cadeia de informação de uma parte da solução sorteada e cujo tamanho e localização no cromossomo escolhido também são definidos por sorteio. No exemplo da Figura 7, uma cadeia de tamanho três é copiada de um componente do hospedeiro. O agente plasmídeo retira do cromossomo escolhido uma sequência de cidades visitadas, bem como seus carros de aluguel associados, como mostra a Figura 7. No ataque busca-se em todas as posições possíveis de transcrição no cromossomo uma posição

de inserção desta cadeia de forma a alcançar uma melhoria no custo da solução.

*Algoritmo 3. Algoritmo Transgenético*

```

1. main(tamPop, nIterations)
2. instanceRead(nameInstance)
3. Pop[ ]  $\leftarrow$  generateInitPop(tamPop)
4. HRep[ ]  $\leftarrow$  loadHostRepository()
5. for t  $\leftarrow$  1 to nIterations do
6.   #probPlasm  $\leftarrow$  setprob(t)
7.   for z  $\leftarrow$  1 to #evstage do
8.     prob  $\leftarrow$  rand(0,1)
9.     for r  $\leftarrow$  1 to tamPop do
10.      r  $\leftarrow$  1 to tamPop do
11.        if (prob < #probPlasm)
12.          informação  $\leftarrow$  random_select(HRep)
13.          Sol  $\leftarrow$  attackPlasmid(Pop[r], informação)
14.        else
15.          Sol  $\leftarrow$  attackTransp(Pop[r])
16.        end_if
17.        Sol  $\leftarrow$  attackInvertSolTransp(Sol)
18.        if (f(Sol) < f(Pop[r]))
19.          Pop[r]  $\leftarrow$  Sol; f(Pop[r])  $\leftarrow$  f(Sol)
20.          if (f(Pop[r] < f(Pop[0]))
21.            Pop[0]  $\leftarrow$  Pop[r]; f(Pop[0])  $\leftarrow$  f(Pop[r])
22.          end_if
23.        end_if
24.      end_for
25.    updateHostRepository(Pop)
26.  end_for
27. end_for
28. Return(Pop[0])

```

Pode ser necessário reparar os cromossomos caso o método de inserção da cadeia do plasmídeo ou a manipulação do transposon transforme o cromossomo em uma solução inviável como, por exemplo, possuindo o aluguel repetido de um mesmo carro. O procedimento de reparação substitui as ocorrências de aluguel repetitivo sempre com o carro imediatamente anterior ao carro repetido. O procedimento de reparação também elimina repetição ou ausência de cidades no ciclo.

Os transposons manipulam diretamente os cromossomos da população sem depender do material do hospedeiro. No algoritmo são usados três tipos diferentes de transposons, cada um possuindo um método diferente

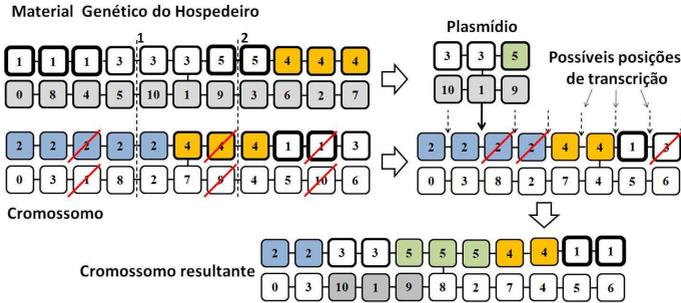


Figura 7. Funcionamento do plasmídio.

de manipulação: *2-Shift*, *InvertSol* e *Insert&Saving*. Os transposons sorteiam o trecho do cromossomo a manipular e a manipulação representa a aplicação dos procedimentos de busca local da etapa VND do algoritmo memético sobre o trecho da configuração selecionada.

O transposon *2-Shift*, gera uma vizinhança *2-Shift* conforme definido no procedimento VND, dentro do intervalo restrito de sua atuação, como exemplifica a Figura 8.

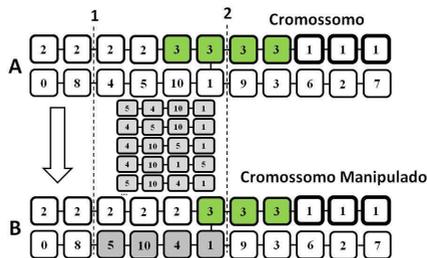


Figura 8. Funcionamento do operador *2-Shift*.

Os trechos reduzidos com 4 cidades representados na Figura 8 entre os dois cromossomos exemplificam as várias combinações possíveis para o trecho marcado entre as suas linhas tracejadas. Ao final, a sequência 5-10-4-1 é a adotada na transcrição final. Assim como ocorre com o plasmídio, as soluções geradas por transposição podem exigir reparação. O transposon *Insert&Saving* executa a mesma operação do procedimento *Insert&Saving* da fase VND do algoritmo memético, restrito ao intervalo sorteado para

sua atuação. O transposon *InvertSolTransp* inverte o sentido da rota do cromossomo manipulado, trocando as cidades de aluguel e de retorno.

Visando manter a diversidade da população os endossimbiontes que não evoluem por 15 iterações são substituídos por novos cromossomos construídos pelo procedimento de inicialização. Os ataques dos vetores transgenéticos são coordenados segundo etapas evolutivas. Em cada etapa evolucionária a probabilidade da utilização de cada vetor transgenético é alterada, de forma que ao início do processo os plasmídios que transportam informações obtidas *a priori* tenham mais chance de atuar e, ao final, os transposons sejam os de uso mais provável.

O algoritmo transgenético proposto emprega 10 etapas de evolução (variável *#evstage*). Cada etapa é composta de dez iterações evolucionárias. Na primeira etapa são utilizados somente plasmídios. Na segunda os plasmídios são sorteados com probabilidade de 90% enquanto os transposons *2-Shift* possuem 10% de chance de atuação, na terceira a probabilidade é 20-80%, na quarta 30-70%, na quinta 40-60% e na sexta 50-50%. Nas últimas 30 gerações atuam plasmídios, transposons *2-Shift* e *Insert&Saving*. Após as transcrições ou transposições o agente *InvertSolTransp* ataca o cromossomo alterado, preservando-se sua contribuição caso melhore a aptidão do cromossomo.

## 5. Resultados Computacionais

### 5.1 Metodologia

O presente tópico desenvolve um experimento computacional no sentido de validar o algoritmo transgenético proposto, e o faz através de uma comparação de desempenho com o algoritmo memético proposto por Goldbarg et al. (2011). Os experimentos foram executados em microcomputadores com processador Intel Xeon QuadCore W3520 2,8 GHz, 8GB de RAM em Linux com C++. Foram realizadas 30 execuções independentes em cada instância. São relatados resultados para instâncias disponíveis em: <http://www.dimap.ufrn.br/lae/en/projects/CaRS.php>. Observe-se que os métodos de manipulação dos vetores do algoritmo transgenético são semelhantes aos métodos de geração de vizinhança de busca do algoritmo memético e a formação da população inicial é a mesma nos dois algoritmos. Finalmente, ambos compartilham uma estratégia semelhante de melhoria da população inicial. Tais cuidados visam permitir uma avaliação melhor do método evolucionário, tentando-se isolar eventuais interferências decorrentes das estratégias de busca empregadas. No experimento é solucionado o mesmo conjunto de 40 instâncias relatados em Goldbarg et al. (2011) com o mesmo número de execuções.

## 5.2 Resultados obtidos

A Tabela 2 apresenta os resultados alcançados no conjunto de instâncias testadas em relação aos valores médios e melhores resultados das 30 execuções, considerado o tempo de execução igual ao tempo demandado pelo algoritmo transgenético. Da análise dos resultados médios da Tabela 2 observa-se que os  $p$ -valores do  $U$ -test da comparação do transgenético contra o memético indicam 30 resultados favoráveis ao algoritmo transgenético contra 7 resultados favoráveis ao algoritmo memético e 3 resultados não conclusivos. Aplicando-se o teste de proporções de Taillard et al. (2008) com o alvo de sucesso centrado na média, obtém-se um  $p$ -valor igual a 0 para a hipótese de que o algoritmo transgenético é estatisticamente superior ao algoritmo memético. Para o caso do número de melhores soluções encontradas verifica-se 11 vitórias do memético, 25 do transgenético e 4 casos inconclusivos, sustentando-se novamente a hipótese da superioridade estatística do algoritmo transgenético.

A Tabela 3 resume os resultados em relação aos valores médios e melhores resultados quando os algoritmos são executados segundo o tempo demandado pelo algoritmo memético. Da análise dos resultados médios observa-se que os  $p$ -valores indicam 25 resultados favoráveis ao algoritmo memético contra 11 resultados favoráveis ao algoritmo transgenético e 4 resultados não conclusivos. Aplicando-se o teste de comparação de proporções bicaudal de Taillard et al. (2008) obtém-se  $p$ -valor 0,00259, menor que 0,05, sustentando aqui a hipótese contrária, com a conclusão da superioridade estatística do algoritmo memético. O mesmo se verifica em relação ao número de melhores resultados alcançados.

Em relação aos tempos de execução dos algoritmos observa-se que o tempo de execução do algoritmo memético se mostra, de uma forma geral, menor que o tempo do algoritmo transgenético. Observa-se que ambos os algoritmos obtêm os melhores resultados qualitativos em seu respectivo tempo de parada, o que, provavelmente, indica que o processo de parametrização foi eficiente.

Os experimentos mostram que o algoritmo memético não se mostra capaz de aproveitar com a mesma eficiência do transgenético um eventual aumento na disponibilidade do tempo computacional. Por exemplo, na instância BrasilCO40n o algoritmo memético gasta 24 segundos para alcançar a melhor solução, que possui o valor de 627. Quando é executado por 136 segundos, reduz a solução para 597. Já o algoritmo transgenético passa de 634 aos 24 segundos para 577 aos 136 segundos. No conjunto de casos de teste examinados, em nenhum caso o algoritmo memético obtém resultados melhores em relação ao transgenético do que os obtidos no seu próprio tempo de parada. Em outras palavras, se o algoritmo memético não possui melhor resultado em seu próprio tempo de parada, no conjunto de instâncias do teste realizado não há esperança de que melhore o seu resultado relativo em relação ao algoritmo transgenético.

Tabela 2. Execuções segundo o tempo do algoritmo transgenético.

N.	Instância		Valor	Transgenético		Memético		TRG	p-val
	Nome	nCid-nCar	Melhor	Média	Melhor	Média	Melhor	T(s)	
1	BrasilRJ14e	14-2	294	294	294	294	294	2	0,83
2	BrasilRN16e	16-2	375	376	375	375	375	3	0,96
3	BrasilPR25e	25-3	508	509	508	517	509	17	0
4	BrasilAM26e	26-3	467	467	467	494	493	15	0
5	BrasilMG30e	30-4	529	548	529	555	529	30	0
6	BrasilSP32e	32-4	588	593	588	612	588	39	0
7	BrasilRS32e	32- 4	491	494	491	532	512	29	0
8	BrasilCO40e	40-5	668	678	668	752	711	84	0
9	BrasilNO45e	45-5	829	882	829	877	845	94	0
10	BrasilNE50e	50-5	756	776	756	819	766	178	0
20	Betim100e	100-3	1396	1414	1396	1429	1429	495	0
27	Vitoria100e	100-5	1354	1396	1354	1463	1381	668	0
29	PortoVelho200e	200-3	2307	2401	2307	2413	2413	4803	0,08
32	Cuiaba200e	200-3	2327	2453	2327	2495	2368	4735	0,03
41	Belem300e	300-4	2993	3081	2993	3612	3012	9492	0
54	berlin52eA	52-3	8948	8954	8948	9050	8948	76	0
63	eil76eB	76-4	1776	1849	1776	1925	1822	468	0
89	rat99eB	99-5	3105	3182	3105	3296	3223	713	0
92	rd100eB	100-4	9933	9957	9933	9951	9951	463	0,01
95	st70eB	70-4	1805	1911	1805	1932	1875	389	0,06
101	BrasilRJ14n	14-2	167	167	167	167	167	4	0,99
102	BrasilRN16n	16-2	188	191	188	189	188	7	1
103	BrasilPR25n	25-3	226	227	226	230	227	27	0
104	BrasilAM26n	26-3	202	202	202	210	207	28	0
105	BrasilMG30n	30-4	272	279	272	291	280	49	0
106	BrasilSP32n	32-4	254	258	254	269	258	70	0
107	BrasilRS32n	32-4	269	273	269	290	273	65	0
108	BrasilCO40n	40-5	577	585	577	632	597	136	0
109	BrasilNO45n	45-5	547	556	547	634	603	194	0
110	BrasilNE50n	50-5	625	639	625	693	661	380	0
122	Londrina100n	100-3	1225	1269	1225	1285	1223	1957	0
123	Osasco100n	100-4	1009	1085	1009	1212	1142	1835	0
131	Aracaju200n	200-3	1966	2097	1966	2365	2224	8058	0
135	Teresina200n	200-5	1946	2064	1946	2218	1984	3917	0
144	Curitiba300n	300-5	3005	3757	3005	3696	3362	6395	0,97
154	berlin52nA	52-3	1385	1512	1434	1448	1398	75	1
157	ch130n	130-5	2424	2778	2525	2593	2385	647	1
159	d198n	198-4	3353	4581	3353	4596	4382	7665	1
170	kroB150n	150-3	3018	3168	3018	3430	3310	9202	0
192	rd100nB	1 00-4	1485	1561	1485	1675	1556	2419	0

Tabela 3. Execuções segundo o tempo do algoritmo memético

Instância			Valor	Memético		Transgenético		MEM	p-val
N.	Nome	nCid-nCar	Melhor	Média	Melhor	Média	Melhor	T(s)	
1	BrasilR J14e	14-2	294	294	294	294	294	1	0,96
2	BrasilRN16e	16-2	375	375	375	384	375	1	1
3	BrasilPR25e	25-3	508	522	510	543	517	2	1
4	BrasilAM26e	26-3	467	494	493	499	469	3	0,98
5	BrasilMG30e	30-4	529	578	541	589	559	8	1
6	BrasilSP32e	32-4	588	621	613	635	611	7	1
7	BrasilRS32e	32-4	491	537	524	510	497	7	0
8	BrasilCO40e	40-5	668	786	726	751	685	23	0
9	BrasilNO45e	45-5	829	932	848	932	872	30	0,4
10	BrasilNE50e	50-5	756	869	814	925	840	35	1
20	Betim100e	100-3	1396	1425	1396	1437	1396	247	1
27	Vitoria100e	100-5	1354	1495	1381	1445	1354	292	0,03
29	PortoVelho200e	200-3	2307	2413	2413	2499	2393	1862	1
32	Cuiaba200e	200-3	2327	2502	2492	2580	2421	1682	1
41	Belem300e	300-4	2993	3672	3298	3265	3082	5184	0
54	berlin52eA	52-3	8948	9090	8948	8983	8948	43	0
63	eil76eB	76-4	1776	1946	1864	1926	1831	184	0,03
89	rat99eB	99-5	3105	3304	3239	3280	3196	248	0
92	rd100eB	100-4	9933	9951	9951	9989	9935	255	0,98
95	st70eB	70-4	1805	1970	1891	1989	1945	148	0,96
101	BrasilR J14n	14-2	167	167	167	172	169	1	1
102	BrasilRN16n	16-2	188	191	188	214	206	1	1
103	BrasilPR25n	25-3	226	241	229	248	233	5	1
104	BrasilAM26n	26-3	202	213	207	214	208	6	0,94
105	BrasilMG30n	30-4	272	299	286	306	295	9	1
106	BrasilSP32n	32-4	254	284	275	289	274	12	1
107	BrasilRS32n	32-4	269	311	280	305	291	13	0,01
108	BrasilCO40n	40-5	577	660	627	659	634	24	0,33
109	BrasilNO45n	45-5	547	667	613	644	614	39	0
110	BrasilNE50n	50-5	625	736	687	830	749	48	1
122	Londrina100n	100-3	1225	1338	1278	1452	1404	492	1
123	Osasco100n	100-4	1009	1259	1126	1263	1167	385	0,54
131	Aracaju200n	200-3	1966	2446	2331	2618	2430	2491	1
135	Teresina200n	200-5	1946	2241	1971	2203	2027	2514	0,02
144	Curitiba300n	300-5	3005	3726	3334	3617	3414	6599	0
154	berlin52nA	52-3	1385	1447	1385	1502	1450	86	1
157	ch130n	130-5	2424	2630	2424	2722	2535	743	1
159	d198n	198-4	3353	4665	4297	4905	4642	2756	1
170	kroBI50n	150-3	3018	3675	3491	3970	3737	1527	1
192	rd100nB	100-4	1485	1832	1681	2074	1898	464	1

## 6. Conclusões

O problema do caixeiro alugador pode representar um modelo de otimização importante para tratar situações típicas do processo de aluguel de carros, hoje um segmento de significativa importância econômica na engenharia de transportes.

No sentido de testar o desempenho de abordagens evolucionárias não clássicas no contexto do problema proposto, o presente trabalho desenvolveu um algoritmo da classe transgenética para o problema.

Os resultados obtidos no experimento apontam para a possibilidade de significativa melhoria qualitativa das soluções alcançadas em Goldberg et al. (2011), desde que seja admissível um maior consumo do tempo de processamento computacional.

Como trabalhos futuros serão desenvolvidos algoritmos segundo outras classes de metaheurísticas para a solução do CaRS. Será aperfeiçoado o ataque dos vetores transgenéticos no sentido da redução de seu tempo de execução. O algoritmo memético será modificado no sentido de alcançar uma maior diversificação na população sem penalizar proibitivamente o tempo de processamento.

## Agradecimentos

Este trabalho foi parcialmente suportado pelo CNPq e CAPES.

## Referências

- Applegate, D.; Bixby, R.; Chvátal, V. & Cook, W., TSP cuts which do not conform to the template paradigm. In: Junger, M. & Naddef, D. (Eds.), *Proceedings of Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions*. London, UK: Springer-Verlag, p. 261–303, 2001.
- Avis, , Avis Europe PLC – Financial and Strategic Analysis Review. *Global Markets Direct* 17. Acesso em Dezembro de 2010, 2009. <http://www.researchandmarkets.com/reports/690999/>.
- Avis, , Avis Budget Rental Car Funding. Acesso em Dezembro de 2010, 2010. <http://www.dbrs.com/research/232065/avis-budget-rental-car-funding-aesop-llc-series-2010-2-3/rating-report-series-2010-2-3.pdf>.
- Bouuaert, C. & Chalmers, R.M., Gene therapy vectors: the prospects and potentials of the cut-and-paste transposons. *Genetica*, 138:473–484, 2010.
- Car, , Car Rental Business – Global Strategic Business Report Global Industry Analysts. Acesso em Dezembro de 2010, 2008. <http://www.researchandmarkets.com/reports/338373/>.

- Chan, T.; Man, K.F.; Tang, K.S. & Kwong, S., Jumping gene algorithm for multiobjective resource management in wideband CDMA. *Computer Journal*, 48(6):749–768, 2005.
- Conrad, C. & Perlut, A.H., Enterprise Rent-A-Car Hits New Billion-Dollar Revenue Mark for 3rd Consecutive Year. Acesso em Dezembro de 2010, 2006. [http://www.enterpriseholdings.com/NewsReleases/Enterprise\\_FYO6\\_Sept06.pdf](http://www.enterpriseholdings.com/NewsReleases/Enterprise_FYO6_Sept06.pdf).
- Deb, K.; Agrawal, S.; Pratap, A. & Meyarivan, T., A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization: NSGA-II. In: et al., M.S. (Ed.), *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature*. Paris, France, p. 849–858, 2000.
- Deb, K.; Pratap, A.; Agrawal, S. & Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- Edelstein, M. & Melnyk, M., The pool control system. *Interfaces*, 8(1):21–36, 1977.
- Enterprise Rent-A-Car, , Enterprise Rent-A-Car Company – Strategic Analysis Review. Global Markets Direct 9. Acesso em Dezembro de 2010, 2009. <http://www.researchandmarkets.com/reports/690685/>.
- Feo, T. & Resende, M., Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.
- Garey, M. & Johnson, D., *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York, USA: Freeman, 1979.
- Goldbarg, M.; Asconavieta, P. & Goldbarg, E., Memetic algorithm for the car renter salesman problem. In: *Evolutionary Algorithms*. INTECH Open Access Publisher, 2011.
- Goldbarg, M.; Bagi, L. & Goldbarg, E.F.G., Transgenetic algorithm for the traveling purchaser problem. *European Journal of Operational Research*, 119(1):36–45, 2009.
- Gutin, G. & Punnen, A., *The Traveling Salesman Problem and Its Variations*. Springer, 2002.
- Harvey, I., The microbial genetic algorithm. In: *Advances in Artificial Life. Darwin Meets von Neumann*. Heidelberg, Germany: Springer-Verlag, v. 5778 de *LNCS*, p. 126–133, 2011.
- Henze, K.; Schnarrenberger, C. & Martin, W., Endosymbiotic gene transfer: A special case of horizontal gene transfer germane to endosymbiosis, the origins of organelles and the origins of eukaryotes. In: Syvanen, M. & Kado, C. (Eds.), *Horizontal Gene Transfer*. London, UK: Academic Press, 2a edição, p. 351–362, 2002.

- Hertz, , The Hertz Corporation – Strategic Analysis Review. Global Markets Direct 12. Acesso em Dezembro de 2010, 2009. <http://www.researchandmarkets.com/reports/690555/>.
- Hertz, A.; Schindl, D. & Zufferey, N., A solution method for a car fleet management problem with maintenance constraints. *Journal of Heuristics*, 5:425–450, 2009.
- Heywood, M. & Lichodziejewski, P., Symbiogenesis as a mechanism for building complex adaptive systems: a review. In: *Proceedings of AAAI Symposium on Complex Adaptive Systems*. p. 1–12, 2010.
- Jain, R.; Rivera, M.; Moore, J. & Lake, J., Horizontal gene transfer accelerates genome innovation and evolution. *Molecular Biology and Evolution*, 20(10):1598–1602, 2003.
- Kim, K.; Kim, J. & Kim, Y., An endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines. *European Journal of Operational Research*, 168(3):838–852, 2006.
- Kubota, N.; Shimojima, K. & Fukuda, T., The role of virus infection in virus-evolutionary genetic algorithm. *Journal of Applied Mathematics and Computer Science*, 6(3):415–429, 1996.
- Kumar, P.; Gospodaric, D. & Bauer, P., Improved genetic algorithm inspired by biological evolution. *Soft Computing*, 11(10):923–941, 2007.
- Kumar, V.; Kumar, P.; Tiwari, M.K. & Chan, F., Stochastic make-to-stock inventory deployment problem: an endosymbiotic psychoclonal algorithm based approach. *International Journal of Production Research*, 44(11):2245–2263, 2006.
- Li, Z. & Tao, F., On determining optimal fleet size and vehicle transfer policy for a car rental company. *Computers & Operations Research*, 37(2):341–350, 2010.
- Margulis, L., Serial endosymbiotic theory (set) and composite individuality. *Microbiology Today*, 31:172–174, 2004.
- Mladenović, N. & Hansen, P., Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
- Moscato, P., *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithm*. Technical Report 826, Pasadena, USA, 1989.
- Muhlenbein, H. & Voigt, H.M., Gene pool recombination in genetic algorithms. In: Eshelman, L. (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*. San Mateo, USA: Morgan Kaufmann, p. 104–113, 1995.
- Pachon, J.; Iakovou, B.; Ip, C. & Aboudi, R., A synthesis of tactical fleet planning models for the car rental industry. *IIE Transactions*, 35(9):907–916, 2003.

- Perales-Graván, C. & Lahoz-Beltra, R., An AM radio receiver designed with a genetic algorithm based on a bacterial conjugation genetic operator. *IEEE Transactions on Evolutionary Computation*, 12(2):1–29, 2008.
- Prins, C., Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22(6):916–928, 2009.
- Radcliffe, N. & Surry, P., Formal memetic algorithms. In: Fogarty, T. (Ed.), *Evolutionary Computing*. Heidelberg, Germany: Springer-Verlag, v. 865 de *LNCS*, p. 1–16, 1994.
- Ralphs, T.; Kopman, L.; Pulleyblank, W. & Trotter, L., On the capacitated vehicle routing problem. *Mathematical Programming*, 94:343–359, 2003.
- Rieffel, J., *Evolutionary Fabrication – The Co-Evolution of Form and Formation*. PhD thesis, The Faculty of the Graduate School of Arts and Sciences, Brandeis University, Waltham, USA, 2006.
- Shin, K.; Park, J. & Kim, Y., Multi-objective fms process planning with various flexibilities using a symbiotic evolutionary algorithm. *Computers & Operations Research*, 38(3):702–712, 2011.
- Simões, A. & Costa, E., Enhancing transposition performance. In: Press, I. (Ed.), *Proceedings of the 1999 Congress on Evolutionary Computation*. Piscataway, USA, 1999.
- Simões, A. & Costa, E., On biologically inspired genetic operators: Transformation in the standard genetic algorithm. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. p. 584–591, 2001.
- Taillard, E.; Waelti, P. & Zuber, J., Few statistical tests for proportions comparison. *European Journal of Operational Research*, 185(3):1336–1350, 2008.
- Taylor, F., Implications and extensions of the serial endosymbiosis theory of the origin of eukaryotes. *Taxon*, 23(2/3):229–258, 1974.
- Wernegreen, J., For better or worse: genomic consequences of intracellular mutualism and parasitism. *Current Opinion in Genetics & Development*, 15(6):572–583, 2005.
- Yang, Y.; Jin, W. & Hao, X., Car rental logistics problem: a review of literature. In: *Proc. of IEEE International Conference on Service Operations and Logistics, and Informatics*. p. 2815–2819, 2008.
- Yeung, S.; Ng, H. & Man, K.F., Multi-criteria design methodology of a dielectric resonator antenna with jumping genes evolutionary algorithm. *AEU - International Journal of Electronics and Communications*, 62(4):266–276, 2008.

Zaneveld, J.; Nemergut, D. & Knight, R., Are all horizontal gene transfers created equal? prospects for mechanism-based studies of hgt patterns. *Microbiology*, 154(1):1–15, 2008.

---

## Notas Biográficas

**Marco Cesar Goldberg** é Engenheiro de Fortificação e Construção pelo Instituto Militar de Engenharia (1982), mestrado em Sistemas e Computação pelo Instituto Militar de Engenharia (1987), doutorado em Sistemas e Computação pela COPPE/UFRJ (1990) e pós-doutorado em Ciência da Computação pela UFMG (1999). Atualmente é professor associado na Universidade Federal do Rio Grande do Norte.

**Paulo Henrique Asconavieta da Silva** tem mestrado em Tecnologia (UTFPR, 2008) e é Professor do Departamento de Computação do Instituto Federal de Educação, Ciência e Tecnologia Sul-Riograndense (IFSUL). Atualmente é doutorando no programa de pós-graduação em Sistemas e Computação da Universidade Federal do Rio Grande do Norte (UFRN).

**Elizabeth F. G. Goldberg** tem graduação em Engenharia Industrial pelo Centro Federal de Educação Tecnológica – Celso Suckow da Fonseca (1985), mestrado em Sistemas e Computação pelo Instituto Militar de Engenharia (1993), doutorado em Engenharia de Produção pela COPPE/UFRJ (2001) e pós-doutorado pela COPPE/UFRJ, Sistemas e Computação (2006). Atualmente é professora associado na Universidade Federal do Rio Grande do Norte.

### **Algoritmo Micro-genético Aplicado ao *Scheduling* de uma Rede de Distribuição de Derivados de Petróleo**

Henrique Westphal, Flávio Neves-Jr e Lúcia Valéria R. de Arruda\*

---

**Resumo:** Este capítulo modela o problema de distribuição de derivados de petróleo através de redes de dutos como um problema de otimização multiobjetivo. A rede estudada é composta por duas fontes, dois nós intermediários com capacidade de estoque e três terminais. O principal objetivo é satisfazer a demanda de produtos nos terminais consumidores de acordo com um horizonte de tempo determinado a priori, garantindo ainda que uma produção mínima seja enviada das fontes. Um conjunto de restrições operacionais tais que capacidade do parque de armazenamento, estoques, vazão dos dutos e outros devem ser respeitados. Propõe-se a utilização de algoritmos genéticos com elitismo e pequena população (algoritmo micro-genético) na solução do problema. O modelo desenvolvido e resultados simulados são apresentados para validar o algoritmo proposto.

**Palavras-chave:** Otimização multiobjetivo, Algoritmo micro-genético, Escalonamento, Rede de dutos.

**Abstract:** *In this chapter, distribution of derived petroleum products through pipelines networks is approached as a multiobjective optimization problem. The studied pipe network contains two sources, two intermediate connections with storage capacity and three consumer terminals. The main goal is to satisfy the demanded products at the destination points in a due time while assuring that a minimal production is delivered by each source. Moreover, a number of operational constraints must be satisfied, such as storage capacity limits, product inventories, pipe flow rates and others. The proposed methodology uses a genetic algorithm with small population and elitist strategy to find a set of feasible solutions. The developed model and simulated results are presented in order to validate the proposed Micro Genetic Algorithm.*

**Keywords:** *Multiobjective optimization, Micro-genetic algorithm, Scheduling, Pipelines network.*

---

\*Autor para contato: [lvrarruda@utfpr.edu.br](mailto:lvrarruda@utfpr.edu.br)

## 1. Introdução

A diminuição de custos de produção e a melhoria em produtos e serviços são objetivos comuns a diversos setores industriais. A complexidade do planejamento (*planning*) e programação (*scheduling*) da produção na indústria é em geral, contornada pela adoção de políticas operacionais conservativas, que não utilizam a capacidade máxima de operação do sistema produtivo. A necessidade de modelagem das estruturas industriais e a busca pela eficiência no transporte de derivados de petróleo determinam a execução deste trabalho. O objetivo principal é a modelagem das operações envolvidas no gerenciamento (*scheduling*) de operações de uma rede de distribuição de derivados de petróleo. Estas redes são compostas por um número de refinarias e terminais, ou áreas, interligadas por um conjunto de polidutos, ou trechos de dutos, os quais operam o transporte de produtos (derivados de petróleo e produtos orgânicos) entre áreas adjacentes. A operação da rede deve considerar restrições físico-químicas de contato entre produtos distintos, prioridade no envio de determinados produtos e preferências operacionais no bombeamento de bateladas com maior volume, atendimento a demanda de mercado, buscando-se sempre a solução que apresenta o menor custo operacional.

Esta é uma atividade importante na economia de um país e, por ser complexa, demanda cuidados na sua realização. Considerando que a taxa de ocupação das redes está cada vez mais elevada, faz-se necessário uma política ótima de utilização dos recursos. Uma melhor eficiência no uso da rede, quando possível, é uma solução mais barata e mais rápida que investimentos na ampliação da malha de distribuição. De acordo com [Neves-Jr et al. \(2007\)](#), é um grande desafio a busca por modelos de *scheduling* para redes de dutos complexas integrando diversas áreas e transportando diferentes produtos que possam ser implementados na prática, considerando uma carga computacional aceitável. Atualmente a programação das operações em redes reais é realizada por especialistas experientes, entretanto técnicas de pesquisa operacional têm se tornado uma ferramenta eficiente na otimização destes processos operacionais, auxiliando na modelagem e resolução de problemas combinatórios ([Relvas et al., 2006](#)). Por outro lado, nas últimas décadas, técnicas metaheurísticas baseadas em população têm sido amplamente estudadas e utilizadas na resolução de vários problemas complexos de *scheduling* de sistemas produtivos e mostram ser muito eficientes na busca de soluções satisfatórias em um tempo mais curto ([Hertz & Widmer, 2003](#); [Dréo et al., 2006](#)). Através da análise destes trabalhos, em especial dos que utilizam algoritmos genéticos em problemas de *scheduling* de atividades de produção, verificou-se que uma abordagem por AG multiobjetivo pode ser uma alternativa para a solução deste tipo de problema que, classicamente, demanda o uso de *solvers* de programação inteira mista de larga escala ([Shah, 1998](#)).

Neste contexto, este trabalho propõe então uma metodologia baseada em algoritmo genético multiobjetivo para realizar o *scheduling* de uma rede de distribuição de derivados de petróleo.

Um problema de otimização multiobjetivo envolve a minimização (ou maximização) de vários critérios conflitantes que não podem ser satisfeitos simultaneamente. Nestes casos, procura-se um conjunto de soluções ótimas em que nenhuma solução do conjunto pode ser considerada superior às outras quando todos os critérios são analisados separadamente. Este conjunto é denominado conjunto ótimo de Pareto e as soluções que o compõem são conhecidas como soluções não dominadas (Coello, 2003).

Uma revisão bibliográfica e comparativa dos vários métodos evolucionários existentes na literatura, para a resolução de problemas multiobjetivo (*Multi-Objective Evolutionary Algorithm* – MOEA) pode ser encontrados em Coello (2003). No entanto, apesar da variedade de métodos existentes e das melhorias alcançadas nos últimos anos (Dréo et al., 2006), a eficiência computacional dos métodos MOEA tem limitado o seu uso na solução de problemas combinatórios de grande porte como é o caso do problema de *scheduling* da rede de dutos. É sabido que os dois principais processos que demandam tempo computacional em um algoritmo MOEA são: o ranqueamento da população que requer um mecanismo de comparação entre todos os indivíduos a fim de determinar sua dominância e um mecanismo para manter a diversidade a fim de evitar a convergência prematura do algoritmo devido à reprodução preponderante de super-indivíduos (Coello & Pulido, 2001).

Para contornar estes problemas, propõe-se o uso de um algoritmo genético com uma população pequena e um processo de reinicialização com uma memória externa para armazenar as soluções não dominadas previamente encontradas. Esta classe de algoritmo genético é denominada na literatura de algoritmo micro-genético (Coello & Pulido, 2001). Além disto, no algoritmo micro-genético proposto é incluído um mecanismo de formação de nichos capaz de manter várias soluções diferentes por um período de tempo infinito em relação ao tamanho da população garantindo, assim, a diversidade da população. O algoritmo micro-genético assim desenvolvido é utilizado para calcular o *scheduling* de uma rede de distribuição de derivados de petróleo.

Este artigo é composto por sete seções. A Seção 2 traz um breve estado da arte sobre os trabalhos relacionados às redes dutoviárias, com especial atenção para as soluções de *scheduling* que usam técnicas evolucionárias. O algoritmo micro-genético proposto é desenvolvido na Seção 3. A rede dutoviária estudada e o problema de distribuição de produtos a ela associado são discutidos respectivamente nas Seções 4 e 5. A Seção 6 apresenta os resultados alcançados e, por fim, a Seção 7 expõe as conclusões deste trabalho.

## 2. Algoritmos Genéticos Aplicados ao *Scheduling* de Dutos

Vários métodos metaheurísticos, em especial aqueles baseados em AG, têm sido utilizados para resolver problemas combinatórios em aplicações de *scheduling* (Hertz & Widmer, 2003; Dréo et al., 2006; Souza-Filho et al., 2006). Entretanto, somente um número limitado de artigos refere-se ao *scheduling* de um poliduto ou de redes de dutos (Sasikumar et al., 1997; Crane et al., 1999; Garcia et al., 2004; Alves & Ferreira-Filho, 2007; Arruda et al., 2010).

O problema de *scheduling* mensal de uma rede de dutos para distribuição de produtos derivados do petróleo foi inicialmente proposto por Sasikumar et al. (1997). Estes autores desenvolveram um método heurístico cuja solução consiste de quatro componentes: o produto a ser enviado, sua quantidade, o modal de transporte e a sequência de envio. A solução leva em consideração restrições do tamanho da batelada, do sequenciamento de envio, e de tangagem na origem e no destino. Também são calculadas as datas de envio e chegada dos produtos nas diferentes localidades. As soluções obtidas foram consideradas aceitáveis pelos usuários para um período de 30 dias.

Crane et al. (1999) apresentaram um modelo aplicando algoritmo genético para a resolução do problema simplificado de *scheduling* de dutos. A rede estudada é composta por 8 nós terminais com 2 tanques em cada, ligados por 7 dutos unidirecionais em que não é permitido a reversão, 2 produtos e 3 níveis de capacidade de armazenamento dos produtos nos tanques. A solução ótima para este problema foi obtida, mostrando, assim, a possibilidade de solução de um problema complexo como o *scheduling* de dutos através da aplicação de algoritmo genético.

Em Cruz et al. (2003) um modelo para otimização multiobjetivo baseado em algoritmo evolucionário com restrições é proposto para o problema de distribuição de produtos derivados do petróleo através de uma malha de dutos. O modelo foi aplicado em um problema simplificado, com o objetivo principal de satisfazer a demanda de produtos em um tempo mínimo, minimizando a interface entre diferentes produtos.

Garcia et al. (2004) usaram duas técnicas para a resolução do problema de *scheduling* de uma rede de dutos: um método heurístico do tipo algoritmo evolucionário multiobjetivo, e programação matemática usando o método das restrições na programação linear inteira mista. Seus resultados mostraram que os métodos evolucionários facilmente solucionam problemas de otimização combinatória não-lineares. São capazes de evitar ótimos locais, e ambas as técnicas alcançaram soluções similares. Entretanto, o uso do método híbrido, que combina programação linear e algoritmo evolucionário encontrou melhores soluções em um tempo menor de execução, se consideradas ambas as técnicas separadamente.

Em [Alves & Ferreira-Filho \(2007\)](#), um modelo baseado em algoritmo genético para a transferência de petróleo em rede de dutos é proposto, considerando a disponibilidade de estoques no porto, a programação dos cortes de petróleo na refinaria, o tamanho da batelada e o sequenciamento de envio. A solução calculava um *scheduling* viável para um horizonte de tempo de 7 a 14 dias. No entanto considerações físicas de operação tais que compatibilidade e restrições de temperatura, além de paradas de bombeio não são levadas em conta na solução apresentada.

[Arruda et al. \(2010\)](#) desenvolveram uma abordagem semelhante à apresentada neste trabalho em que utiliza um algoritmo genético multiobjetivo (MOGA) para estabelecer o *scheduling* mensal de uma rede de dutos. No entanto, por apresentar uma função de adequabilidade modelada em programação linear inteira mista, o custo computacional da solução apresentada se torna inaceitável para aplicações reais.

Tentando amenizar os problemas de carga computacional acima citados, é que se propõe a utilização de um algoritmo micro-genético na solução do problema. Tem sido relatado na literatura que soluções baseadas em algoritmo micro-genético permitem a redução do tempo de processamento enquanto mantêm uma precisão aceitável, se comparado com soluções obtidas via AG padrão ([Coello & Pulido, 2001](#); [Chen et al., 2007](#); [Carpinelli et al., 2010](#)). Esta redução do tempo computacional deve-se ao fato que o pequeno número de indivíduo na população reduz o esforço para avaliação e ranqueamento de cada indivíduo. Além disto, utiliza-se o método do critério global para avaliar cada indivíduo, evitando a formação de múltiplas populações comum aos métodos MOEA ([Coello, 2003](#); [Arruda et al., 2008](#)). Também se propõe uma aproximação do método da ponderação de objetivos para gerar prioridades entre os objetivos e preencher a memória externa do algoritmo com o conjunto de soluções não dominadas.

### 3. Algoritmo Micro-Genético com Nichos e Castas

O algoritmo micro-genético foi proposto por [Krishnakumar \(1989\)](#) para o caso monobjetivo e mais tarde estendido para o caso multiobjetivo em ([Coello & Pulido, 2001](#)). Em ambos os artigos, os algoritmos micro-genéticos desenvolvidos convergiram para a fronteira de Pareto de modo mais rápido que suas contrapartidas padrão (AG) que foram analisadas. Segundo estes trabalhos, a garantia da convergência do algoritmo micro-genético está no uso de estratégias de reinicialização da população após poucas gerações, a fim de assegurar a diversidade da população e evitar convergência prematura. Em especial, o algoritmo proposto por [Coello & Pulido \(2001\)](#) utiliza duas memórias auxiliares para esta reinicialização. A primeira é denominada memória populacional e é usada como fonte de diversidade na reinicialização das populações. A segunda é uma memória externa que é utilizada para guardar o conjunto de soluções de Pareto encontradas ao

longo da evolução. Além disto, a memória populacional é dividida em duas partes, uma substituível e outra não substituível que se mantém ao longo das gerações, assegurando a diversidade da população. Ainda, segundo [Coello & Pulido \(2001\)](#), para favorecer a busca ao longo das gerações em diferentes pontos na fronteira de Pareto, técnicas elitistas variadas devem ser utilizadas. Assim, para cada reinicialização, as soluções não-dominadas armazenadas na memória externa devem ser selecionadas para fazer parte da população não substituível de modo que, ao final, a população será exclusivamente formada por soluções não-dominadas. Apesar dos mecanismos de formação de população, um algoritmo micro-genético funciona exatamente igual a um algoritmo genético padrão através de procedimentos de seleção e aplicação de operadores de recombinação e mutação ([Goldberg, 1989](#)).

A partir destas idéias, este trabalho apresenta uma adaptação do algoritmo genético multi-objetivo com formação de nichos e castas proposto por [Westphal \(2006\)](#); [Arruda et al. \(2008\)](#) para uma arquitetura micro-genética. Este algoritmo apresenta um método de seleção baseado na técnica elitista de ranqueamento e formação de nichos ([Goldberg, 1989](#)), mas com algumas particularidades. Neste caso, o elitismo mantém determinado número de cópias de bons elementos dentro de uma casta isolada do programa principal (casta elitista). Estes indivíduos não participam do processo de seleção e não sofrem alterações dos operadores genéticos. Além disto, dentro desta casta não existem soluções repetidas ([Westphal, 2006](#)). A casta elitista guarda semelhanças com a memória não-substituível do trabalho de ([Coello & Pulido, 2001](#)).

As operações genéticas ocorrem somente no corpo principal do algoritmo genético, no qual existem cópias dos elementos guardados na casta elitista. A casta elitista tem também a função de manter os melhores indivíduos encontrados durante toda a evolução, mesmo no momento da convergência total de um único elemento nas demais castas da população. Para escolha dos indivíduos em cada população foi empregado o método de seleção por torneio, com elitismo de 10% do tamanho da população. Foram determinadas quatro castas distribuídas conforme a [Figura 1](#). A casta elitista é composta por 10% do tamanho da população, e abriga os melhores elementos da geração anterior, sem que haja repetição de uma mesma solução. A casta de reprodução é formada pela reprodução dos 20% melhores elementos da população corrente, podendo haver repetições. A casta intermediária é formada pela transmissão direta de 50% da população corrente referente aos indivíduos com nível de aptidão intermediário. A última casta é composta pelas soluções menos adequadas da população corrente, que sofrem um processo de eliminação e são transmitidas apenas em parte para a nova geração. As porcentagens determinando o tamanho das castas foram escolhidas experimentalmente. A [Figura 1](#) também apre-

	População corrente		Próxima geração	Casta
Maior nível de aptidão ↑			10%	Casta Elitista
	20%	→	25%	Casta de reprodução
	50%	→	50%	Casta intermediária
	30%	→	15%	Casta de eliminação

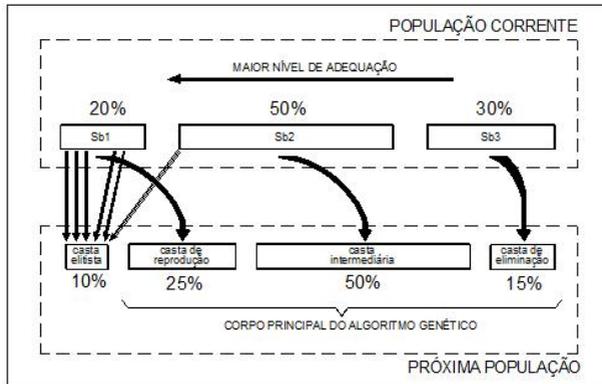


Figura 1. Distribuição da população e construção das castas.

senta o procedimento utilizado pelo algoritmo genético para a construção das castas descritas.

De acordo com a Figura 1, primeiramente os indivíduos da população corrente são avaliados e em seguida ordenados de acordo com seus níveis de aptidão. A população é então, subdividida em três partes correspondentes a 20, 50 e 30% do tamanho total da população. A seguir, o algoritmo repassa os indivíduos com melhor nível de aptidão para a casta elitista da população seguinte, até completar o número de 10% do tamanho da população, sem que haja indivíduos iguais. Como pode existir indivíduos repetidos dentro da subdivisão *Sb1* (20% melhores da população corrente), a casta elitista pode ser formada por elementos das outras subdivisões. Posteriormente os indivíduos da subdivisão *Sb1* são reproduzidos através da atuação da seleção por torneio formando a casta de reprodução. Os indivíduos da subdivisão *Sb2* são repassados diretamente para a casta intermediária, e, finalmente, alguns indivíduos da subdivisão *Sb3* sofrem eliminação, também através da utilização da seleção por torneio novamente, gerando a casta de eliminação. Após este procedimento, são utilizados os operadores genéticos de mutação e recombinação somente no corpo principal do algoritmo genético (Figura 1), para a finalização da

construção da nova geração. Esta nova geração tem todos os seus indivíduos avaliados, ordenados e posteriormente subdivididos nas três partes (*Sb1*, *Sb2* e *Sb3*) para a continuação do ciclo de busca.

Com a utilização da metodologia apresentada, ao final da evolução, são disponíveis na casta elitista, não apenas um, mas vários indivíduos que formam o conjunto das melhores soluções encontradas pelo algoritmo genético. Neste conjunto podem existir ou não soluções de Pareto. Para descobri-las é aplicado o algoritmo proposto por Deb (1999), o qual rotula as soluções em dominadas e não-dominadas. Após esta rotulação, as soluções não-dominadas são enviadas para uma memória externa do algoritmo, de onde serão selecionadas algumas soluções para a reinicialização completa da população do algoritmo micro-genético que ocorre após uma convergência parcial.

De acordo com Carpinelli et al. (2010) nos algoritmos micro-genéticos duas condições podem ser testadas para determinar a reinicialização das populações:

- o controle do melhoramento da adequabilidade das soluções (*fitness*) existentes na população corrente.
- um número máximo de indivíduos gerados e/ou de gerações pré-estabelecidas.

Neste artigo adotou-se a segunda estratégia, e após um número fixo de gerações, a população é completamente substituída. Para esta substituição, as subdivisões *Sb2* e *Sb3* são aleatoriamente inicializadas, enquanto a subdivisão *Sb1* recebe elementos da memória externa, os quais são soluções não-dominadas. O algoritmo micro-genético pára após um número fixo de reinicializações e, como resultado final, a memória externa possui um conjunto de soluções não-dominadas bem distribuídas ao longo da fronteira de Pareto do problema.

#### 4. Rede de Dutos

O modelo utilizado neste trabalho é uma simplificação de uma rede real e é mostrado na Figura 2. Este modelo é composto por um conjunto de fontes (refinarias) representadas na Figura 2 pelos nós *N1* e *N2*, um conjunto de terminais (pontos de destino) representados pelos nós *N5*, *N6* e *N7*, e um conjunto de tanques intermediários que podem ser receptores ou fornecedores de produtos (parque de armazenagem) representados pelos nós *N3* e *N4*. Ainda na Figura 2, as linhas que unem os nós representam os dutos e as flechas representam a direção em que flui o produto transportado. Ambos os nós, *N1* e *N2*, fornecem produtos para *N3* e *N4* através das conexões *D1*, *D2*, *D3* e *D4*. Estas conexões são unidirecionais, ou seja, o produto flui somente na direção da fonte para os nós intermediários. Os nós *N3* e *N4* são conectados através de um poliduto bidirecional *D5* e *D8*.

A conexão  $D6$  e  $D7$  conectam os nós  $N5$  e  $N6$  a  $N3$  respectivamente. As conexões  $D9$  e  $D10$  conectam os nós  $N6$  e  $N7$  a  $N4$  respectivamente. Os produtos produzidos nas fontes  $N1$  e  $N2$  são designados de forma genérica como produto  $P1$ ,  $P2$ ,  $P3$  e  $P4$ . O produto que cada fonte produz é configurado conforme desejado. Por exemplo, a fonte  $N1$  pode ser configurada para produzir produtos do tipo 1 e 2 e a fonte  $N2$  para produzir produtos do tipo 3 e 4.

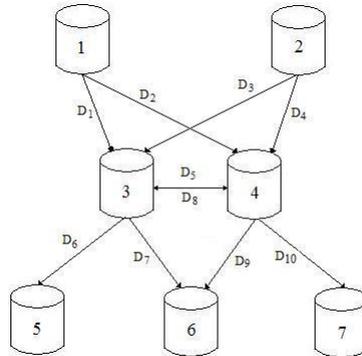


Figura 2. Rede estudada.

Considera-se a tancagem agregada por produto, isto é, cada nó intermediário e/ou terminal possui a quantidade de tanques necessária para receber cada produto. Ou seja, se em um nó é possível receber quatro tipos de produtos, então se assume que neste nó existam pelo menos quatro tanques para a armazenagem destes produtos.

O modelo assume uma discretização uniforme do tempo e considera ainda que os produtos são entregues na forma de bateladas discretas. Uma batelada representa um volume mínimo de um produto a ser transportado em uma unidade de tempo. Uma batelada unitária é um volume mínimo que preenche um duto. Cada conexão possui uma distância normalizada em termos de unidades de tempo necessárias para que uma dada batelada seja transportada de um ponto a outro. Com isto o tempo de transporte de uma batelada de um nó a outro na rede é um número inteiro e, os tempos envolvidos em todas as conexões são mostrados na Tabela 1.

A solução para o problema é dada por cada tipo de batelada que é enviada em cada instante de tempo em cada conexão da rede. Os programadores de *scheduling* da rede devem atender aos seguintes objetivos:

- Demanda de fornecimento: definida como a quantidade mínima de bateladas de cada produto que deve ser enviada de uma refinaria,

Tabela 1. Matriz de tempos de percursos (conexão) na rede.

Conexão	Nó de origem	Nó de destino	Tempo
1	1	3	1
2	1	4	3
3	2	3	3
4	2	4	2
5	3	4	3
6	3	5	3
7	3	6	4
8	4	3	2
9	4	6	3
10	4	7	2

evitando assim a saturação dos seus tanques de armazenagem. Não é desejável paralisar a produção por falta de parque de armazenagem.

- Demanda de recebimento: definida como a quantidade de bateladas de cada produto que deve ser entregue aos pontos solicitantes (terminais).
- Minimização do tempo para atendimento às demandas, este tempo é denominado horizonte de programação.
- Minimização da fragmentação das bateladas nas conexões: evitar a alternância no envio das bateladas dos produtos, ou seja, tentar enviar sempre sequências de bateladas do mesmo produto. A alternância no tipo de produto a ser enviado pode provocar contaminações nas fronteiras do produto enviado e por isto a fragmentação de bateladas não é desejada

#### 4.1 Representação da rede no AG

Uma matriz onde cada linha representa um indivíduo é usada para codificar a população. Cada indivíduo é composto por instantes de tempo, que por sua vez contém o valor da batelada (tipo de produto transportado) de todas as conexões naquele instante. Por exemplo, a Figura 3 mostra um indivíduo que representa o estado da rede em 10 instantes de tempo: os primeiros  $n$  elementos representam as bateladas enviadas através das  $n$  conexões no primeiro instante de tempo, os  $n$  elementos seguintes representam as bateladas para o segundo instante de tempo e assim sucessivamente. Os valores estão no intervalo  $[0; \text{numerodeprodutos}]$  onde  $\theta$  representa uma conexão vazia.

Para aproximar o modelo da situação real, a fonte representada pelo nó  $N1$  pode produzir tipos de produtos diferentes de  $N2$ . Desta forma, algumas conexões não podem assumir determinados valores. Uma tabela

TEMPO	Instante 1	....	Instante 10
CONEXÃO	1 ... 10	....	1 ... 10
SOLUÇÃO	1 0 1 2 2 3 3 0 2 3	....	2 2 2 2 2 3 3 4 2 2

Figura 3. Codificação do AG.

de compatibilidade de produtos por conexão, respeitando as produções nas refinarias foi implementada (Tabela 2), onde *x* indica que o produto não pode passar na conexão.

Tabela 2. Matriz de compatibilidade de produtos por conexão.

Conexão	P1	P2	P3	P4
1	1	2	x	x
2	1	2	x	x
3	x	x	1	2
4	x	x	1	2
5	1	2	3	4
6	1	2	3	4
7	1	2	3	4
8	1	2	3	4
9	1	2	3	4
10	1	2	3	4

### 5. Formulação do Problema de Otimização Multiobjetivo

Para a formulação matemática do problema de *scheduling* da rede mostrada na Figura 2 foi utilizada a seguinte nomenclatura:

$B_{ij}$  : Quantidade de bateladas demandada do produto *j* pelo destino *i*.

$R_{ij}$  : Quantidade de bateladas recebida do produto *j* no destino *i*.

$P_{ij}$  : Quantidade mínima de bateladas do produto *j* que deve ser enviada pela fonte *i*.

$C_{ij}$  : Quantidade de bateladas do produto *j* armazenado no tanque *i*.

$E_{ij}$  : Quantidade de bateladas do produto *j* enviado pela fonte *i*.

$Frag_i$  : Quantidade de fragmentações na conexão *i*.

$Mask$  : Vetor contendo a quantidade de produto em cada conexão em um dado instante de tempo. Este vetor é resultante da aplicação

da máscara dada na Tabela 2 em um indivíduo codificado como na Figura 3.

$Tchegada_{ij}$ : Tempo de chegada da última batelada do produto  $j$  no destino  $i$ .

$Tmin_i$ : Menor tempo de recebimento de uma batelada no destino  $i$ .

$Tmax$ : Horizonte de tempo do modelo ou horizonte de programação.

$NC$ : Número de colisões na conexão bidirecional.

$LCm_{ij}$ : Limite inferior da quantidade de bateladas do produto  $j$  no nó  $i$ .

$LCM_{ij}$ : Limite superior da quantidade de bateladas do produto  $j$  no nó  $i$ .

$Nf$ : Quantidade de fontes.

$Nd$ : Quantidade de destinos.

$N_{conex}$ : Quantidade de conexões.

$Nt_i$ : Quantidade de tanques no nó  $i$ .

$Nn$ : Quantidade de nós.

$No$ : Quantidade de objetivos.

## 5.1 Restrições

O modelo matemático desenvolvido está sujeito a uma série de restrições operacionais descritas pelas Equações 1 a 4. A Equação 1 indica que uma quantidade mínima de cada produto deve ser enviada de cada fonte, para evitar a paralisação da produção em uma refinaria devido à indisponibilidade de recursos (tanques) para armazenagem dos produtos.

$$P_{ij} \leq E_{ij} \leq LCM_{ij} \text{ para } i = 1, \dots, Nf \text{ e } j = 1, \dots, Nt_i \quad (1)$$

A demanda em cada destino deve ser cumprida para evitar desabastecimento no mercado e cada destino não deve receber mais do que o planejado pois pode não haver tanques suficientes para armazenar os estoques.

$$R_{ij} = B_{ij} \text{ para } i = 1, \dots, Nd \text{ e } j = 1, \dots, Nt_i \quad (2)$$

Um nó não pode enviar e receber ao mesmo tempo produtos de uma conexão bidirecional, portanto ou o nó está em um estado de envio ou de recebimento ou ainda em estado ocioso.

$$NC = 0 \quad (3)$$

A capacidade de armazenamento em cada nó não pode ser violada.

$$LCm_{ij} \leq C_{ij} \leq LCM_{ij} \text{ para } i = 1, \dots, Nn \text{ e } j = 1, \dots, Nt_i \quad (4)$$

As restrições 1 e 2 serão tratadas como objetivos. Para tratar as restrições 3 e 4 adotou-se uma estratégia de reparação da solução que consiste em reparar um indivíduo inactível, transformando-o em uma solução factível.

A reparação da restrição 3 considera que a conexão bidirecional é desdobrada em duas conexões, uma representando o envio em um sentido, e a outra representando o envio no sentido oposto. Desta maneira, toda vez que duas bateladas existirem na conexão ao mesmo tempo, a batelada que iniciar primeiro será mantida, e a outra será retirada. Se as bateladas iniciaram ao mesmo tempo o envio, então uma delas é retirada aleatoriamente.

Para a reparação da restrição 4 e a fim de eliminar a sobredemanda, uma função recursiva é implementada. Esta função é responsável por fazer a contagem das bateladas e por atualizar o estado de cada tanque. Para cada conexão, e em cada instante de tempo, a função identifica se está acontecendo um evento de envio ou de recebimento de uma batelada. Se o evento for um envio, então a única violação passível de ocorrer seria a superação do limite inferior do tanque. Caso ocorra esta violação, a batelada é retirada da conexão e adicionada ao estoque na origem da batelada. Caso o evento seja um recebimento de uma batelada, então ocorreria uma violação no limite superior da tancagem agregada. Neste caso, a batelada é retirada da conexão, decrementada no destino e adicionada na origem. Ao adicionar uma batelada na origem, o estado passado do estoque se altera podendo ocorrer uma nova violação e por isto a função deve ser recursiva e recomençar a contar as bateladas novamente após terminar a verificação em todo o horizonte de tempo.

## 5.2 Função de *fitness*

O método aplicado para calcular o desempenho de um indivíduo é uma aproximação do método do critério global, onde faz se necessário o conhecimento do valor ótimo da função objetivo como forma de normalizar a rede (Coello & Christiansen, 2000). Critérios com grandezas diferentes podem dominar a avaliação final e com isto prejudicar os critérios com magnitudes menores. O método do critério global é apresentado na Equação 5:

$$F(x) = \sum_{i=0}^{N_0} \frac{f_i(x) - f_i^0}{fmax_i - f_i^0} \quad (5)$$

onde  $F(x)$  é o resultado da composição de todos os objetivos,  $fmax_i$  é o pior caso para o critério  $i$ ,  $f_i(x)$  é o resultado do critério  $i$  e  $f_i^0$  é o valor ideal para o objetivo  $i$ . Ao normalizar a rede, os resultados ficam dentro da faixa  $[0; 1]$ , de tal forma que 0 é o valor ótimo desejado, ou seja, todos os critérios foram cumpridos.  $F(x)$  com valor igual a 1 significa que nenhum critério foi atendido. Qualquer valor de  $F(x)$  entre 0 e 1 mostra que o critério foi atendido parcialmente.

Conforme discutido na Seção 4, os programadores estabelecem quatro objetivos a serem alcançados pela política de *scheduling* da rede de dutos. A formulação matemática destes objetivos é dada pelas Equações 6 a 9.

A fim de atender a demanda de fornecimento, a Equação 6 considera a minimização do envio de uma produção mínima por produto e por fonte, considerando todas as fontes.

$$\min \frac{1}{Nf} \sum_{j=0}^{Nf} \frac{1}{Nt_j} \sum_{i=0}^{Nt_j} \left( \begin{array}{l} 1 - \frac{E_{ij}}{P_{ij}}, \quad E_{ij} \leq P_{ij} \\ 0, \quad E_{ij} > P_{ij} \end{array} \right) \quad (6)$$

Para atender a demanda de recebimento, a Equação 7 apresenta a minimização do atendimento à demanda de cada produto em cada destino, considerando todos os destinos conjuntamente.

$$\min \frac{1}{Nd} \sum_{j=0}^{Nd} \frac{1}{Nt_j} \sum_{i=0}^{Nt_j} 1 - \frac{E_{ij}}{P_{ij}} \quad (7)$$

Para minimizar o tempo de atendimento à demanda de recebimento é necessário penalizar o resultado quando esta demanda não é atendida, considerando que a variável  $Tchegada_{ij}$  coincide com  $Tmin_j$ . A Equação 8 miniza o tempo de atendimento à demanda por produto em cada destino, considerando todos os destinos conjuntamente.

$$\min \frac{1}{Nd} \sum_{j=0}^{Nd} \frac{1}{Nt_j} \sum_{i=0}^{Nt_j} \frac{Tchegada_{ij} - Tmin_j}{Tmax - Tmin_j} + B_{ij} - R_{ij} \quad (8)$$

A Equação 9 minimiza a fragmentação no envio de bateladas em todas as conexões do modelo.

$$\min \frac{1}{N_{conex}} \sum_{i=0}^{N_{conex}} \left( \begin{array}{l} \frac{Frag_i - Mask[i] - 1}{Mask[i]}, \quad Frag_i \leq Mask[i] \\ 0, \quad Frag_i > Mask[i] \end{array} \right) \quad (9)$$

A avaliação final de um indivíduo (medida de adequabilidade ou *fitness*) é obtida pela média da soma de todos os objetivos, levando em consideração seus respectivos pesos, conforme mostrado na Equação 10.

$$J = \frac{\sum_{i=0}^{N_0} \omega_i * f_i}{\sum_{i=0}^{N_0} \omega_i} \text{ para } \omega_i \geq 0 \quad (10)$$

A metodologia desenvolvida é mostrada na Figura 4 e foi implementada em linguagem C++ utilizando a ferramenta Builder 5.0. Primeiramente uma população é inicializada com valores aleatórios. Em seguida, a função reparadora corrige as soluções infactíveis desta população de acordo com a violação de cada restrição. As violações referem-se aos limites máximo e

mínimo dos tanques, a sobreprodução e a existência de colisões. A população é em seguida avaliada e ranqueada de acordo com a função objetivo na Equação 10. Para isto são calculados os quatro objetivos referentes à produção mínima, ao atendimento à demanda (restrições modeladas como objetivos), ao tempo deste atendimento e por fim à fragmentação (objetivos propriamente ditos). Após o ranqueamento, as castas elitistas são formadas e alguns indivíduos são selecionados para a aplicação dos operadores genéticos de recombinação e mutação de acordo com o explicado na Seção 3. A nova população é então formada e fica sujeita novamente a função reparadora, pois soluções infactíveis podem surgir em qualquer ocasião após as operações genéticas.

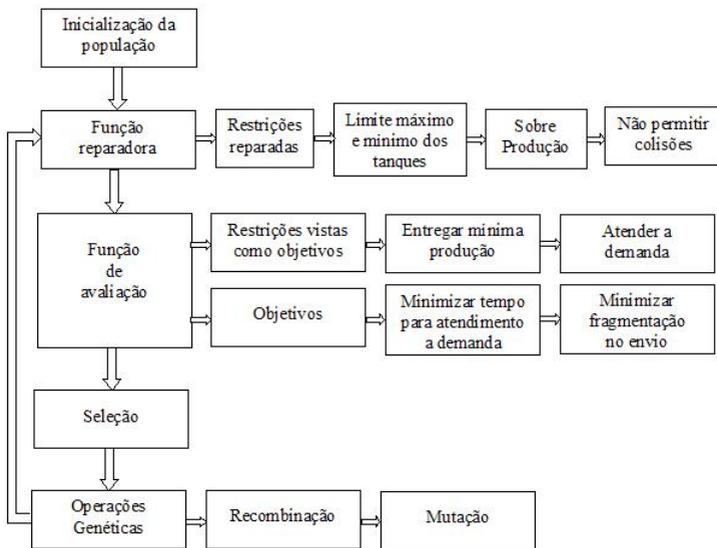


Figura 4. Metodologia baseada em AG para otimização multiobjetivo do *scheduling* da rede de dutos.

Como mostrado na Figura 4, o algoritmo micro-genético a menos do mecanismo de formação da população funciona exatamente como um algoritmo genético padrão em que os operadores genéticos são responsáveis por criar novos e melhores indivíduos a partir dos já existentes. Os operadores implementados neste trabalho são a recombinação de ponto único e a mutação, conforme descrito em Westphal (2006).

## 6. Resultados e Discussão

Para validar o modelo proposto, foi realizado um estudo de caso em que os objetivos das Equações 6 a 9 são ponderados na Equação 10 com os pesos dados na Tabela 3, os quais levam o denominador da função de *fitness* na Equação 10 a assumir o valor 10. Nesta tabela, considera-se que o objetivo relativo à minimização da fragmentação no envio das bateladas é um critério que pode prejudicar o desempenho da otimização dos outros critérios. Por esta razão o objetivo possui um peso associado menor. Isto significa que evitar a fragmentação é importante, mas não é a prioridade neste estudo de caso. Por outro lado, o atendimento às demandas de fornecimento e recebimento, bem como a minimização do tempo de transporte dos produtos na rede são igualmente priorizados. O horizonte de programação considerado em ambas as simulações é de 48 unidades de tempo, ou seja, o último recebimento de produto possível será na unidade de tempo 48 e nenhum envio poderá ocorrer após este tempo.

Tabela 3. Ponderações utilizadas na função objetivo.

<b>Critério</b>	<b>Objetivo</b>	<b>Peso</b>
Demanda de fornecimento	$(f_1)$	3
Demanda de recebimento	$(f_2)$	3
Minimização do tempo de programação	$(f_3)$	3
Minimização da Fragmentação	$(f_4)$	1

Para este estudo de caso, os parâmetros do algoritmo micro-genético foram escolhidos de acordo com as recomendações de Grefenstette (1986) e são dados na Tabela 4. Os limites máximo de armazenamento (volume dos tanques) para cada produto em cada nó são dados na Tabela 5.

Tabela 4. Parâmetros do AG.

<b>Parâmetro</b>	<b>Valor</b>
Tamanho da população	20
Taxa de mutação	0,1
Taxa de recombinação	0,8
Quantidade de indivíduos para o torneio	8
Quantidade de gerações para a reinicialização	1000

Para este estudo de caso, as demandas de fornecimento são configuradas através de uma quantidade mínima de bateladas de cada produto que deve ser enviada a partir de cada refinaria. Do mesmo modo, as demandas de recebimento são configurada através da quantidade de bateladas de

Tabela 5. Limite máximo de estoque de produto em cada nó.

<b>Nó</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
1	10	10	10	10
2	10	10	10	10
3	10	10	10	10
4	10	10	10	10
5	6	6	6	6
6	6	6	6	6
7	6	6	6	6

cada produto que cada terminal deverá receber. Estas demandas de fornecimento e recebimento por nó são apresentadas na Tabela 6. Os valores iniciais de estoque por produto e por nó são dados na Tabela 7.

Tabela 6. Demandas de fornecimento e recebimento.

<b>Demanda</b>	<b>Nó</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
Fornecimento	1	5	5	-	-
Fornecimento	2	-	-	5	5
Recebimento	3	3	3	3	3
Recebimento	4	3	3	3	3
Recebimento	5	3	3	3	3

Tabela 7. Valores iniciais de estoque de produto em cada nó.

<b>Nó</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
1	8	8	0	0
2	0	0	8	8
3	1	1	1	1
4	1	1	1	1
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0

Todas as simulações foram realizadas em um computador HP Intel Pentium Mobile 1600Mhz, 512 MB de memória RAM. Foram executados 20 experimentos de 5000 gerações cada. Em um experimento, a cada mil gerações, parte da população (subdivisões *Sb2* e *Sb3*) é reiniciada aleatoriamente nos espaços de busca descritos acima, excetuando a subdivisão *Sb1*. Esta subdivisão é reinicializada com soluções não-dominadas aleato-

riamente selecionadas da memória externa. Após formação da nova população, os indivíduos são ranqueados para a formação das castas conforme mostrado na Figura 1 e, a evolução continua. Os resultados estatísticos em termos de média, mediana, desvio padrão (DP), máximo e mínimo de cada um dos objetivos (Equações 6 a 9) e da função de avaliação (Equação 10) estão mostrados na Tabela 8. Estes resultados também são apresentados no *boxplot* da Figura 5. Em todas estas execuções foram encontradas 31 soluções de Pareto diferentes.

Tabela 8. Resultados para um conjunto de 20 simulações

Função	Mínimo	Máximo	Média	Mediana	DP
$f_1$	0	0,5114	0	0,0714	0,1584
$f_2$	0	0,4087	0,0093	0,0601	0,0936
$f_3$	0,2877	0,6208	0,3263	0,3866	0,1018
$f_4$	0	0,3017	0,1000	0,1200	0,0851
<b>J</b>	<b>0,0490</b>	<b>0,4416</b>	<b>0,0622</b>	<b>0,1184</b>	<b>0,1010</b>

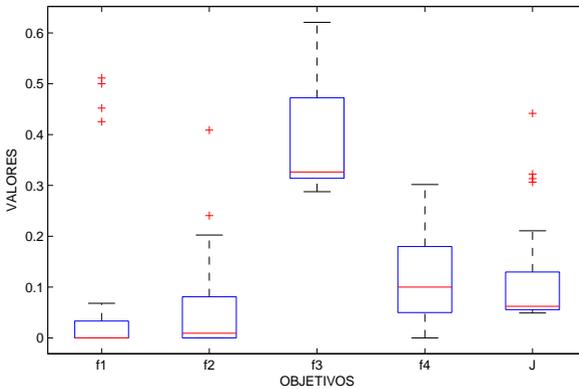


Figura 5. Resultados estatísticos de 20 simulações.

Por exemplo, para o experimento 1 (primeira simulação com 5.000 gerações), foram encontradas 3 diferentes soluções de Pareto que são dadas na Tabela 9. Onde o valor zero significa que o objetivo foi cumprido, o valor um significa que o objetivo não foi cumprido e qualquer valor entre zero e um significa que o objetivo foi atendido parcialmente. O tempo de execução para achar a solução foi de 30 segundos. O estado final de cada

tanque em cada nó é apresentado na Tabela 10. A carta de Gantt apresentada na Figura 6 mostra a distribuição de cada produto nas conexões do melhor indivíduo ao longo do horizonte de programação.

Tabela 9. Soluções de Pareto para o experimento 1.

Indivíduo (solução)	Demanda de Recebimento	Demanda de Fornecimento	Tempo de atendimento	Fragmen- tação
1	0	0	0,35	0,26
7	0,11	0	0,44	0,24
9	0,16	0	0,58	0,18

Tabela 10. Valores finais de estoque de produto em cada nó.

Nó	P1	P2	P3	P4
1	0	0	0	0
2	0	0	0	0
3	1	1	0	1
4	0	0	1	0
5	3	3	3	3
6	3	3	3	3
7	3	3	3	3

Pode-se notar que tanto a demanda de recebimento (objetivo  $f_1$ ) quanto a de fornecimento (objetivo  $f_2$ ) foram atendidas totalmente. No gráfico da Figura 6 pode ser visto que os envios se concentram do lado esquerdo do gráfico, o que significa que houve uma minimização do tempo de atendimento (objetivo  $f_3$ ) a demanda. Atingir o valor zero para o objetivo de minimização do tempo não é possível, pois fisicamente é inviável que todos os produtos de um terminal sejam recebidos ao mesmo tempo com apenas uma ou duas conexões. A Figura 6 mostra também uma minimização na fragmentação das bateladas (objetivo  $f_4$ ) nos envios, isto pode ser notado pela baixa alternância de produtos no gráfico.

Para fins de comparação, a Figura 7 traz a carta de Gantt da solução de Pareto representada pelo indivíduo 7 da Tabela 9. Analisando a conexão 1 desta figura, comprova-se que esta solução apresenta uma menor fragmentação (3 ocorrências) entre os produtos  $P1$  e  $P2$ , em comparação com a mesma conexão da solução ótima dada na Figura 6 (4 ocorrências).

Os resultados encontrados são satisfatórios em termos de solução e velocidade de convergência e validam o modelo e a metodologia propostos. Apesar da complexidade do modelo, a implementação pode ser considerada simples. O bom desempenho da rede está associado à capacidade de

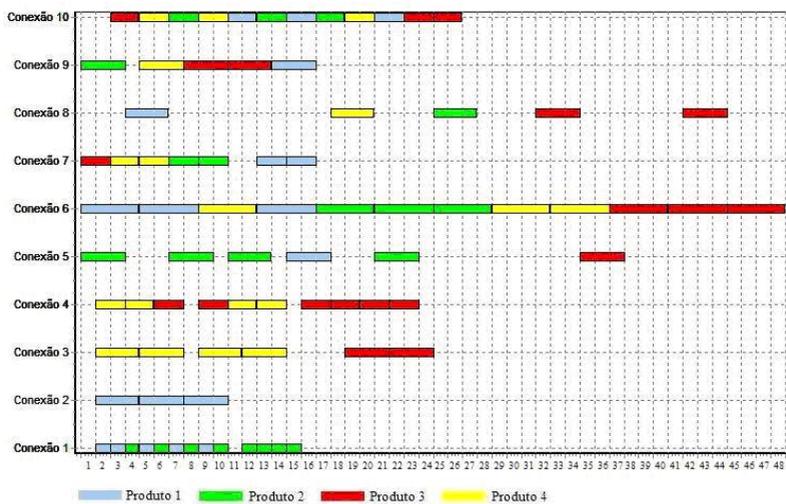


Figura 6. Carta de Gantt da melhor solução (indivíduo 1).

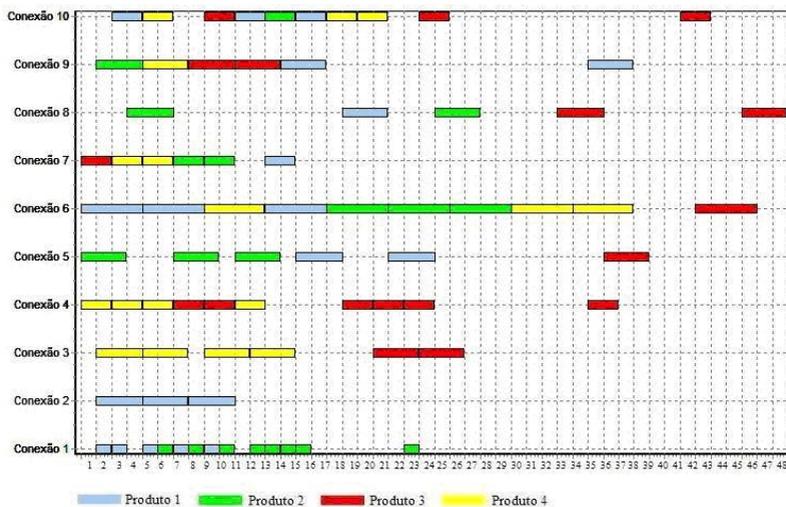


Figura 7. Carta de Gantt de uma solução não-dominada (indivíduo 7).

corrigir soluções infactíveis. As funções reparadoras devem ser tratadas com atenção para evitar inconsistências nas soluções durante as gerações. Os resultados também mostram que a conexão 6 é uma conexão crítica visto que devido a sua extensão, representada por uma elevada distância em unidades de tempo, era necessário que ela estivesse 100% do tempo ocupada para o atendimento da demanda dentro do horizonte de tempo programado.

Por fim, os resultados obtidos com o algoritmo micro-genético proposto foram comparados com os resultados obtidos com o algoritmo MOGA desenvolvido em [Arruda et al. \(2010\)](#). O micro-genético foi 10 vezes mais rápido que o MOGA em termos de carga computacional, e apresentou uma diversidade de soluções na fronteira de Pareto equivalente: 31 para micro-genético e 28 para o MOGA.

## 7. Conclusões

Este trabalho apresentou uma modelagem de um problema de distribuição de derivados de petróleo através de uma rede de dutos a qual é uma simplificação de uma rede real. O método de solução proposto é baseado em algoritmo micro-genético com elitismo, técnica de ranqueamento e formação de nicho, o qual garante, ao final da evolução, um conjunto de boas soluções em relação aos diferentes objetivos do problema de otimização. A utilização de uma pequena população com uma memória externa e uma casta interna garantiu a diversidade na população e permitiu alcançar um conjunto de soluções bem distribuído ao longo da fronteira de Pareto do problema multi-objetivo. O método do critério global foi utilizado para construção de uma função de *fitness* multiobjetivo e a utilização de ponderações permitiu a elaboração de um método sistemático para a inserção de prioridades no modelo. O tratamento das restrições via função reparadora foi uma contribuição essencial para evitar soluções infactíveis. Através do método recursivo desenvolvido, restrições deixaram de ser vistas como objetivo e puderam ser corrigidas dinamicamente sem depender da evolução do modelo.

As simulações realizadas no estudo de caso demonstraram o potencial do modelo desenvolvido para auxiliar o processo de tomada de decisão em problemas de *scheduling* de uma rede de distribuição de derivados de petróleo, sugerindo alternativas de soluções, de acordo com os objetivos considerados. Uma das principais vantagens do método proposto foi o baixo tempo computacional para alcançar a solução quando comparado com outra técnica do tipo MOEA.

Neste trabalho a modelagem apresentada não contemplou o uso inteligente das conexões bidirecionais e, com isto, não proporcionou um ganho real ao modelo, no que se refere à utilização deste tipo de conexão. O uso

racional destas conexões é importante, uma vez que elas são o gargalo de muitas operações em redes reais.

## Agradecimentos

Os autores agradecem à Agência Nacional do Petróleo – ANP – e à Financiadora de Estudos e Projetos – FINEP – por meio do Programa de Recursos Humanos da ANP para o Setor Petróleo e Gás – PRH-ANP/MCT (PRH10-UTFPR) e ao CNPq (Processos 305188/2007-0 e 311877/2009-5).

## Referências

- Alves, V. & Ferreira-Filho, V., Programação de transferência de derivados de petróleo em rede dutoviária usando algoritmo genético. In: *Anais do XXXIX Simpósio Brasileiro de Pesquisa Operacional*. Fortaleza, CE, p. 1–12, 2007.
- Arruda, L.; Neves-Jr, F. & Yamamoto, L., Using MOGA to order batches in a real world pipeline network. In: García-Pedrajas, N.; Herrera, F.; Fyfe, C.; Benítez, J.M. & Ali, M. (Eds.), *Trends in Applied Intelligent Systems – Part III*. Heidelberg, Alemanha: Springer-Verlag, v. 6098 de *Lecture Notes in Artificial Intelligence*, p. 546–555, 2010.
- Arruda, L.; Swiech, M.; Delgado, M. & Neves-Jr., F., PID control of MIMO process based on rank niching genetic algorithm. *Applied Intelligence*, 29(3):290–305, 2008.
- Carpinelli, G.; Protoa, D.; Noceb, C.; Russoc, A. & Varilone, P., Optimal allocation of capacitors in unbalanced multi-converter distribution systems: A comparison of some fast techniques based on genetic algorithms. *Electric Power Systems Research*, 80:642–659, 2010.
- Chen, D.; Hsu, Z. & Lin, F., Micro-genetic approach for surface meshing on a set of unorganized points. *Computer Methods in Applied Mechanics and Engineering*, 196:1997–2008, 2007.
- Coello, C., Evolutionary multiobjective optimization: Current and future challenges. In: Benítez, J.; Cordón, O.; Hoffmann, F. & Roy, R. (Eds.), *Advances in Soft Computing: Engineering, Design and Manufacturing*. Heidelberg, Alemanha: Springer-Verlag, p. 243–256, 2003.
- Coello, C. & Christiansen, A., Multiobjective optimization of trusses using genetic algorithms. *Computers & Structures*, 75(6):647–660, 2000.
- Coello, C. & Pulido, G., A micro-genetic algorithm for multiobjective optimization. In: Zitzler, E.; Deb, K.; Thiele, L.; Coello, C.C. & Corne, D. (Eds.), *Evolutionary Multi-criterion Optimization*. Berlin, Alemanha: Springer-Verlag, v. 1993 de *Lecture Notes in Computer Science*, p. 126–140, 2001.

- Crane, D.; Wainwright, R. & Schoenefeld, D., Scheduling of multi-product fungible liquid pipelines using genetic algorithms. In: *Proceedings of the ACM Symposium on Applied Computing*. San Antonio, EUA, p. 280–285, 1999.
- Cruz, J.; Andrés-Toro, B.; Hérran, A.; Besada, E. & Blanc, P., Multi-objective optimization of the transport in oil pipeline networks. In: *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*. p. 566–573, 2003.
- Deb, K., Evolutionary algorithms for multi-criterion optimization in engineering design. In: K. Miettinen M.M. Mäkelä, P.N. & Periaux, J. (Eds.), *Proceedings of Evolutionary Algorithms in Engineering and Computer Science*. Chichester, Reino Unido: John Wiley & Sons, p. 135–161, 1999.
- Dréo, J.; Siarry, P.; Petrowski, A. & Taillard, E., *Metaheuristics for Hard Optimization*. 2ª edição. Heidelberg, Alemanha: Springer-Verlag, 2006.
- Garcia, J.; Martin, J.; Gonzales, A. & Blanc, P., Hybrid heuristic and mathematical programming in oil pipelines networks. In: *Proceedings of the IEEE Congress on Evolutionary Computation*. v. 2, p. 1479–1486, 2004.
- Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, EUA: Addison-Wesley, 1989.
- Grefenstette, J., Optimization of control parameters for genetic algorithms. *IEEE transactions on Systems, Man and Cybernetics*, 16(1):122–128, 1986.
- Hertz, A. & Widmer, M., Guidelines for the use of meta-heuristics in combinatorial optimization. *European Journal of Operational Research*, 151(2):247–252, 2003.
- Krishnakumar, K., Micro-genetic algorithms for stationary and non-stationary function optimization. In: Rodriguez, G. (Ed.), *Proceedings of SPIE Intelligent Control and Adaptive Systems Conference*. p. 289–296, 1989.
- Neves-Jr, F.; Magatão, L.; Stebel, S.; Boschetto, S.; Felizari, L.; Czaikowski, D.; Rocha, R. & Ribas, P., An efficient approach to the operational scheduling of a real-world pipeline network. In: Plesu, V. & Agachi, P. (Eds.), *Proceedings of 17<sup>th</sup> European Symposium on Computer Aided Process Engineering*. Bucareste, Romênia: Elsevier B.V., p. 697–702, 2007.
- Relvas, S.; Matos, H.; Barbosa-Póvoa, A.; Fialho, J. & Pinheiro, A., Pipeline scheduling and inventory management of a multiproduct distribution oil system. *Industrial & Engineering Chemistry Research*, 45(23):7841–7855, 2006.

- Sasikumar, R.; Prakash, S.; Patil, R. & Ramani, S., PIPES: A heuristic search model for pipeline schedule generation. *Knowledge-Based Systems*, 10(3):566–573, 1997.
- Shah, N., Single and multisite planning and scheduling: Current status and future challenges. *American Institute of Chemical Engineering Symposium Series*, 320(94):75–90, 1998.
- Souza-Filho, E.; Alves, V. & Ferreira-Filho, V., Utilização de técnicas de pesquisa operacional em problemas de distribuição dutoviária: Uma revisão. In: *Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional*. Goiânia, GO, p. 1873–1880, 2006.
- Westphal, H., *Algoritmo Genético Aplicado à Otimização Multiobjetivo em Redes de Distribuição de Petróleo*. Dissertação de mestrado, Programa de Pós-graduação em Engenharia Elétrica e Informática, Universidade Tecnológica Federal do Paraná, Curitiba, PR, 2006.

---

## Notas Biográficas

**Henrique Westphal** é Mestre e Bacharel em Engenharia Elétrica pela Universidade Tecnológica Federal do Paraná. Atualmente trabalha como Engenheiro de Pesquisa e Desenvolvimento na Landys+Gyr.

**Flávio Neves Jr.** é Doutor em Automação pela Université Paul Sabatier - França (LAAS, 1998) e atualmente é Professor Associado III do Departamento de Eletrônica e do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI), ambos da Universidade Tecnológica Federal do Paraná.

**Lúcia Valéria Ramos de Arruda** é Doutor em Automação pela Université de Nice Sophia-Antipolis - França (UNSA, 1992) e atualmente é Professora Associada III do Departamento de Eletrônica e do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI), ambos da Universidade Tecnológica Federal do Paraná.

## Aplicação de Algoritmos Bio-Inspirados para a Síntese e Otimização de Sistemas para Produção de Petróleo *Offshore*

Beatriz Souza Leite Pires de Lima\* e Breno Pinheiro Jacob

---

**Resumo:** A indústria de petróleo vem enfrentando novos desafios na produção em águas profundas. No caso Brasileiro, as recentes descobertas de grandes reservas na camada do pré-sal têm impulsionado atividades de pesquisa, dentre elas o desenvolvimento de ferramentas computacionais de análise e projeto de sistemas para produção *offshore*. Neste cenário, este capítulo descreve resumidamente algumas aplicações de métodos de computação evolutiva na solução de diferentes problemas de síntese e otimização de sistemas de produção de petróleo *offshore*. As aplicações abordadas incluem o projeto de sistemas de *risers*, de operações de canhoneio de poços, e de definição de rotas de dutos submarinos.

**Palavras-chave:** Petróleo e gás, Sistemas *offshore*, *Risers*, Poços de petróleo, Dutos submarinos, Síntese e otimização, Computação evolutiva.

**Abstract:** *The oil & gas industry has been facing new challenges in deepwater production activities. The recent discoveries of huge oil reserves on the Brazilian pre-salt region have pushed research activities, including the development of computational tools for the analysis and design of offshore production systems. In this context, this chapter summarizes some applications of evolutionary computation on the solution of different synthesis and optimization problems, including the design of riser systems; perforating procedures of oil wells, and definition of submarine pipeline routes.*

**Keywords:** *Oil and gas, Offshore systems, Risers, Oil wells, Submarine pipelines, Synthesis and optimization, Evolutionary computation.*

---

\* Autor para contato: [bia@coc.ufrj.br](mailto:bia@coc.ufrj.br)

## 1. Introdução

A crescente demanda por óleo, gás e energia vem levando a indústria de petróleo a enfrentar novos desafios, relacionados principalmente a atividades de produção em águas profundas. No caso brasileiro, as recentes descobertas de grandes reservas na camada do pré-sal têm impulsionado atividades de pesquisa, incluindo o desenvolvimento de ferramentas computacionais de análise e projeto de sistemas para produção *offshore*. Dentre os diferentes tipos de sistemas que têm sido estudados, neste capítulo será abordada, resumidamente, a aplicação de métodos de computação evolutiva na solução de problemas de síntese e otimização relacionados ao projeto de *risers*, de operações de “canhoneio de poços” e de definição de “rotas de dutos submarinos”.

Para a produção de petróleo *offshore*, empregam-se plataformas flutuantes mantidas em posição por sistemas de ancoragem. As facilidades de produção na plataforma são conectadas a equipamentos submarinos (como as cabeças de poço ou *manifolds*) por conjuntos de dutos flexíveis conhecidos como *risers*, que levam o óleo e gás à plataforma, e também podem ser empregados para injetar água e gás nos poços. Uma configuração típica de *riser* conectado a uma plataforma flutuante é ilustrada na Figura 1.

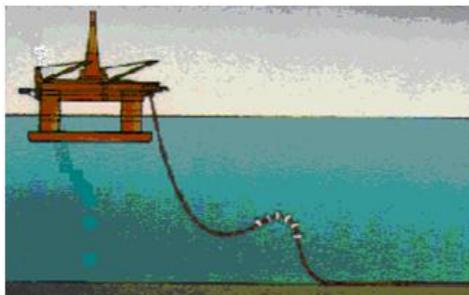


Figura 1. *Riser* conectado a plataforma flutuante.

Por sua vez, o projeto e execução dos poços inclui diversas atividades, incluindo perfuração, revestimento e cimentação. Estas últimas compõem a etapa de “completação”, que visa garantir produção satisfatória com o mínimo de intervenções. Nesta etapa, para manter a estabilidade geomecânica do poço e evitar o colapso, pode-se instalar um revestimento de aço. Como isto restringe o fluxo de óleo entre a rocha-reservatório e o poço, necessita-se executar as operações de “canhoneio” para restaurar o fluxo. Tais operações consistem em empregar cargas explosivas para abrir túneis que atravessam o revestimento, o cimento e perfuram a rocha-reservatório, como ilustrado na Figura 2.



Figura 2. Poço de petróleo canhoneado.

Finalmente, os sistemas *offshore* incluem, ainda, dutos submarinos empregados para exportar a produção de uma plataforma até uma base em terra, ou até outra plataforma instalada em águas mais rasas. A Figura 3 ilustra duas rotas alternativas para um duto apoiado no leito marinho, passando próximo a obstáculos e conectando equipamentos submarinos próximos a duas plataformas.

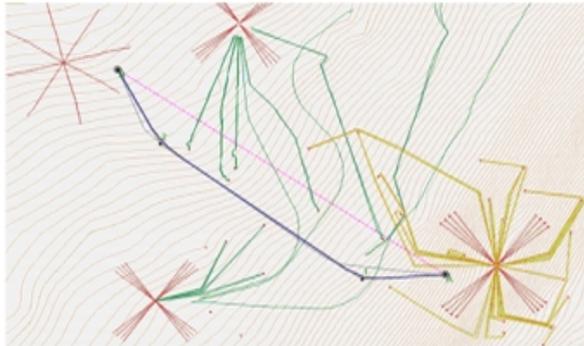


Figura 3. Rota de duto submarino.

Tradicionalmente, o projeto de tais sistemas vinha sendo feito manualmente por engenheiros, por inspeção e/ou efetuando uma série de simulações numéricas que empregam métodos custosos de análise (por exemplo, o Método dos Elementos Finitos – MEF), compondo estudos paramétricos exaustivos. Para ilustrar estes procedimentos tradicionais de projeto, pode-se considerar o caso dos três tipos de sistemas *offshore* mencionados anteriormente.

No caso do projeto de *risers*, para executar tais estudos usualmente empregavam-se ferramentas computacionais baseadas no MEF, variando-se

os parâmetros que definem sua configuração (por exemplo, o comprimento e posição dos segmentos com flutuadores). O resultado das simulações permitia indicar melhores valores para estes parâmetros, que levam a uma configuração eficiente, com menores custos e capaz de resistir aos carregamentos aplicados.

Por outro lado, no projeto das operações de canhoneio, diferentes fatores afetam a produtividade do poço, incluindo o número de tiros, a profundidade e o diâmetro do túnel. Neste caso, o objetivo também seria determinar os valores para estes parâmetros que levem à minimização dos custos e ao aumento do fluxo gerado pelo túnel do canhoneio, de modo a maximizar a produção.

Em relação ao projeto de dutos submarinos, a seleção de uma rota para o duto tem sido feita manualmente por engenheiros, pela inspeção dos dados geofísicos e geotécnicos (obtidos da batimetria e sonografia de fundo) e dos dados de obstáculos (equipamentos submarinos, regiões que devem ser evitadas, etc).

Assim, verifica-se que, mesmo com o auxílio de métodos de simulação numérica, estes procedimentos são altamente dependentes da experiência do engenheiro. Portanto, devem ser formalmente descritos e tratados como problemas de síntese e otimização. Sabe-se também que estes e outros problemas de engenharia são complexos por natureza e de difícil modelagem, o que dificulta a aplicação de métodos de otimização clássicos. Por outro lado, os métodos de computação evolutiva tem se mostrado promissores, como ilustrado em trabalhos anteriores com a participação dos autores (Vieira et al., 2003; Albrecht, 2005; de Lima et al., 2005; Vieira et al., 2008, 2010; Pina et al., 2010) onde tais métodos são aplicados a problemas *offshore* reais.

Assim, o restante deste capítulo descreve resumidamente desenvolvimentos de ferramentas de síntese e otimização baseadas em métodos de computação evolutiva, para os três tipos de sistemas *offshore* descritos: *risers*, canhoneio de poços e rota de dutos. Tais desenvolvimentos vêm sendo efetuados no contexto de trabalhos acadêmicos (dissertações de Mestrado e teses de Doutorado orientadas pelos autores), e em atividades de consultoria e desenvolvimento em parceria com a Petrobras. Os seguintes métodos de computação evolutiva são empregados na implementação das ferramentas aqui descritas: Algoritmos Genéticos (AG), Enxame de Partículas (PSO) e Sistemas Imunológicos Artificiais (SIA).

A descrição das ferramentas inclui a modelagem de cada problema, em termos da montagem da função-objetivo que avalia cada configuração candidata gerada no processo de busca. Esta função, que depende dos aspectos relevantes envolvidos no respectivo projeto, geralmente envolve um termo de custo (procurando alcançar os menores custos de construção/operação) e termos de penalização (procurando respeitar as restrições de projeto, de modo a obedecer todas as normas e critérios associados).

Em aplicações de algoritmos evolutivos a problemas de otimização de engenharia, o custo computacional para avaliação da função objetivo pode ser muito alto. Esta questão se mostra especialmente importante para as aplicações específicas em engenharia *offshore* apresentadas neste trabalho. Por exemplo, na otimização de sistemas de *risers* cada configuração candidata deve ser avaliada através de análises dinâmicas não-lineares de modelos de Elementos Finitos, que demandam tempo de processamento muito elevado. Portanto, o foco dos estudos apresentados também reside na redução do número de análises para o cálculo da função objetivo. Esta questão é tratada através de aprimoramentos nos algoritmos, customizando seus parâmetros de acordo com cada problema específico estudado, e na utilização de metamodelos.

## 2. Modelagem dos Problemas

### 2.1 *Risers*

#### 2.1.1 Descrição do problema

Os *risers* empregados em plataformas flutuantes podem ser classificados como flexíveis ou rígidos (Jacob et al., 1999). *Risers* flexíveis são estruturas complexas, com camadas de diferentes tipos de material. Tem sido empregados com sucesso em águas com profundidade menor do que 1000m. No entanto, em águas mais profundas podem alcançar seus limites de viabilidade técnica e econômica, devido às elevadas pressões externas a que são submetidos.

Mais recentemente, o conceito de *riser* rígido em catenária (SCR) tem se mostrado capaz de superar estas limitações, pois são de estruturas mais simples, fabricadas com tubos de aço. Estudos anteriores (Jacob et al., 1999) avaliaram configurações alternativas para SCRs, desde a catenária livre até outras mais complexas com elementos de flutuação. Os resultados demonstraram que estas últimas apresentam um comportamento mais favorável, e dentre elas a configuração *lazy-wave* (ilustrada na Figura 1) foi selecionada para estudos mais detalhados. A configuração *lazy-wave* é um arranjo em dupla catenária, com flutuadores distribuídos instalados em uma seção intermediária do *riser*. Estes flutuadores aliviam o peso suportado pela plataforma, e contribuem com forças de restauração quando o *riser* é submetido a cargas laterais.

Em Jacob et al. (1999) o comportamento desta e de outras configurações de *risers* foi estudado através de um grande número de análises não-lineares no domínio do tempo empregando modelos de Elementos Finitos. Tais análises compuseram um exaustivo estudo paramétrico, envolvendo a variação manual de diferentes parâmetros do *riser* e de dados de carregamento ambiental, para avaliar o comportamento estrutural e determinar valores melhores para alguns dos parâmetros que definem a configuração

do *riser* (incluindo, por exemplo, o comprimento e posicionamento do segmento com flutuadores). Naquele trabalho, o objetivo era alcançar uma configuração com custos razoáveis e capaz de resistir às cargas ambientais extremas e operacionais.

No entanto, aqueles estudos foram efetuados para um cenário particular, e as conclusões obtidas não podem ser diretamente estendidas para outros cenários como, por exemplo, em águas ultra-profundas, onde outro conjunto semelhante de exaustivos estudos paramétricos deveria ser realizado. Assim, deve-se reconhecer que a seleção de uma configuração de *riser* com bom desempenho estrutural e baixos custos deve, de fato, ser formalmente descrita e tratada como um problema de síntese e otimização.

Neste capítulo são sintetizados alguns dos resultados obtidos em trabalhos anteriores (Vieira et al., 2003; de Lima et al., 2005; Vieira et al., 2008), onde métodos de computação evolucionária foram empregados no desenvolvimento de uma ferramenta para a otimização de configurações de *risers*. Nas duas próximas seções será descrita a modelagem do problema e, mais adiante, na Seção 4.1, serão apresentados resultados de estudos de casos.

### 2.1.2 Variáveis de projeto

A Figura 4 apresenta um modelo esquemático indicando as variáveis de projeto consideradas no processo de otimização de um *riser* em configuração *lazy-wave*, a ser instalado em um cenário com profundidade da lâmina d'água indicada pelo parâmetro  $H$ .

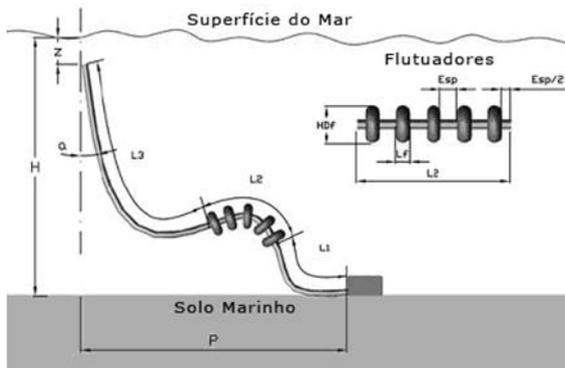


Figura 4. Variáveis de projeto para o *riser*.

Os parâmetros geométricos que definem o *riser* são os comprimentos dos segmentos de topo, com flutuadores, e inferior (respectivamente  $L1$ ,  $L2$  e  $L3$ ); o comprimento, diâmetro e espaçamento das bóias ( $Lf$ ,  $HDf$  e  $Esp$ ); o ângulo ( $\alpha$ ) que o eixo do *riser* faz com a direção vertical na conexão com

a plataforma, medido na configuração de equilíbrio neutro; a profundidade da conexão ( $z$ ), e a projeção horizontal ( $P$ ).

No entanto, estes dois últimos parâmetros são ditados pela característica da plataforma e do posicionamento dos poços. Portanto, não são considerados como variáveis de projeto do *riser* propriamente dito. Por outro lado, é possível demonstrar que o ângulo  $\alpha$  está geometricamente relacionado à projeção  $P$  e à soma dos comprimentos ( $L1 + L2 + L3$ ). Desta forma, as variáveis de projeto tratadas no problema de otimização são  $L1$ ,  $L2$ ,  $L3$ ,  $HDf$ ,  $Lf$  e  $Esp$ . Os valores para as demais variáveis são calculados a partir destes parâmetros de busca, ou dados como parâmetros fixos de entrada, como definido em [Vieira et al. \(2008\)](#).

### 2.1.3 Função objetivo e restrições

Uma função de avaliação do *riser* pode ser definida em função do comprimento e custo de cada segmento, bem como do custo das bóias, de acordo com a Equação 1:

$$f = \frac{f_{min}}{\left[ \sum_{i=1}^n IC_i \times L_i \right] + (V_{flut} \times IC_{flut})} \quad (1)$$

onde  $i = 1, \dots, n$  representa o índice de cada segmento;  $IC_i$  e  $L_i$  são, respectivamente, a ponderação de custo relativo e o comprimento de cada segmento;  $V_{flut}$  e  $IC_{flut}$  representam o volume das bóias e o respectivo fator de ponderação de custo; e  $f_{min}$  é o valor mínimo que pode ser atribuído a  $f$  de modo a normalizar o resultado no intervalo  $[0,1]$ .

As restrições de projeto, determinadas a partir de resultados de análises estruturais do *riser*, são as seguintes:

- a) Um valor máximo admissível para a tensão equivalente de Von Mises atuando nas seções do *riser* (para garantir sua integridade estrutural);
- b) Um valor máximo para o ângulo entre o eixo do *riser* e a direção vertical, na conexão de topo com a plataforma (para atender a requisitos da instalação);
- c) A variação máxima entre o ângulo de *built-in*, medido a partir do eixo do *riser* no topo, entre a configuração de equilíbrio neutro e qualquer outra configuração alcançada pelo *riser* durante a aplicação das cargas ambientais e do movimento da plataforma (para atender a limitantes do projeto da junta flexível que compõe a conexão com a plataforma);
- d) A tração máxima no topo (também para atender ao projeto da junta flexível);
- e) A tração mínima do *riser* no segmento de fundo (para evitar compressão, flambagem e colapso do *riser*).

Uma função de penalização é associada à violação de cada uma destas cinco restrições, pela seguinte expressão em termos da razão  $x$  entre o limite de projeto e o valor extremo calculado pelas análises, afetada por um fator  $k$  para favorecer soluções não restritas:

$$P = \begin{cases} k \times (1 - x^3), & \text{se } x < 1 \\ 0, & \text{se } x \geq 1 \end{cases} \quad (2)$$

Finalmente, a aptidão de uma configuração de *riser* é dada por:

$$f_{\text{aptidão}} = 100 \left[ \frac{f}{1.0 + \sum P_j} \right] \quad (3)$$

onde  $P_j$  é a penalização associada à violação do  $j$ -ésimo critério de restrição. Com esta expressão normalizada, a faixa de valores possíveis para a aptidão varia no intervalo  $[0,100]$ .

## 2.2 Canhoneio

### 2.2.1 Descrição do problema

A principal meta do processo de canhoneio de poços de petróleo é elevar a produtividade, procurando maximizar o desempenho da operação e minimizar os custos. Contudo, determinar a melhor estratégia para o canhoneio e a produção do poço é um processo complexo, visto que há um grande número de variáveis envolvidas, incluindo as propriedades geométricas do poço e do túnel do canhoneado (ilustrados na Figura 5), as propriedades físicas da rocha-reservatório, as condições operacionais e o cenário econômico.

Para avaliar a influência destas variáveis e determinar quais delas efetivamente influenciam no desempenho do poço de petróleo, em trabalhos anteriores foram efetuados estudos paramétricos baseados em modelos de Elementos Finitos (da Silva et al., 2008; Baioco et al., 2008). No entanto, em razão da grande quantidade de parâmetros, e da ampla faixa de variação de cada um deles, verifica-se que efetuar manualmente a geração de modelos e a execução de análises é um procedimento muito exaustivo.

Assim, o desenvolvimento de uma ferramenta computacional de síntese e otimização, empregando algoritmos evolutivos, contribui para a obtenção e o uso de informações sobre a influência de todos estes parâmetros, isoladamente ou combinados, objetivando a maximização da produtividade, reduzindo os riscos na tomada de decisão e, conseqüentemente, obtendo maiores ganhos para a indústria do petróleo.

O desenvolvimento de tal ferramenta atende a diferentes objetivos, incluindo os seguintes:

- a) Formar um “banco de conhecimentos” com resultados de análises paramétricas realizadas automaticamente, possibilitando uma visão mais abrangente das operações de canhoneio;

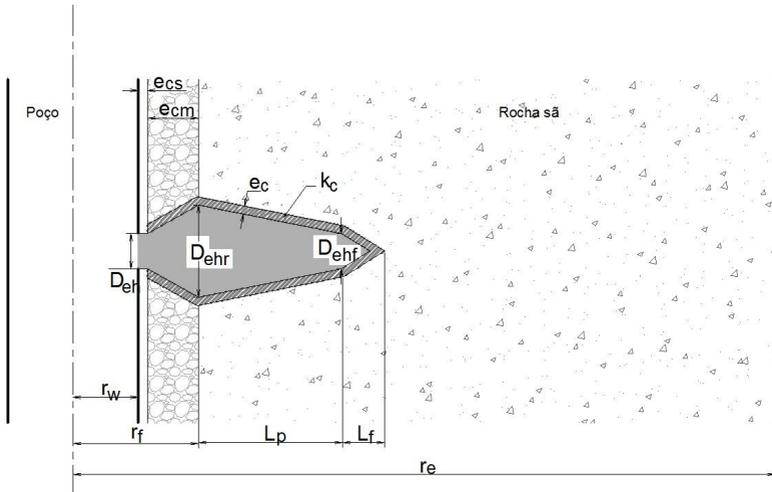


Figura 5. Representação esquemática do túnel canhoneado.

- b) Definir valores melhores para parâmetros que caracterizam a operação, de modo a obter procedimentos mais eficientes, que levem a maiores ganhos na produtividade;
- c) Quando estiverem disponíveis índices de custo atribuídos a diferentes componentes da operação de canhoneio, a definição da configuração ótima poderá atender também ao critério de custos menores.

Os resultados apresentados consistem de uma implementação preliminar da ferramenta de otimização, tendo-se obtido um primeiro protótipo, a partir do qual, após novos ciclos de testes, aplicações e desenvolvimentos (incluindo a disponibilização de mais informações sobre índices de custo), será possível obter uma versão de produção da ferramenta. Tal ferramenta poderá ser de grande valia para a gestão de operações de canhoneio, quando aplicada à tomada de decisão de projetos de completação.

Partindo-se de um módulo de otimização baseado em implementações de algoritmos evolutivos descritos na Seção 3, as tarefas para implementação desta ferramenta podem ser sintetizadas da seguinte forma:

- a) Dentre os parâmetros relacionados ao problema de fluxo em canhoneados estudados em (da Silva et al., 2008; Baioco et al., 2008), selecionar os mais adequados para compor as variáveis de projeto que devem ser otimizadas. Para isto torna-se necessário estudar a correlação e a interdependência entre os diferentes parâmetros.

- b) Montagem da função objetivo que avalia cada configuração, prevendo a definição de índices de custos relativos. O procedimento padrão para a análise do problema de fluxo emprega o programa de Elementos Finitos MSC/MARC. Alternativamente, foi desenvolvido um metamodelo para o cálculo da vazão no canhoneado baseado em uma solução analítica aproximada.
- c) Estabelecimento de restrições e penalidades para controlar o espaço de busca e evitar o surgimento de indivíduos ou configurações dos parâmetros do procedimento de canhoneio que não sejam factíveis ou realistas.

### 2.2.2 Variáveis de projeto

A tarefa (a), mencionada anteriormente, é crucial para se obter uma ferramenta factível, já que, caso fossem considerados todos os parâmetros da extensa quantidade envolvida em procedimentos de canhoneio, o espaço de busca do processo de otimização seria demasiadamente complexo. Assim, o passo inicial na montagem da ferramenta consistiu-se na execução de uma análise estatística de correlação entre as variáveis, utilizando como referencial um grande banco de dados obtido com os resultados dos estudos paramétricos relacionados ao problema de fluxo (da Silva et al., 2008; Baioco et al., 2008).

O resultado desta análise estatística permitiu estabelecer a correlação dos parâmetros entre si e com a produtividade, de modo a identificar quais influenciam mais significativamente no processo e, portanto, devem ser considerados como variáveis livres no processo de otimização. Na seleção das variáveis considera-se a distinção entre parâmetros geométricos do problema do canhoneio (dentre os quais se incluem aqueles que realmente podem ser controlados durante operações de canhoneio), e os parâmetros físicos que são características da rocha-reservatório e, portanto, não podem ser considerados como variáveis de projeto a otimizar.

Considerando-se estes critérios, e observando-se a representação esquemática de um túnel canhoneado indicado na Figura 5, as variáveis livres selecionadas para o procedimento de otimização são as seguintes:

- a) Densidade de tiros, expressa como o “número de tiros por pé” (*spf*, *Shots per Foot*), variando entre 2-5 *spf*;
- b) Comprimento do túnel ( $L_p$ ), variando entre 3-80 in;
- c) Diâmetro do túnel na rocha ( $D_{ehr}$ ), variando entre 2 e 1 in;
- d) Diâmetro final do túnel ( $D_{ehf}$ ), variando entre 0,1 e 0,75 in;
- e) Espessura da região da rocha danificada pelo canhoneio ( $e_c$ ), variando entre 0,1 e 1 in.

### 2.2.3 Função objetivo e restrições

Para a avaliação da função objetivo dos indivíduos, ou configurações de canhoneio candidatas no processo de otimização, deve-se determinar o comportamento do problema de fluxo de óleo da rocha-reservatório até o poço, passando pelos túneis do canhoneado.

O modelo matemático para representar o problema de fluxo é baseado na lei de Darcy para um sistema radial, dado pela Equação 4, onde  $k$  representa a permeabilidade da rocha;  $\mu$  é a viscosidade do fluido;  $r_e$  e  $r_w$  são os raios do reservatório e do poço, respectivamente;  $h$  é a altura do sistema de fluxo radial, e  $\Delta P$  é a variação da pressão:

$$q = \frac{2 \cdot \pi \cdot k \cdot h}{\mu \cdot \ln\left(\frac{r_e}{r_w}\right)} \cdot \Delta P \quad (4)$$

O método mais preciso para resolver este modelo matemático e calcular o fluxo no poço canhoneado consiste em gerar modelos numéricos tridimensionais e efetuar análises de Elementos Finitos (EF), empregando-se, por exemplo, um programa comercial como o MSC/MARC. No entanto, cada análise é computacionalmente muito dispendiosa. Além disto, sabe-se que em procedimentos típicos de otimização por algoritmos evolutivos podem ser geradas dezenas de populações, cada uma podendo incluir até centenas de indivíduos. Portanto, a avaliação de todos os indivíduos de todas as populações poderia requerer milhares de análises.

Desta forma, para o desenvolvimento, validação e testes da ferramenta é necessário contar com uma ferramenta de avaliação mais rápida e eficiente do que uma simulação por EF. Para tanto, foi desenvolvido um metamodelo baseado na solução de uma formulação analítica para o cálculo da vazão no canhoneado, a partir de uma analogia da Lei de Darcy para fluxo em meios porosos com transferência de calor e resistências térmicas. Esta formulação, descrita em detalhes em Baioco et al. (2010), foi validada a partir dos resultados obtidos pelo modelo numérico, e possui a vantagem de exigir muito menos esforço computacional, reduzindo o tempo de análise, podendo, portanto, ser utilizada na calibração do algoritmo evolutivo empregado.

Tomando o fluxo  $q$  obtido, o desempenho ou aptidão de cada indivíduo gerado pelo algoritmo de otimização pode então ser avaliado pela função apresentada a seguir, que define a rentabilidade teórica do processo (sem levar em conta o declínio da vazão com o tempo), calculada em unidades monetárias, em função da vazão  $q$  e do custo total:

$$\text{Aptidão} = \text{Max}(3,76 \cdot 10^{-2} \cdot q \cdot P_{oil} \cdot T_{prod} - \text{Cost}) \quad (5)$$

Esta função considera o preço do óleo  $P_{oil}$ , o tempo de produção  $T_{prod}$  e o custo da operação de canhoneio  $\text{Cost}$ . A constante que multiplica a equação serve para conversão de unidades. Alternativamente, caso estes

parâmetros de preço e custo não estejam disponíveis, pode-se simplesmente tomar o próprio valor do fluxo  $q$  como a aptidão, de modo a maximizar o fluxo de óleo.

Finalmente, as restrições na geração dos modelos devem ser definidas para controlar o espaço de busca e evitar o surgimento de indivíduos ou configurações dos parâmetros do canhoneio que não sejam factíveis ou realistas. As restrições dizem respeito à geometria do canhoneado e às propriedades de permeabilidade  $k$  da rocha sã e das regiões danificadas pela perfuração e pelo canhoneio, como descrito a seguir com base no esquema da Figura 5.

As restrições geométricas procuram evitar que sejam gerados canhoneados distorcidos que não ocorrem em aplicações reais. Assim, os diâmetros do túnel na entrada no revestimento do poço ( $D_{eh}$ ) e na ponta ( $D_{ehf}$ ) devem ser menores que o diâmetro na rocha ( $D_{ehr}$ ):  $D_{eh}, D_{ehf} < D_{ehr}$ . Para haver coerência no raio de drenagem do reservatório, o raio externo do reservatório ( $r_e$ ) deve ser maior que o raio do poço ( $r_w$ ) somado ao comprimento do túnel ( $L_p$ ) e da ponta ( $L_f$ ):  $r_e > r_w + L_p + L_f$ . Para evitar sobreposição dos túneis dos canhoneados, o produto da densidade de tiros ( $spf$ ) pelo diâmetro do túnel na rocha ( $D_{ehr}$ ) não pode exceder o perímetro do poço e a altura ( $h$ ):  $spf \cdot D_{ehr} < 2 \cdot \pi \cdot r_w$  e  $spf \cdot D_{ehr} < h$ . A última restrição geométrica estabelece que a espessura do dano do canhoneio ( $e_c$ ), deve ser menor que o diâmetro do túnel na rocha ( $D_{ehr}$ ):  $D_{ehr} > e_c$ .

Finalmente, considera-se uma restrição relacionada aos parâmetros de permeabilidade, que estabelece que a permeabilidade do túnel ( $k_t$ ) deve ser maior que a da rocha sã ( $k_r$ ), a qual, por sua vez, deve ser maior do que a das regiões danificadas pelo canhoneio e pela própria perfuração:  $k_t > k_r > k_c, k_f$ .

Estas restrições são verificadas para cada indivíduo gerado. Caso alguma delas seja violada, um novo valor para o parâmetro correspondente é gerado aleatoriamente, dentro da faixa admissível, mantendo-se o indivíduo assim modificado na população.

## 2.3 Rota de dutos

### 2.3.1 Descrição do problema

Como mencionado na Introdução, tradicionalmente a seleção de uma rota para dutos submarinos tem sido efetuada manualmente pelo engenheiro, por inspeção dos dados de batimetria e obstáculos no fundo do mar. Trata-se de um processo complexo e governado por diversas variáveis, seguindo recomendações de projeto estabelecidas por normas. Eventualmente, a avaliação de alguma rota poderia ser feita através de métodos de análise e simulação numérica, mas, de qualquer forma, o processo é altamente dependente da experiência do engenheiro.

Intuitivamente, a melhor rota seria aquela com o menor comprimento, levando a menores custos de material. No entanto, outros fatores que afetam o desempenho também devem ser considerados, incluindo, por exemplo, as informações geofísicas/geotécnicas/geográficas obtidas através de medições de sonografia e levantamento da batimetria de fundo. Tais dados definem os obstáculos e regiões no leito marinho que devem ser evitadas, levam a um valor mínimo de raio de curvatura para a rota, e também ao número de vãos livres ao longo da rota que devem ser minimizados ou mitigados.

Assim, torna-se aparente que o processo de seleção de uma rota de duto submarino com elevado desempenho e baixo custo também deve ser formalmente descrito e tratado como um processo de síntese e otimização. Neste contexto, trabalhos anteriores (Vieira et al., 2010; de Lima Jr. et al., 2011) descreveram os passos iniciais na direção do desenvolvimento e implementação de uma ferramenta para a síntese e otimização de rotas de dutos submarinos, baseada em métodos de computação evolucionária. Neste capítulo, as próximas seções resumem os aspectos relacionados à modelagem do problema, e a Seção 4.3 apresenta alguns resultados preliminares obtidos. Maiores detalhes podem ser encontrados em (de Lima Jr. et al., 2011).

### 2.3.2 Parametrização da rota

A representação paramétrica que define a configuração geométrica de uma rota, em termos de seus pontos extremos  $A$  e  $B$ , e de um conjunto de trechos retos e curvas é apresentada a seguir.

A Figura 6 ilustra um trecho em curva, definido como um arco circular de raio  $R$  e pontos inicial e final  $PC$  e  $PT$  respectivamente. Retas conectam o ponto final e inicial de duas curvas consecutivas; o prolongamento destas retas se intersecta em um ponto de inflexão ( $PI$ ). A posição do conjunto de  $PI$ s pode ser associada a “pontos base” distribuídos ao longo da linha reta  $AB$  conectando os extremos da rota. A posição de cada  $PI$  relativa ao respectivo ponto-base pode então ser definida em termos das coordenadas polares indicadas na Figura 7: as coordenadas radiais ( $\delta$ ) e angulares ( $\alpha$ ).

Podemos demonstrar (Vieira et al., 2010) que o conjunto de retas e curvas que compõem uma rota pode ser completamente descrito apenas por essas coordenadas  $\delta$  e  $\alpha$  de cada  $PI$ , e pelo raio  $R$  da curva associada. Com base nestes parâmetros principais, expressões para calcular os demais parâmetros secundários são demonstradas em Vieira et al. (2010), incluindo o ângulo central  $AC$  do setor circular; as coordenadas dos pontos  $PC$  e  $PT$ ; o comprimento  $T$  dos segmentos de reta tangentes conectando estes pontos ao ponto de inflexão  $PI$  e as coordenadas do centro  $C$  do círculo contendo o arco.

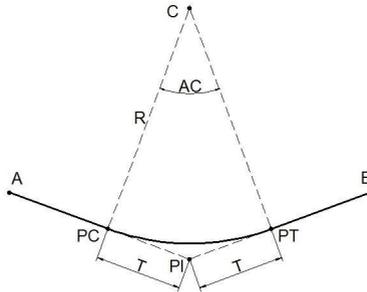


Figura 6. Representação geométrica de uma rota.

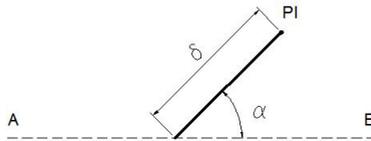


Figura 7. Coordenadas polares definindo um ponto de inflexão  $PI$ .

Observa-se que esta representação planar é complementada pela informação das coordenadas verticais obtidas de informações de batimetria que definem a topografia do fundo marinho.

### 2.3.3 Função objetivo e restrições

A função que avalia a aptidão de cada rota candidata deve levar em conta os aspectos relevantes considerados no projeto de rotas submarinas. Dentre eles, um dos mais importantes é o comprimento total, que deve ser minimizado para reduzir custos de material. Desta forma, se outros aspectos fossem ignorados, a melhor rota seria trivialmente definida como a linha reta ligando os pontos extremos  $A$  e  $B$ .

Evidentemente, existem diversos outros fatores que afetam o custo e a segurança de uma rota de duto, relacionados às restrições físicas, geométricas e estruturais. Para tratar estes fatores, introduzem-se  $Np$  termos de penalidade na função objetivo, de modo que a aptidão de cada rota é calculada pela seguinte expressão:

$$\text{Aptidão} = 100 \cdot \frac{dist_{AB}}{L_{Rota} + \sum_{i=1}^{N_p} (K_i \cdot penal_i)} \quad (6)$$

O termo  $dist_{AB}$  representa o comprimento da reta entre os extremos da rota. Se uma dada rota não viola nenhuma restrição, sua aptidão é definida simplesmente como  $dist_{AB}/L_{Rota}$ , e a solução trivial de uma rota não penalizada teria o valor máximo de aptidão igual a 100 já que  $dist_{AB} = L_{Rota}$ .

Nesta expressão, o valor de cada termo de penalidade é representado como  $penal_i$ , e  $K_i$  representa um fator de ponderação atribuído a cada penalidade para controlar sua influência relativa. As restrições associadas a estas penalidades são descritas em detalhes em [de Lima Jr. et al. \(2011\)](#), e algumas delas são apresentadas de forma resumida a seguir.

### Auto-Cruzamento

A metodologia de parametrização descrita na Seção 2.3.2 poderia gerar rotas com segmentos que se cruzam. Como tal configuração geométrica não é factível em aplicações reais, introduz-se o seguinte termo de penalização para evitar o auto-cruzamento de uma rota, com base em um algoritmo que percorre todos os segmentos e conta o número de cruzamentos  $nSelfCross$ :

$$penal_{SelfCross} = \exp(nSelfCross - 1) \quad (7)$$

### Interferência com Obstáculos

A região de solo marinho ao longo das rotas pode incluir áreas críticas tais como corais, ou outros obstáculos como equipamentos submarinos e outros dutos já existentes. Assim, para cada rota gerada verifica-se a ocorrência de interferência com tais obstáculos, e acumula-se o número total de ocorrências em uma variável  $nInter$ . Considera-se também os seguintes graus de importância de cada obstáculo, de acordo com as consequências de uma possível interferência:

- a) Poucas consequências, pode ser tolerada;
- b) Consequências médias;
- c) Severas, não se permite interferência.

A cada um destes níveis se atribui um fator de ponderação  $imp$ , permitindo que o algoritmo aceite rotas que cruzam obstáculos menos importantes, e o termo de penalização é definido pela seguinte expressão:

$$penal_{Obst} = \exp\left(\sum_{i=1}^{nInter} imp\right) - 1 \quad (8)$$

## Comprimento Mínimo dos Trechos Retos

As operações de lançamento de dutos requerem trechos retos entre duas curvas, e também nos segmentos inicial e final da rota, para garantir as manobras da embarcação de lançamento. Trechos retos já são naturalmente incluídos na parametrização descrita na Seção 2.3.2, mas é desejável garantir que um comprimento mínimo  $L_{min}$  seja respeitado. Para isto, introduz-se um termo de penalidade quando a rota apresenta um trecho reto com comprimento menor do que este valor mínimo.

## Raio Mínimo de Curvatura

Dutos lançados em curvas com raios pequenos podem escorregar sobre o solo e deixar a rota predefinida. Para evitar tal situação, estabelece-se o seguinte termo de penalidade sempre que o raio  $R_i$  de uma curva é menor do que um valor  $R_{Rotamin}$ , função do coeficiente de atrito solo-duto  $\mu$ , do peso submerso do duto  $w_s$ , e da tração residual no duto  $T_{residual}$  na sua configuração de equilíbrio após o lançamento:

$$R_{Rotamin} = \frac{T_{residual}}{\mu \cdot w_s} \quad (9)$$

$$penal_{Rmin} = \exp(R_{Rotamin} - R_i) - 1 \quad (10)$$

## Declividade

Um outro aspecto importante, relacionado à batimetria do solo marinho, é a declividade da superfície do fundo do mar onde o duto está apoiado. O algoritmo de geração de rotas calcula a declividade de cada ponto ao longo do duto tomando uma malha de pontos que representa o fundo, gerada através das curvas de nível. Sempre que a declividade ( $\theta_L$ ) assim calculada para uma série de pontos ao longo da rota ( $nNodes$ ) excede um valor preestabelecido ( $\theta_{LLim}$ ), introduz-se o seguinte termo de penalidade:

$$penal_{Decliv} = \frac{\sum_{i=1}^{nNodes} \exp(\theta_L - \theta_{LLim})}{nNodes} \quad (11)$$

## 3. Algoritmos

Nesta seção são apresentados os métodos de computação evolutiva que foram empregados na construção das ferramentas de síntese e otimização para as aplicações *offshore* descritas na seção anterior.

É importante ressaltar que, como o foco deste capítulo reside exatamente na modelagem e no estudo do desempenho dos métodos para estas

aplicações, apenas um breve resumo de cada método será apresentado, respectivamente para os Algoritmos Genéticos (AG), Enxame de Partículas (PSO) e Sistemas Imunológicos Artificiais (SIA). Para maiores detalhes, o leitor pode consultar as referências bibliográficas citadas ao longo do texto.

### 3.1 Algoritmos genéticos – AG

Dentre os métodos baseados em computação evolutiva mais populares, destacam-se os Algoritmos Genéticos (AG) (Goldberg, 1989). São inspirados na Teoria de Darwin sobre a evolução das espécies por seleção natural. Na aplicação destes métodos para a solução de um problema de otimização, os indivíduos (ou soluções candidatas) mais aptos dentro de uma população têm uma probabilidade mais alta de sobreviverem, e de gerarem descendentes mantendo suas boas características genéticas.

Cada problema é caracterizado por um conjunto de variáveis ( $x_n$ ) a otimizar. Um conjunto de valores  $x_1, x_2, \dots, x_n$  para estas variáveis define um dado indivíduo ou solução candidata. Nos AGs, cada variável é codificada em um “gene”, e o conjunto de genes para todas as variáveis, que identifica cada indivíduo, compõe um “cromossomo”. Uma população é então representada por um conjunto de indivíduos com seus respectivos cromossomos. Veja, por exemplo, a codificação de uma rota de duto apresentada na Seção 4.3.

Nos AGs simples, inicia-se o processo de evolução com uma população de indivíduos gerada aleatoriamente. A evolução das populações é realizada através de operadores genéticos que, em cada geração, efetuam a seleção probabilística dos melhores indivíduos para reprodução. Para esta seleção, avalia-se a qualidade ou “aptidão” de cada indivíduo, relacionada ao valor da função objetivo associada ao problema a ser otimizado (veja, por exemplo, as Seções 2.1.3, 2.2.3 e 2.3.3 da seção anterior).

Os indivíduos selecionados são colocados numa população intermediária para gerar os descendentes através das operações de cruzamento e mutação. Estas operações são fundamentais na evolução, pois permitem explorar diversas regiões do espaço de busca. O cruzamento é aplicado a um par de indivíduos criando dois descendentes, enquanto a mutação é aplicada somente num indivíduo após a operação de cruzamento. Este processo segue os mesmos preceitos da teoria da evolução, onde a probabilidade de cruzamento é alta e a de mutação é muito baixa.

Como resultado das operações de cruzamento e mutação, são criados descendentes que compõem as novas gerações. Cada geração tende a ser composta por melhores indivíduos, e o processo continua até satisfazer um determinado critério de parada, com o melhor indivíduo da última geração sendo a solução obtida.

Ao longo dos anos, diferentes variações e melhorias vem sendo incorporadas no AG canônico (Deb et al., 2002; Lemonge & Barbosa, 2004; Giger et al., 2007). Como já mencionado, não faz parte do escopo deste capítulo

descrever em detalhes o AG canônico e suas variações. Para isto, o leitor pode consultar Michalewicz (1999).

Quanto à escolha dos parâmetros de controle dos AGs, em princípio ela deveria ser feita para cada aplicação específica, o que é uma tarefa complexa. Porém, de modo geral podem ser empregados os seguintes valores, sugeridos na literatura (De Jong & Spears, 1990): taxa de cruzamento entre 0,60 e 0,90 e de mutação entre 0,001 e 0,05 por bit. O tamanho da população também pode variar de acordo com a aplicação. Algumas análises empíricas sugerem valores entre 50 e 100 indivíduos. Embora a maioria dos trabalhos ajuste os parâmetros do AG empiricamente, alternativas foram empregadas procurando ajustar estes parâmetros automaticamente (Eiben et al., 1999).

### 3.2 Enxame de partículas – PSO

O método de otimização por Enxame de Partículas – PSO (*Particle Swarm Optimization*), proposto por um psicólogo da área social e por um engenheiro eletricista (Kennedy & Eberhart, 1995), também é um algoritmo populacional que tem apresentado excelentes resultados na solução de problemas de otimização em engenharia. O método foi inspirado no comportamento social de bandos de pássaros e cardumes de peixes. Esta inspiração se deve à essência da inteligência social, onde os indivíduos aprendem com o sucesso dos vizinhos, se comparam a eles e imitam o comportamento dos que obtiveram sucesso.

No PSO, cada solução candidata é representada por uma partícula, que se move pelo espaço de busca em iterações sucessivas na busca pelo ponto ótimo. Este movimento é baseado numa teoria sócio-cognitiva onde cada partícula possui sua própria experiência e é capaz de avaliar a qualidade desta experiência. As partículas também possuem conhecimentos sobre o desempenho de suas vizinhas, caracterizando um comportamento social. Estes dois tipos de informação correspondem à aprendizagem individual (cognitiva) e à transmissão cultural (social), respectivamente.

Considerando um espaço de busca com dimensão  $L$  igual ao número de variáveis de projeto, cada partícula  $i$  é composta por vetores  $L$ -dimensionais de posição  $X_i$  e velocidade  $V_i$ , que são atualizados ao longo do tempo. O algoritmo armazena estes vetores em uma iteração  $t$  e os emprega para a atualização da população na iteração  $t + 1$ .

Em cada iteração, a posição atual de cada partícula é avaliada através da função objetivo associada ao problema, para determinar sua aptidão. A melhor posição encontrada pela partícula ao longo de toda a evolução até o momento atual é armazenada em um vetor  $p_i$ , e a melhor posição encontrada por todas as partículas do enxame também é armazenada em um vetor  $p_g$ .

Para atualizar o vetor de velocidade emprega-se a seguinte expressão:

$$V_i(t+1) = \omega \cdot V_i(t) + C_1 \cdot rnd_1 \cdot \frac{p_g - X_i(t)}{\Delta t} + C_2 \cdot rnd_2 \cdot \frac{p_i - X_i(t)}{\Delta t} \quad (12)$$

onde  $\Delta t$  é um intervalo de tempo, que pode ser considerado como tendo valor unitário;  $rnd_1$  e  $rnd_2$  são variáveis aleatórias de distribuição uniforme entre 0 e 1;  $\omega$  é um coeficiente de inércia que permite ajustar um balanço entre exploração e exploração (ou busca global e local) (Shi & Eberhart, 1998);  $C_1$  e  $C_2$  são os chamados parâmetros social e cognitivo, uma vez que ajustam o balanceamento entre os termos da equação que dependem respectivamente da distância entre a posição atual da partícula e a melhor posição das partículas vizinhas ( $p_g$ ) e da própria partícula ( $p_i$ ).

Para atualizar o vetor posição de cada partícula, utiliza-se a seguinte equação:

$$X_i(t+1) = X_i(t) + V_i(t+1) \cdot \Delta t \quad (13)$$

Em síntese, o algoritmo PSO pode ser descrito pelos seguintes passos:

1. Inicializar um conjunto de partículas em um tempo  $t=0$  com velocidades e posições aleatoriamente distribuídas dentro do espaço de busca;
2. Avaliar a função objetivo de cada uma das partículas da população;
3. Atualizar a melhor posição de cada partícula individualmente e a melhor posição do enxame;
4. Atualizar a posição de cada partícula no tempo  $t+1$  baseado na posição e velocidade no tempo  $t$ ;
5. Repetir os passos 2 a 4 até que uma condição de parada seja satisfeita.

### 3.3 Sistemas imunológicos artificiais

O sistema imune natural (SI) é o principal mecanismo de defesa do nosso organismo contra a invasão de microorganismos externos patogênicos. Sua função principal é distinguir e classificar, entre todas as moléculas presentes no organismo, aquelas que são próprias (do próprio organismo) e não-próprias (patogênicas). O SI é ativado quando reconhece uma molécula não-própria e produz uma defesa apropriada a esta ativação.

Os sistemas imunológicos artificiais (SIA) (de Castro & Timmis, 2002) são ferramentas numéricas inspiradas em teorias e princípios adaptativos dos sistemas imunológicos naturais. Diversas variantes dos algoritmos SIA têm sido aplicadas a problemas reais de otimização em engenharia. Dentre eles destaca-se o algoritmo CLONALG (de Castro & von Zuben, 2002), que é o utilizado nas aplicações descritas neste trabalho. Como o nome

sugere, ele foi inspirado pelo princípio da seleção clonal, em particular na geração e proliferação de clones e na sua modificação através da aplicação de hipermutação.

Este algoritmo, assim como o princípio que o inspirou, apresenta características evolutivas que se assemelham ao AG. No SIA, os anticorpos são as soluções candidatas e sua qualidade é tratada como “afinidade”, ou seja, o equivalente à aptidão no AG.

O algoritmo possui algumas características importantes, dentre as quais se destacam a capacidade de auto-adaptação da mutação, introdução de diversidade na população e preservação dos melhores anticorpos. A hipermutação somática é inversamente proporcional ao valor de afinidade, de forma que os melhores candidatos são pouco alterados devido às taxas baixas de mutação, enquanto os piores buscam melhorar sofrendo maiores efeitos da mutação. A taxa de mutação ( $\alpha$ ) pode ser apresentada da seguinte forma:

$$\alpha = e^{-\rho F^*} \quad (14)$$

onde  $\rho$  é um parâmetro que controla a suavidade da exponencial inversa, e  $F^*$  a afinidade normalizada entre 0 e 1. Esta taxa é empregada na mutação Gaussiana largamente utilizada para codificação usando números reais (Michalewicz, 1999).

Em síntese, o algoritmo original de de Castro & von Zuben (2002) pode ser descrito da seguinte forma:

1. Gerar uma população de anticorpos aleatoriamente;
2. Avaliar os valores de afinidade desta população;
3. Gerar  $N_c$  clones, clonando todos anticorpos da população;
4. Aplicar hipermutação nos clones proporcionalmente à afinidade;
5. Avaliar os valores de afinidade da população de clones mutados;
6. Selecionar os melhores clones para compor a nova população de anticorpos;
7. Repetir os passos 3 a 6 até atingir um critério de convergência.

#### 4. Estudos de Caso

Esta seção apresenta resultados de estudos de casos com a aplicação das ferramentas desenvolvidas para cada um dos problemas *offshore* considerados, de acordo com a modelagem descrita na Seção 2, e empregando os algoritmos descritos na Seção 3.

Procura-se abordar aspectos relacionados ao desempenho de cada ferramenta. Como já mencionado na Seção 1, no contexto de problemas reais de

engenharia tais como os problemas *offshore* o desempenho se mede em termos da redução do número de avaliações da função-objetivo (evidentemente associada ao aumento da aptidão média e à redução do desvio-padrão das aptidões em um conjunto de realizações aleatórias de cada experimento).

Assim, sabendo-se que o desempenho dos métodos de computação evolutiva em geral é significativamente dependente dos valores escolhidos para seus parâmetros, apresenta-se resultados de experimentos variando os parâmetros de modo a obter uma customização do algoritmo para a aplicação considerada. No caso dos *risers*, o foco está no ajuste dos parâmetros do método PSO. No caso das operações de canhoneio, na Seção 4.2.2 relata-se sucintamente resultados da comparação do desempenho dos três métodos (AG, PSO e AIS/CLONALG), cada um com parâmetros ajustados para esta aplicação em particular.

Ainda no caso das operações de canhoneio, e também procurando melhorar o desempenho do algoritmo, na Seção 4.2.1 apresentam-se resultados da aplicação de um metamodelo analítico simplificado que se compor em uma alternativa eficiente para a avaliação da função-objetivo.

Finalmente, na Seção 4.3 apresenta-se os primeiros resultados da aplicação da ferramenta de otimização da rota de dutos submarinos com AG.

#### 4.1 *Risers*

Estudos paramétricos relacionados à aplicação de diferentes algoritmos evolutivos para a otimização de *risers* foram apresentados em Pina et al. (2010) e Vieira et al. (2008). Nesta seção são resumidos alguns dos resultados obtidos, considerando uma configuração *lazy-wave* a ser instalada em uma lâmina d'água com profundidade de 1290m e uma projeção horizontal de 2000m.

Os limites atribuídos para as variáveis de projeto (Figura 4) são os seguintes: comprimento dos segmentos L1 e L3 variando entre [800m, 2000m], comprimento do segmento com flutuadores L2 [400m, 800m], diâmetro das bóias [0,8m, 2m], comprimento das bóias [0,5m, 2m] e espaçamento entre as bóias [1m, 3m]. Assume-se que o segmento com bóias custa duas vezes mais do que um segmento de *riser* sem flutuadores. Os limites para as restrições são os seguintes: tensão máxima de Von Mises 415,15 mPa, ângulo de topo máximo e mínimo 18° e 5° respectivamente, tração máxima no topo 1500 kN, e tração mínima 300 kN. Foi empregada na codificação das soluções candidatas a representação real.

A seguir são sintetizados alguns dos resultados de experimentos realizados com o PSO especificamente para aplicação de otimização de *risers*, procurando estudar o comportamento do algoritmo com a variação de seus parâmetros.

### 4.1.1 Descrição dos experimentos

Uma descrição detalhada dos parâmetros considerados pode ser encontrada em [Pina et al. \(2010\)](#). Em síntese, tomam-se os coeficientes de inércia da Equação 12, empregando tanto valores fixos  $\omega$ , quanto um valor inicial  $\omega_0$  associado a uma variação linear ou não linear, de acordo com [Eberhardt & Shi \(2000\)](#) e [Chatterjee & Siarry \(2006\)](#), respectivamente; e os parâmetros social e cognitivo  $C_1$  e  $C_2$ , empregando diferentes valores iniciais e finais que definem sua variação linear de acordo com a seguinte equação dada por [Ratnaweera et al. \(2004\)](#), onde  $N$  é o número máximo de iterações:

$$C_i(t) = (C_{i_{fin}} - C_{i_{ini}}) \frac{t}{N} + c_{i_{ini}} \quad (15)$$

Os testes consideraram uma população de 10 partículas, levando em conta resultados anteriores apresentados em [Albrecht \(2005\)](#) que indicaram que, com menos partículas, o espaço de busca não é suficientemente explorado e pode-se chegar a uma convergência prematura. Por outro lado, o uso de mais de 10 partículas aumenta o custo computacional sem melhorar significativamente os resultados.

O algoritmo é executado 30 vezes para cada conjunto de parâmetros, cada execução correspondendo a uma realização dos termos aleatórios. Cada execução é terminada quando a aptidão média permanece acima de 98% da aptidão da melhor partícula ao longo de cinco iterações sucessivas, o que indica que as partículas estão concentradas próximas a uma solução ótima.

### 4.1.2 Resultados

Resultados estatísticos detalhados destes experimentos, em termos de média e desvio-padrão dos valores de aptidão, são apresentados em [Pina et al. \(2010\)](#). Um bom desempenho do algoritmo é indicado por altos valores para a média e baixos valores para o desvio-padrão.

Sumarizando as conclusões obtidas, vale ressaltar a influência do coeficiente de inércia  $\omega$ . Observou-se que o uso de valores fixos para  $\omega$  levou a resultados piores (com menores médias e maiores desvios-padrão de aptidão), enquanto o melhor desempenho do algoritmo foi verificado com o uso da variação não-linear de  $\omega$ .

Pôde ser observado, ainda, que o desempenho é significativamente prejudicado pelo uso de valores menores para  $\omega$ . Este comportamento é mais notável para o valor fixo  $\omega = 0,4$  (ou para o valor inicial  $\omega_0 = 0,4$  nos casos com variação linear ou não-linear). Nestes casos, o algoritmo resulta numa convergência precoce para configurações de *risers* apresentando valores menores de aptidão média e valores maiores de desvio-padrão. Os resultados não são notavelmente diferentes para valores maiores do que  $\omega = 0.8$ , embora para  $\omega > 2,0$  tenham sido observados valores ligeira-

mente menores para a média e maiores para o desvio-padrão. Isto sugere, portanto, a seleção de  $\omega$  (ou  $\omega_0$ ) em uma faixa entre 1,2 e 2,0 .

Dentre todos os experimentos executados, o que levou à maior aptidão média (61,54) corresponde ao conjunto de parâmetros  $C_1/C_2$  com  $C_1$  variando linearmente entre 1 e 2, e  $C_2$  variando linearmente entre 2 e 1, com variação não-linear do coeficiente de inércia  $\omega$  e valor inicial  $\omega_0 = 1,6$ . Dentre as 30 rodadas deste experimento, o maior valor individual de aptidão obtido foi 61,69. A pequena diferença entre esta aptidão máxima e o valor médio (apenas 0,15, ou 0,24%) é refletida no fato de que este experimento também levou a um dos menores valores de desvio-padrão (0,108) dentre todos os testes. Como mencionado anteriormente, valores baixos de desvio-padrão indicam a qualidade e a robustez do algoritmo, o que, juntamente com a capacidade de fornecer maiores médias de aptidão, pode ser considerada uma das características mais importantes do método de otimização.

### 4.1.3 Desempenho computacional

Evidentemente, outra importante característica do método é seu desempenho computacional. Para as aplicações em problemas *offshore* descritas neste capítulo, os custos do algoritmo de otimização propriamente dito podem ser considerados irrelevantes. Nestas aplicações, os custos se concentram no método de avaliação de cada indivíduo. No caso dos *risers*, trata-se de análises dinâmicas não-lineares de modelos de Elementos Finitos. Portanto, o desempenho do método de otimização deve ser medido em termos do número de avaliações efetuadas.

Durante a execução dos estudos paramétricos, observou-se que o número de avaliações cresce com o aumento do valor do coeficiente de inércia  $\omega$ . Para o menor valor fixo utilizado, o algoritmo requereu cerca de 400 avaliações. Tomando valores fixos maiores o número de avaliações cresce notavelmente e, na maior parte dos casos, o algoritmo é interrompido ao se exceder o número limite de avaliações estabelecido em 1000.

Por outro lado, observou-se que o número de avaliações é reduzido com o uso da variação linear ou não-linear de  $\omega$ . Valores iniciais de  $\omega_0$  em uma faixa entre 0,4 e 1,2 levaram a um número de avaliações entre 200 e 500. Tomando valores iniciais maiores o número de avaliações cresce, mas nunca excede 900.

Em resumo, considerando-se apenas a eficiência e estabilidade do algoritmo (em termos de valores de média e desvio-padrão da aptidão), os resultados dos experimentos sugerem que um comportamento mais preciso e estável do PSO pode ser obtido empregando a variação não-linear do coeficiente de inércia  $\omega$ , associado ao valor inicial  $\omega_0$  em uma faixa entre 1,2 e 2,0 . Ao se levar em conta o custo computacional, lembrando que o número de avaliações cresce com  $\omega_0$ , conclui-se que o valor mais adequado para  $\omega_0$  é igual a 1,2 .

## 4.2 Canhoneio

### 4.2.1 Comparação dos métodos de avaliação

Inicialmente são apresentados resultados de experimentos comparando o desempenho do processo de otimização que utiliza avaliações pelo método dos Elementos Finitos e pelo metamodelo composto pela avaliação analítica aproximada, como descrito na Seção 2.2.3 (Baico et al., 2008, 2010). As variáveis de projeto a otimizar são as descritas na Seção 2.2.2. Foi empregada a representação real na codificação das soluções candidatas.

A Figura 8 e a Figura 9 apresentam os gráficos da evolução dos algoritmos, usando, respectivamente, a avaliação numérica por Elementos Finitos e o metamodelo analítico. Observa-se que esta última demandou um número superior de avaliações para obter a convergência e chegar ao resultado ótimo. Mesmo assim, como a avaliação analítica tem um custo computacional muito inferior à da avaliação numérica, o tempo total de execução do procedimento de otimização com a avaliação analítica foi de apenas 00:00:14, muito menor do que o requerido com a avaliação numérica, 15:12:25.

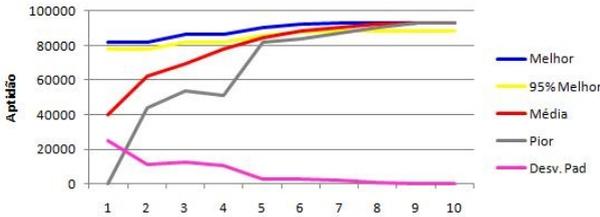


Figura 8. Evolução do algoritmo utilizando avaliação por elementos finitos.

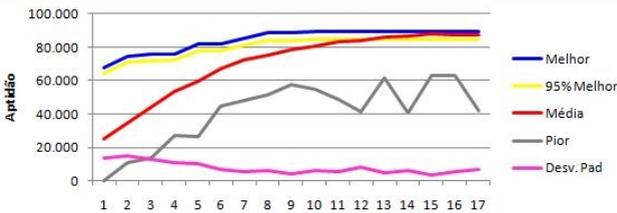


Figura 9. Evolução do algoritmo utilizando a avaliação analítica.

Na Tabela 1 estão apresentados os valores das variáveis que definem a configuração do melhor indivíduo, obtidos pela avaliação analítica aproximada e pelo *solver* de Elementos Finitos, bem como para a vazão obtida.

Como se pode observar, a avaliação analítica fornece uma aproximação razoável dos resultados mais rigorosos de EF, tanto para os parâmetros ótimos, quanto para a vazão.

Portanto, observa-se que, devido à sua elevada eficiência computacional, este metamodelo analítico pode se mostrar útil para auxiliar na depuração da ferramenta de otimização, bem como para a calibração dos parâmetros do método evolutivo. Apesar das simplificações envolvidas, o modelo fornece uma aproximação razoável, seguindo a tendência do resultado mais preciso fornecido pelo modelo numérico, e apresentando resultados mais conservadores.

Tabela 1. Comparativo dos valores ótimos.

	Elementos Finitos	Analítico	Diferença (%)
Número Tiros por Passo	5	5	0,0%
Comprimento do Túnel	76,97"	80,00"	3,9%
Diâmetro de Entrada no Revestimento	0,5"	0,5"	0,0%
Espessura do Dano do Canhoneio	0,10"	0,10"	0,0%
Vazão ( $in^3/d$ )	4114,66	3949,51	4,0%

#### 4.2.2 Comparação dos algoritmos

Estudos relacionados à aplicação dos diferentes algoritmos evolutivos foram apresentados em [Baio et al. \(2010\)](#). Diferentes parâmetros dos algoritmos foram variados, incluindo o tamanho da população, a taxa de cruzamento e de mutação do AG, o número de anticorpos, clones e o valor do parâmetro  $\rho$  do AIS/CLONALG, e o número de indivíduos e os coeficientes do PSO.

Para cada combinação de parâmetros, foram realizadas 50 execuções da ferramenta de otimização. Os resultados são apresentados em termos de valores estatísticos (média e desvio-padrão) da aptidão do melhor indivíduo, e do número de avaliações da função objetivo (que devem ser reduzidas considerando que o uso de métodos de avaliação por Elementos Finitos tem custos computacionais elevados).

Os resultados são apresentados e comentados de forma detalhada em [Baio et al. \(2010\)](#), onde foram realizadas várias análises de sensibilidade dos parâmetros dos três algoritmos utilizados. Foi observado que, conforme já citado anteriormente e reconhecido pela literatura, a escolha dos parâmetros dos algoritmos é um fator importante e específico para cada aplicação.

Uma síntese dos melhores resultados para cada algoritmo pode ser observada na Tabela 2. Comparando-se o desempenho dos três algoritmos, verifica-se que o PSO apresentou um resultado ligeiramente superior aos demais, tanto em termos de maiores médias e menores desvios-padrão (D.P.) da aptidão máxima, quanto do menor número de avaliações.

Tabela 2. Resumo dos resultados para o canhoneio.

	AG		AIS/CLONALG		PSO	
	Média	D.P.	Média	D.P.	Média	D.P.
Aptidão Máx. (US\$)	91712	370	91774	271	91895	4
Número de Avaliações	528	71	484	94	447	43

### 4.3 Rota de dutos

Nesta seção é ilustrada a aplicação da ferramenta de otimização de rota de dutos submarinos para um cenário típico da Baía de Campos, com resultados de um estudo para avaliar a influência da penalização de declividade descrita na Seção 2.3.3. Resultados detalhados deste e outros estudos são apresentados em [de Lima Jr. et al. \(2011\)](#).

Dentre os algoritmos descritos neste capítulo, na implementação inicial da ferramenta de otimização de rota considerou-se apenas o AG. A codificação de cada rota (ou indivíduo) no contexto do AG é então composta por um cromossomo com  $N$  genes, cada um associado a um  $PI$ , com três genes representando os três parâmetros principais descritos na Seção 2.3.2 (as coordenadas  $\delta$  e  $\alpha$ , e o raio  $R$ ). Além destes existe um quarto gene, associado a uma chave de ativação  $A$  que indica o estado de cada  $PI$ . Isto permite que o número de  $PIs$  varie ao longo do processo de otimização, a partir de um valor máximo especificado pelo usuário. Assim, dependendo da complexidade do problema, a representação geométrica da rota pode ser simplificada pelo algoritmo através da desativação de alguns  $PIs$ , o que indica que o trecho de rota correspondente é uma reta e não uma curva. A chave de ativação  $A$  é um valor binário 0 (indicando que o  $PI$  está inativo e seus parâmetros devem ser ignorados), ou 1 (indicando que o  $PI$  gera uma curva).

A codificação completa de um cromossomo pode então ser escrita como:

$$A_1 \delta_1 \alpha_1 R_1 A_2 \delta_2 \alpha_2 R_2 \dots A_N \delta_N \alpha_N R_N$$

onde  $A_1 \delta_1 \alpha_1 R_1$  são os genes correspondentes ao primeiro  $PI$ , e  $N$  é o número máximo de  $PIs$  especificado pelo usuário. Desta forma, ao longo do processo de otimização da rota, o algoritmo define soluções candidatas selecionando o número de pontos  $PI$ , os valores para as coordenadas que definem suas posições, e para os respectivos raios de curvatura.

Foram efetuadas rodadas do algoritmo tomando-se diferentes valores para a restrição de declividade longitudinal, respectivamente  $3^\circ$  e  $5^\circ$ . Os resultados obtidos (em termos de geometria/comprimento da rota, e o valor da aptidão) são comparados com uma rota projetada manualmente segundo os procedimentos usuais de projeto. A rota manual foi introduzida no

sistema para permitir a avaliação de sua aptidão e comparação com as rotas otimizadas obtidas.

A Tabela 3 resume os resultados obtidos, em termos do comprimento, aptidão e da soma dos valores de violações das restrições das rotas. Os resultados são visualizados na Figura 10, onde a declividade é indicada por escalas de cor. Pode-se observar que as rotas possuem comprimentos muito próximos, diferenciando-se basicamente pela declividade longitudinal máxima estipulada. A rota utilizando como critério uma declividade máxima de  $5^\circ$  apresentou o menor comprimento e, conseqüentemente, a melhor aptidão. Por outro lado, a rota com limite de  $3^\circ$  é mais longa, já que necessita contornar declives maiores. Por sua vez, a rota projetada manualmente tem um comprimento próximo ao da rota otimizada com  $3^\circ$ , mas infringe mais o critério de declividade, e portanto, na sua avaliação, mostrou aptidão menor para ambos os valores-limite de declividade.

Tabela 3. Resumo dos resultados para a otimização de rota.

<b>Rota</b>	<b>Comprimento</b>	<b>Melhor</b>	<b>Soma</b>
	<b>Rota</b>	<b>Aptidão</b>	<b>Penalidades</b>
Declividade Máxima $3^\circ$	13291m	18,0878	4,4631
Rota Original com $3^\circ$	13268 m	16,9680	4,8248
Rota Original com $5^\circ$	13268 m	19,3279	4,1106
Declividade Máxima $5^\circ$	13013 m	19,4031	4,1106

Observando-se estes resultados e os apresentados em [de Lima Jr. et al. \(2011\)](#), conclui-se que a ferramenta de otimização pode-se mostrar adequada para a aplicação a projetos reais. Os resultados são consistentes, satisfatórios e com baixo custo computacional. São coerentes com os observados com projetos já executados pela Petrobras. As pequenas diferenças ocorridas podem ser atribuídas aos limites estipulados para as restrições e penalizações, já que não se teve acesso à informação dos valores empregados no projeto original para as restrições de declividade e de raio de curvatura por exemplo.

## 5. Comentários Finais

Como mencionado na Introdução deste capítulo procurou-se agrupar resultados do desenvolvimento e aplicação de ferramentas de síntese e otimização, baseadas em métodos de computação evolutiva, para diferentes tipos de sistemas *offshore*. O principal objetivo foi apresentar tais resultados de forma consolidada e sintética, de modo a realçar a relevância da aplicação destas ferramentas em projetos reais de engenharia, e também a necessidade de ajustar os parâmetros dos algoritmos para customizá-los e melhorar seu desempenho para cada aplicação específica.

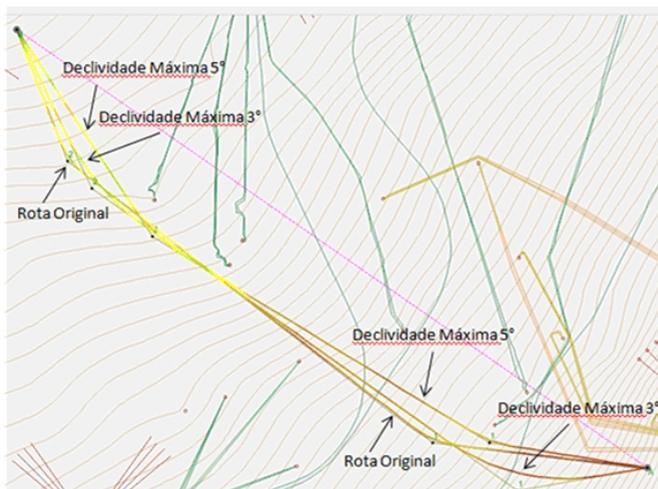


Figura 10. Comparação das rotas.

Evidentemente, não se pretendeu aqui esgotar todos os aspectos relacionados ao desenvolvimento e aplicação de métodos de computação evolutiva a problemas de engenharia em geral ou mesmo a problemas *offshore*, já que se trata de um tema muito vasto. Outros aspectos estão sendo tratados em estudos em andamento, tanto na modelagem dos problemas como em melhoramentos nos próprios algoritmos.

Por exemplo, está sendo considerada a incorporação de tratamentos mais sofisticados para as restrições, incluindo o uso de técnicas de penalização adaptativa (da Silva, 2010). Também está sendo estudado o uso de Redes Neurais Artificiais compondo metamodelos capazes de fornecer uma avaliação aproximada do sistema *offshore* e portanto reduzir os custos computacionais envolvidos nas avaliações da função objetivo (Pina et al., 2010). Com o mesmo intuito de reduzir o número de avaliações da função objetivo, pode-se considerar também a hibridização dos algoritmos, combinando as melhores características de cada um deles.

Além disto, todos os problemas descritos aqui foram modelados como uni-objetivo, mas muitos deles podem se beneficiar de uma representação multi-objetivo. É o caso por exemplo do problema de rotas de dutos submarinos, onde pode-se otimizar tanto o comprimento quanto o peso submerso necessário para manter sua estabilidade. Estudos nesta linha encontram-se em andamento (Vieira, 2010a), já tendo sido obtidos resultados preliminares no desenvolvimento de um método multi-objetivo baseado

no algoritmo SIA/CLONALG, denominado SIAMEP – Sistema Imunológico Artificial com Memória Externa de Pareto.

Como resultado destes estudos, espera-se obter ferramentas eficientes para uso prático em aplicações reais de projeto de sistemas *offshore* e, ao mesmo tempo, avaliar, comparar e aprimorar os algoritmos evolutivos de modo a torná-los mais eficientes para estas aplicações.

## Referências

- Albrecht, C., *Algoritmos Evolutivos Aplicados à Síntese e Otimização de Sistemas de Ancoragem*. Tese de doutorado, PEC/COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2005.
- Baioco, J.; Seckler, C.; Albrecht, C.; de Lima, B.; Jacob, B. & Soares, A., Application of optimization algorithms in perforating procedures of oil wells. In: *Proceedings of the 2<sup>nd</sup> International Conference on Engineering Optimization*. Lisboa, Portugal, p. 1–10, 2010.
- Baioco, J.; Seckler, C.; da Silva, K.; Jacob, B.; Silvestre, J.; Soares, A. & Freitas, S., Análise paramétrica aplicada a procedimentos de canhoneio de poços de petróleo. In: *Proceedings of of the Rio Oil & Gas Expo and Conference*. Rio de Janeiro, Brasil, p. 1–7, 2008.
- Chatterjee, A. & Siarry, P., Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & Operations Research*, 33(3):859–871, 2006.
- da Silva, A., *Análise de uma Técnica de Penalização Adaptativa aplicada ao Algoritmo de Enxame de Partículas*. Dissertação de mestrado, PEC/COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2010.
- da Silva, K.; Jacob, B.; Freitas, S.; Soares, A. & Silvestre, J., Predição da razão de produtividade de um poço canhoneado utilizando o método dos elementos finitos. In: *Anais do XXIX Congresso Ibero Latino-Americano sobre Métodos Computacionais em Engenharia*. Maceió, Brasil, p. 1–13, 2008.
- de Castro, L. & Timmis, J., An artificial immune network for multimodal function optimization. In: *Proceedings of the Congress of Evolutionary Computation*. p. 699–704, 2002.
- de Castro, L. & von Zuben, F., Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251, 2002.
- De Jong, K. & Spears, W., An analysis of the interacting roles of population size and crossover in genetic algorithms. In: Schwefel, H.P. & Männer, R. (Eds.), *Parallel Problem Solving from Nature*. Berlin, Germany, v. 496 de *LNCS*, p. 38–47, 1990.

- de Lima, B.; Jacob, B. & Ebecken, N., A hybrid fuzzy/genetic algorithm for the design of offshore oil production risers. *International Journal for Numerical Methods in Engineering*, 64(11):1459–1482, 2005.
- de Lima Jr., M.; Baioco, J.; Albrecht, C.; de Lima, B.; Jacob, B.; Rocha, D. & Cardoso, C., Synthesis and optimization of submarine pipeline routes considering on-bottom stability criteria. In: *Proceedings of the ASME 30<sup>th</sup> International Conference on Ocean, Offshore and Arctic Engineering*. Rotterdam, Netherlands, p. 1–12, 2011.
- Deb, K.; Pratap, A.; Agarwal, S. & Meyarivan, T., A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):181–197, 2002.
- Eberhardt, R. & Shi, Y., Comparing inertia weights and constriction factors in particle swarm optimization. In: *Proceedings of Congress on Evolutionary Computation*. Piscataway, USA: IEEE Press, v. 1, p. 84–88, 2000.
- Eiben, A.; Hinterding, R. & Michalewicz, Z., Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):181–197, 1999.
- Giger, M.; Keller, D. & Ermanni, P., AORCEA – an adaptive operator rate controlled evolutionary algorithm. *Computers and Structures*, 85(19/20):1547–1561, 2007.
- Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, USA: Addison-Wesley, 1989.
- Jacob, B.; de Lima, B.; Reyes, M.; Torres, A.; Mourelle, M. & Silva, R., Alternative configurations for steel catenary risers for turret-moored FPSOs. In: *Proceedings of the 9<sup>th</sup> International Offshore and Polar Engineering Conference*. Brest, France, p. 234–239, 1999.
- Kennedy, J. & Eberhart, R., Particle swarm optimization. In: *Proceedings of the IEEE Conference on Neural Networks*. Piscataway, USA: IEEE Press, p. 234–239, 1995.
- Lemonge, A. & Barbosa, H., An adaptive penalty scheme for genetic algorithms in structural optimization. *International Journal for Numerical Methods in Engineering*, 59(5):703–736, 2004.
- Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*. 3<sup>a</sup> edição. Heidelberg, Germany: Springer-Verlag, 1999.
- Pina, A.; Albrecht, C.; de Lima, B. & Jacob, B., Tailoring the particle swarm optimization algorithm for the design of offshore oil production risers. *Optimization and Engineering*, 12(5):215–235, 2010.
- Ratnaweera, A.; Halgamuge, S. & Watson, H., Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240–255, 2004.

- Shi, Y. & Eberhart, R., A modified particle swarm optimizer. In: *IEEE international Conference on Evolutionary Computation*. Piscataway, USA: IEEE Press, p. 1945–1950, 1998.
- Vieira, I., *Otimização Multi-Objetivo de Rotas de Dutos Submarinos utilizando Algoritmos Bio-Inspirados*. Exame de qualificação de doutorado, PEC/COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2010a.
- Vieira, I.; Albrecht, C.; de Lima, B.; Jacob, B.; Rocha, D. & Cardoso, C., Towards a computational tool for the synthesis and optimization of submarine pipeline routes. In: *Proceedings of the Twentieth International Offshore and Polar Engineering Conference*. Beijing, China, p. 226–233, 2010.
- Vieira, I.; de Lima, B. & Jacob, B., Optimization of steel catenary risers for offshore oil production using artificial immune system. In: Bentley, P.; Lee, D. & Jung, S. (Eds.), *Artificial Immune Systems*. Berlin, Germany: Springer-Verlag, v. 5132 de *LNCS*, p. 254–265, 2008.
- Vieira, L.; de Lima, B.; Evsukoff, A. & Jacob, B., Application of genetic algorithms to the synthesis of riser configurations. In: *Proceedings of the 22<sup>th</sup> International Conference on Offshore Mechanics and Arctic Engineering*. Cancun, Mexico, v. 1, p. 391–396, 2003.

## Notas Biográficas

**Beatriz Souza Leite Pires de Lima** possui graduação em Engenharia Civil pela Universidade do Estado do Rio de Janeiro (1983), mestrado (1991) e doutorado (1996) em Engenharia Civil pela COPPE/UFRJ. É professor associado do Programa de Engenharia Civil da COPPE e do departamento de Expressão Gráfica da Escola Politécnica da Universidade Federal do Rio de Janeiro. Tem experiência na área de sistemas *offshore*, métodos numéricos, inteligência computacional, *data mining*, otimização e algoritmos evolutivos.

**Breno Pinheiro Jacob** possui graduação em Engenharia Civil pela Universidade de Brasília (1979), Mestrado (1983) e Doutorado (1990) em Engenharia Civil pela COPPE/UFRJ. Atualmente é Professor Associado da Universidade Federal do Rio de Janeiro (lotado no Programa de Engenharia Civil da COPPE), e coordenador do LAMCSO – Laboratório de Métodos Computacionais e Sistemas *Offshore*. Atua em ensino, pesquisa, desenvolvimento e consultoria para a indústria, principalmente nos seguintes temas: Petróleo e Gás, Métodos Computacionais, Estruturas e Sistemas *Offshore*, Mecânica das Estruturas, Análise Dinâmica, Análise Não-Linear, Método dos Elementos Finitos.