

## **Proposta de um algoritmo de evolução diferencial adaptativo para resolução de problemas com muitos objetivos**

### **Proposal of an adaptive differential evolution algorithm to solve problems with many objectives**

DOI:10.34117/bjdv8n7-188

Recebimento dos originais: 23/05/2022

Aceitação para publicação: 30/06/2022

**Felipe Augusto Oliveira Mota**

Mestre em Engenharia Elétrica

Instituição: Instituto Federal do Norte de Minas Gerais - Campus Januária

Endereço: Fazenda São Geraldo, S/N, Januária - MG

E-mail: felipeaomota@gmail.com

**Suzana Viana Mota**

Mestra em Engenharia Elétrica

Instituição: Universidade Estadual de Campinas

Endereço: Cidade Universitária – Campinas - SP

E-mail: suzana.svm@gmail.com

#### **RESUMO**

A atual exigência dos processos computacionais tem potencializado o uso de algoritmos otimizados, principalmente os estocásticos. Os algoritmos evolucionários, representados pelos genéticos, despontam como a melhor solução para a programação otimizada. Neste artigo é proposto o desenvolvimento de um algoritmo baseado no Differential Evolution, chamado de Algoritmo de Evolução Diferencial Adaptativo, o mesmo tem o intuito de resolver problemas de muitos objetivos. Os resultados não foram satisfatórios e mudanças na implementação são recomendadas.

**Palavras-chave:** adaptativo, evolução diferencial, muitos objetivos, NSGAIII.

#### **ABSTRACT**

The current demand for computational processes has increased the use of optimized algorithms, especially stochastic ones. The evolutionary algorithms, represented by genetic algorithms, emerge as the best solution for optimized programming. In this paper it is proposed the development of an algorithm based on Differential Evolution, called Adaptive Differential Evolution Algorithm, which aims to solve problems with many objectives. The results were not satisfactory and changes in the implementation are recommended.

**Keywords:** adaptive, differential evolution, many objectives, NSGAIII.

## **1 INTRODUÇÃO**

ATUALMENTE os processos computacionais têm exigido resultados cada vez

melhores. Visando soluções com qualidade tem-se investido nos processos de otimização, que podem ser classificados como estocástico (abordagem que permite modelar problemas de otimização envolvendo incerteza) ou determinísticos (problemas formulados com parâmetros conhecidos). Sendo os estocásticos preferidos para resolver problemas mais complexos.

Uma classe de métodos de otimização estocásticos é o algoritmo evolutivo (AE), que simula o processo da evolução natural. Sua origem ocorreu no final da década de 1960 e desde os anos 70 muitas metodologias evolutivas vêm sendo propostas, principalmente por algoritmos genéticos, programação genética e estratégias evolutivas [1].

Por sua vez, os algoritmos genéticos (AG) são uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação.

Um AG inicia com uma população de indivíduos (cromossomos) que representam possíveis soluções do problema a ser resolvido. A população é avaliada e cada cromossomo recebe uma nota, denominada aptidão. Os melhores cromossomos (mais aptos) são selecionados e os piores descartados. Os selecionados podem sofrer cruzamento e mutação, gerando descendentes para a próxima geração[2]. A figura 1 esquematiza esses princípios.

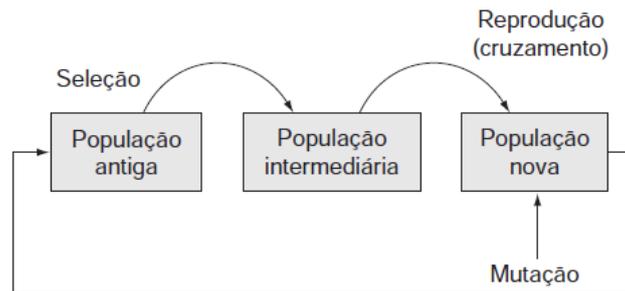
Os problemas de otimização podem variar de mono objetivo (1 objetivo), multiobjectivos (2 ou 3 objetivos) e muitos objetivos (mais de 3 objetivos), os mesmos podendo apresentar ou não restrições.

A maior motivação para usar algoritmos evolutivos, principalmente os genéticos, para resolver problemas de otimização de muitos objetivos é devido ao fato dos AEs lidarem simultaneamente com um conjunto de possíveis soluções (população) que permite encontrar vários membros do conjunto Pareto ótimo (solução) em apenas uma execução do algoritmo ao invés de ter que realizar uma série de execuções separadas, como ocorre em outros tipos de programação.

A caracterização do problema proposto se define em desenvolver um algoritmo para resolver um problema restrito de otimização com muitos objetivos.

O objetivo deste artigo é apresentar os resultados do Algoritmo de Evolução Diferencial Adaptativo.

Fig. 1. Trabalho de um algoritmo genético



## 2 METODOLOGIA

O trabalho foi desenvolvido utilizando como base o algoritmo de Evolução Diferencial, *Differential Evolution* (DE). Ao mesmo tempo foram aplicados os conceitos da distância de multidão do NSGAII e conceito adaptativo do ANSGA, além de utilizar o cálculo da Distância Euclidiana durante a dominância, outro conceito importado da “família” NSGA.

O Algoritmo proposto será executado utilizando como parâmetros de otimização as Funções *Benchmark* DTLZ1 e DTLZ2 e os seus resultados comparados com os do NSGA-III. Considerado atualmente como um dos melhores algoritmos para resolver problemas de muitos objetivos[3].

### 2.1 DIFFERENTIAL EVOLUTION (DE)

Algoritmo proposto por Storn e Prince é uma técnica que tem um bom desempenho em uma grande variedade de problemas. O DE é uma técnica de busca estocástica de base populacional e tem sido aplicado com sucesso em diversos campos da engenharia[4].

O DE utiliza a mutação como o mecanismo principal de busca, e a seleção para indicar a direção dentro da região viável do espaço de decisão. É baseado em populações e utiliza  $N_p$  indivíduos e  $D$  vetores de variáveis para cada geração.

Os melhores ajustes para os parâmetros de controle podem ser diferentes para cada tipo de problema de otimização. Portanto, para resolver com sucesso um problema de otimização específica, é geralmente necessário realizar uma busca de tentativa-e-erro para a combinação mais apropriada de estratégias e os seus valores de parâmetros associados.

A ideia básica do DE é um esquema para a geração do chamado vetor experimental. O DE gera este novo vetor a partir de três membros da população. Se o vetor experimental possui um valor de função objetivo menor a um vetor predeterminado

da população, este substituirá o vetor ao qual está sendo comparado. Além disso, a cada geração o vetor com a melhor função objetivo é sempre armazenado para manter o registro do progresso ao longo do processo de otimização.

A população pode evoluir através de diferentes regiões do espaço de busca, daí a necessidade de se encontrar a melhor estratégia para gerar a solução ótima em cada diferente problema proposto.

As versões multiobjetivo do DE mais populares são as que utilizam o conceito de dominância para geração de novas populações, empregando um operador de diversidade. Alguns autores têm sugerido a utilização dos mecanismos do NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) com o DE [5][6][7]. Nesses trabalhos se nota que os algoritmos mais eficientes utilizam a classificação por não dominância e o operador de diversidade de distância de multidão, *crowding distance*, como no NSGA-II.

## 2.2 DOMINÂNCIA

A dominância [8] vislumbra proporcionar uma comparação mais justa entre os vários objetivos. O conceito de dominância é uma característica dos algoritmos multiobjetivos. Essa classe de algoritmos que leva em consideração a otimização de mais de um objetivo (vários objetivos são minimizados ou maximizados dependendo do tipo de problema), sendo estes normalmente conflitantes.

O critério de dominância segue a seguinte regra que dado dois indivíduos  $p$  e  $q$  pertencentes a uma mesma população  $P$ , então:

Um indivíduo  $p$  domina um indivíduo  $q$ , se no mínimo o valor em um dos objetivos de  $p$  é melhor que o mesmo objetivo em  $q$  e o restante dos valores dos objetivos de  $p$  não podem ser piores que o restante dos mesmos valores nos objetivos em  $q$ . Ou seja, isso significa dizer que  $p$  não pode possuir nenhum objetivo com menos qualidade do que  $q$ .

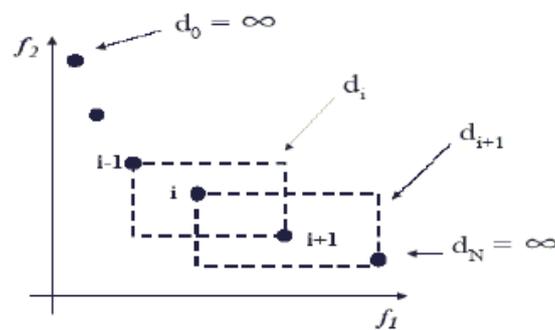
Ao final de cada análise um determinado grupo de indivíduos são classificados como pertencentes a uma categoria específica denominada fronteira e ao ser concluído o processo classificatório todos os indivíduos estarão inseridos em uma das  $n$  fronteiras.

A fronteira 1 é constituída de todas as soluções não-dominadas. A fronteira 2 pode ser conseguida considerando todas as soluções não-dominadas excluídas das soluções da fronteira 1. Para determinação da fronteira 3, exclui-se as soluções previamente classificadas nas fronteiras 1 e 2, e assim por diante até que todos os indivíduos tenham sido classificados em fronteiras.

### 2.3 DISTÂNCIA DE MULTIDÃO

O algoritmo de distância de multidão, *crowding distance*, calcula a distância média entre um ponto central  $i$  selecionado dentro da população e dois pontos localizados nas extremidades do ponto central ( $i-1$ ) e ( $i+1$ ). A ideia é que a partir de um ponto central o operador de diversidade possa encontrar pontos extremos e priorizar os pontos mais distantes durante o processo de seleção a fim de espalhar os resultados ao longo da fronteira pareto. A disposição dos pontos extremos formam um cubóide em relação ao ponto central. A figura 2 exemplifica a distância de multidão.

Fig. 2. Distância de Multidão



A Equação 1 é usada para calcular a distância de cada um dos indivíduos em relação ao ponto médio (levando-se em consideração cada um dos  $m$  objetivos).

A distância atual  $d_j^m$ , que significa a distância do  $j$ -ésimo elemento do conjunto  $I$  em relação ao objetivo  $m$ . A distância será somada com a razão da diferença dos valores consecutivos dos objetivos, correspondentes aos elementos das posições  $j+1$  e  $j-1$  do conjunto  $I$ , que são valores oriundos do  $m$ -ésimo objetivo, e representados por  $f_m^{(I_{j+1}^m)}$  e  $f_m^{(I_{j-1}^m)}$ . E esse valor dividido pela diferença entre seus objetivos máximo e mínimo também pertencentes ao  $m$ -ésimo objetivo, ou seja,  $f_m^{max} - f_m^{min}$ . A fórmula descrita está abaixo representada.

$$d_j^m = d_j^m + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{max} - f_m^{min}} \quad (1)$$

Esse procedimento será repetido dependendo do número de objetivos existentes no problema ( $m$  vezes), isso com a finalidade de contemplar todos os aspectos

considerados importantes durante a busca das soluções.

## 2.4 ADAPTATIVO

O termo adaptativo é definido na biologia como uma conformação vantajosa de um organismo a mudanças no seu ambiente. Inspirado por esta definição, o termo adaptativo foi introduzido na teoria dos algoritmos genéticos.

A característica adaptativa do ANSGA (Algoritmo Genético Adaptativo com Elitismo), proposto por Barbosa, Ribeiro e Arantes[9], foi inserida no cruzamento e mutação. Estes são dois operadores básicos de AG dos quais o desempenho é dependente. O tipo e a implementação desses operadores dependem da codificação e também do problema.

A probabilidade de cruzamento é a frequência na qual o cruzamento é realizado. Se não houver cruzamento, a descendência é uma cópia exata dos pais. Se ocorre o cruzamento, a descendência é constituída de partes dos cromossomos de ambos os pais. Os cruzamentos são realizados na esperança de que os novos cromossomos contêm partes boas dos cromossomos antigos e que portanto os novos cromossomos serão melhores.

A probabilidade de cruzamento adaptativo é definida pela Equação 2. Onde  $ng$  é o número total de gerações e  $t$  é a geração corrente.

$$P_{cru} = 0.9 - 8.8e^{\left(\frac{-0.3t^2}{2ng}\right)} \quad (2)$$

A probabilidade de mutação determina a frequência que partes dos cromossomos sofrerão mutação. Se não houver mutação, a descendência é gerada imediatamente após o cruzamento (ou copiada diretamente) sem nenhuma alteração. Se a mutação ocorre, uma ou mais partes do cromossomo é alterada. Se a probabilidade de mutação é 100%, todos os cromossomos são alterados, se é 0%, nenhum é alterado. A mutação em geral evita que o AG caia num extremo (mínimo ou máximo) local.

A probabilidade de mutação adaptativa é definida pela Equação 3. O intuito é que o início do processo evolucionário tenha uma baixa taxa de mutação, ou seja, pouca alteração das partes cromossômicas. Essa taxa adapta-se ao sistema conforme o passar das gerações.

$$P_{mut} = 0.9 - 8.8e^{\left(\frac{-0.2t^2}{2ng}\right)} \quad (3)$$

## 2.5 DISTÂNCIA EUCLIDIANA

As medidas de distância de uma maneira geral podem ser definidas como medidas de similaridade e dissimilaridade; na qual a primeira é para definir o grau de semelhança entre as instâncias e realizam o agrupamento de acordo com a sua coesão e a segunda de acordo com as diferenças dos atributos das instâncias. A Distância Euclidiana é uma das técnicas das distâncias de similaridade, é definida como a soma da raiz quadrada da diferença entre x e y em suas respectivas dimensões. A Equação 4 apresenta o calcula da Distância Euclidiana [10].

$$d = \sqrt{((x1 - x2)^2 + (y1 - y2)^2)} \quad (4)$$

Em termos relativos, a utilização da Distância Euclidiana se aplica melhor a dados não padronizados (ou seja, dados que não tem nenhum tipo de tratamento de adaptação de escala); e devido a isso o resultado final é insensível a dados muito diferentes do grupo [11].

## 2.6 ALGORITMO PROPOSTO

Como relatado anteriormente o algoritmo desenvolvido é baseado no *Differential Evolution*. O sistema é inicializado após a definição do tamanho fixo da população, número de execuções e número de avaliações.

A população inicial é gerada de forma aleatória, logo após é avaliada através da função de muitos objetivos e classificada através da distância de multidão.

Enquanto o número de avaliações não for suprido serão executadas as operações genéticas e demais ações, que serão repetidas até o número de execuções ser satisfeito.

Iniciando as operações genéticas, a seleção é realizada através da combinação da mutação com a distância de multidão. No torneio binário inicial são selecionados 6 indivíduos comparados de 2 a 2 para selecionar os 3 para a mutação, analisando um dos seguintes casos:

- Indivíduo 1 domina o indivíduo 2, neste caso o indivíduo 1 é selecionado;
- Caso não tenha indivíduos não dominados, seleciona-se o indivíduo com

a maior distância de multidão.

A probabilidade de mutação define se esse método de seleção com mutação será utilizado ou se aplicará uma seleção aleatória.

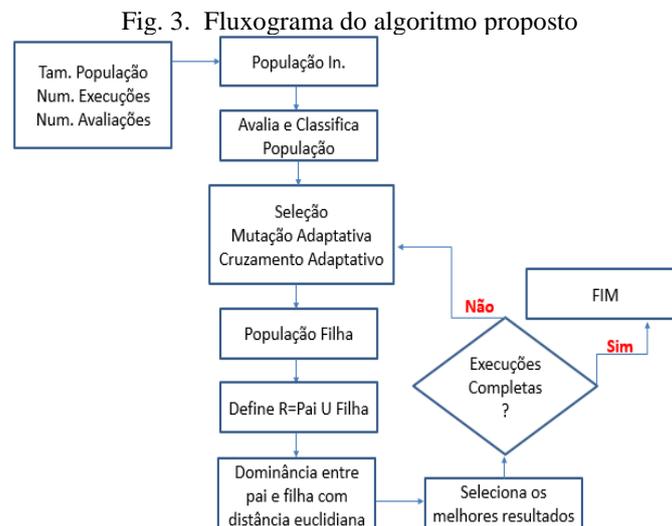
Logo após a mutação será realizado o cruzamento, também definido através do seu termo de probabilidade. A técnica utilizada será a do crossover exponencial para geração do vetor experimental. No fim aplica-se uma nova seleção por torneio entre o pai e indivíduo experimental gerado.

Ao finalizar o processo de operações genéticas é gerada a população de filhas. O próximo passo é unir a população de filhas com a população de pais, formando o cromossoma total.

A etapa seguinte consiste na avaliação da dominância entre população de pais e filhos, executada através da equação base da distância euclidiana. Com isso ordena o indivíduo melhor até o pior, observando a resolução da função de muitos objetivos.

Com a avaliação da dominância concluída é realizada a seleção dos melhores resultados. Nesta etapa são selecionados os primeiros indivíduos do cromossoma até que a população resultante seja do tamanho fixado no início da execução do algoritmo.

Na figura 3 é possível visualizar o fluxograma que exemplifica toda proposta implementada.



## 2.7 NSGA-III

A estrutura básica proposta para o algoritmo de muitos objetivos NSGA-III permanece semelhante ao algoritmo NSGA-II original com mudanças significativas em seu mecanismo de seleção. Mas ao contrário do NSGA-II, a manutenção da diversidade

entre os membros da população do NSGA-III é auxiliado pelo fornecimento e atualização adaptativa uma série de número de pontos de referência bem distribuídos [3].

O NSGA-III propõe substituir o operador de *crowding distance* presente no NSGA-II com as seguintes abordagens:

1. Classificação da população em níveis de não-dominância: identifica-se as fronteiras não-dominadas usando o princípio de dominância, da mesma maneira que no NSGA-II;
2. Determinação dos pontos de referência em um hiperplano: conjunto pré-definido de pontos de referência para garantir a diversidade de soluções obtidas. Uma vez que os pontos de referência criados são amplamente distribuídos em todo o hiperplano normalizado, as soluções obtidas também são susceptíveis a serem amplamente distribuídas próximas a fronteira pareto ótima;
3. Normalização adaptativa dos indivíduos da população: mantém a diversidade adaptativa no espaço gerado pelos indivíduos a cada geração;
4. Operação de associação: associa-se cada indivíduo da população com um ponto de referência;

Operação de preservação de nicho: operação com objetivo de diminuir o número de indivíduos associados a cada ponto de referência.

### 3 RESULTADOS E DISCUSSÕES

Os testes com o algoritmo foram executados observando as seguintes especificações:

- Número de execuções 3 objetivos: 400;
- Número de Execuções 5 objetivos: 600;
- Número de avaliações: 20;
- População 3 objetivos: 91;
- População 5 objetivos: 210.

Com isso obteve um total de 8000 gerações para os problemas de 3 objetivos e 12000 gerações para os problemas que contavam com 5 objetivos.

Os resultados não foram satisfatórios, demonstrando que a proposta ficou aquém do desejado. O algoritmo obteve melhores respostas nos testes com o problema DTLZ2, mesmo assim ficando distante da fronteira pareto ótima. Os fatos podem ser comprovados através das figuras 4,5,6e 7.

Fig. 4. Gráfico do teste com DTLZ1 de 3 objetivos

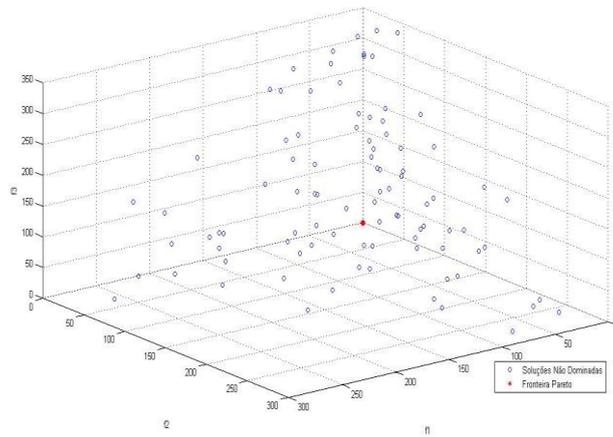


Fig. 5. Gráfico do teste com DTLZ2 de 3 objetivos

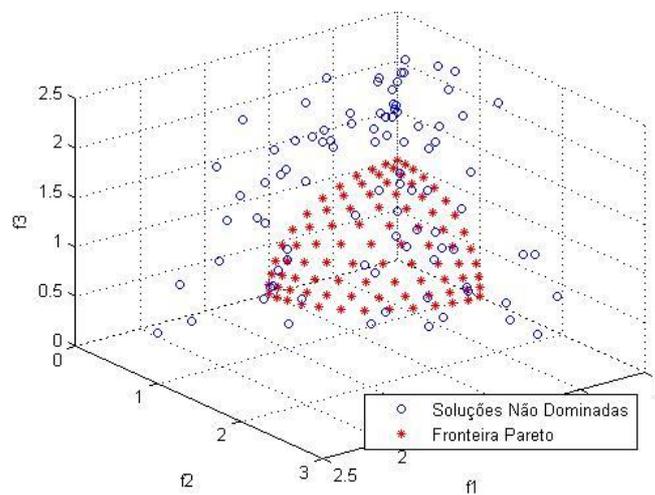


Fig. 6. Gráfico do teste com DTLZ1 de 5 objetivos

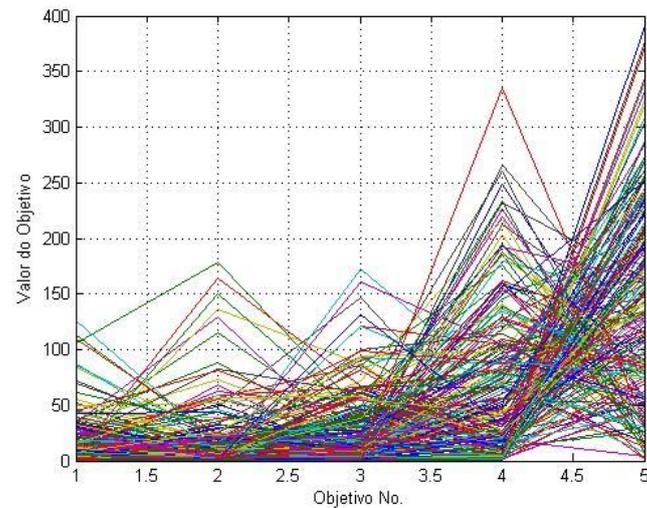
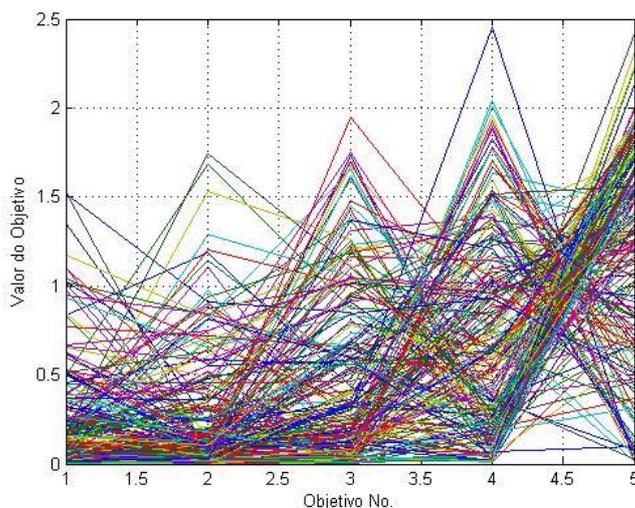


Fig. 7. Gráfico do teste com DTLZ2 de 5 objetivos



Em comparação com o NSGA-III[3], comprova-se a má elaboração do algoritmo proposto, onde percebe-se que o IGD (Distância Geracional Invertida) do algoritmo utilizado como referência nos testes é melhor em todos os casos. Como pode ser visualizado na tabela 1.

#### 4 CONCLUSÕES

O artigo propôs o desenvolvimento de um algoritmo de Evolução Diferencial Adaptativo para resolução de problemas com muitos objetivos. Porém, como comprovado pelos resultados, os testes não foram satisfatórios.

As inovações propostas ao Differential Evolution, não se relacionaram bem, o que também impediu que a proposta do sistema adaptativo de ir melhorando os resultados a cada geração.

Para os próximos trabalhos é recomendado rever como foram implementadas as inovações do código e avaliar quais as que devem ser mantidas. Possibilitando um algoritmo com uma linha de trabalho mais bem definida.

TABELA I  
COMPARAÇÃO DOS IGDS DO NSGA-III COM OS DO ALGORITMO PROPOSTO

Problema	Obj	Gerações	IGD	NSGA-III	Proposta
DTLZ1	3	400	Melhor	$4,880 \times 10^{-3}$	$2,353 \times 10^{-1}$
			Médio	$1,308 \times 10^{-3}$	$6,059 \times 10^{-1}$
			Pior	$4,880 \times 10^{-3}$	$1,089 \times 10^{-2}$
	5	600	Melhor	$5,116 \times 10^{-4}$	$1,259 \times 10^{-1}$
			Médio	$1,799 \times 10^{-4}$	$4,321 \times 10^{-1}$
			Pior	$1,979 \times 10^{-3}$	$7,621 \times 10^{-1}$

DTLZ2	3	400	Melhor	$1,262 \times 10^{-3}$	$4,243 \times 10^1$
			Médio	$1,357 \times 10^{-3}$	$5,339 \times 10^1$
			Pior	$2,114 \times 10^{-3}$	$6,473 \times 10^1$
	5	600	Melhor	$4,254 \times 10^{-3}$	$5,877 \times 10^1$
			Médio	$4,982 \times 10^{-3}$	$6,735 \times 10^1$
			Pior	$5,862 \times 10^{-3}$	$7,730 \times 10^1$

## REFERÊNCIAS

H.M. Ares, L.C.B. dos Santos, R.C. Silva, L.M.O. de Queiroz, "Algoritmo evolutivo para problemas de otimização multiobjetivo com incertezas", presente no XXXIX Simpósio Brasileiro de Pesquisa Operacional, Fortaleza, CE, Agosto, 2007.

A.R.T.T. Argoud, E.V.G. Filho, A.J. Tiberti, "Algoritmo de agrupamento para formação de módulos de arranjo físico," Revista Gestão de Produção v.15 n.2 pp 393-405, São Carlos, SP, 2008.

DEB, K., JAIN, H., An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part i: Solving problems with box constraints, 2013. doi:10.1109/TEVC. 2013.2281534.

PRICE, K. V. e STORN, R. M. Differential evolution - A simple evolution strategy for fast optimization. *Dr. Dobb's Journal*. 22, 18-24. 1997

KWAN, C., YANG, F. e CHANG, C. A., Differential Evolution Variant of NSGA-II for Real World Multiobjective Optimization, *ACAL 2007*, LNAI 4828, pp. 345–356, Springer-Verlag Berlin Heidelberg, 2007

REGULWAR D. G., CHOUDHARI, S. A. E RAJ P. A. Differential Evolution Algorithm with Application to Optimal Operation of Multipurpose Reservoir. *J. Water Resource and Protection*, 2, 560-568. 2010.

ROBIC, T., FILIPIC, B., DEMO: Differential Evolution for Multiobjective Optimization. In: Coello Coello, C.A., Hernandez Aguirre, A., Zitzler, E. (eds.) *EMO 2005*. LNCS, vol. 3410, pp. 520–533. Springer, Heidelberg. 2005

COELHO, C. A., LAMONT, Gary B. e VELDHUIZEN, D. A., Evolutionary Algorithms for solving multi-objective problems, chapter 1, page 11. *Genetic and Evolutionary Computation*. Springer, 2 ed., 2002.

A.M.. Barbosa, L.C. Ribeiro, J.M.O. Arantes "Algoritmo Genético Multiobjetivo: Sistema Adaptativo com Elitismo," 9ª Brazilian Conference on Dynamics Control and their Applications. 2010.

R. Linden, "Técnicas de Agrupamento", *Revista de Sistemas de Informação da FSMA*, n.4 p.18-36, 2009 [Online]. Disponível: <http://www.fsma.edu.br/si/sistemas.html>

Statistica. "Cluster Analysis". [Online]. Disponível: <http://www.statsoft.com/textbook/clueter-analysis/#d>