



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Sistemas de Potência e Automação



Despacho económico de unidades com funções de custo não convexas usando enxames

JOÃO CARLOS BENTO PONTES

(Licenciado em Engenharia Electrotécnica)

Dissertação para a obtenção do grau de Mestre em Engenharia Electrotécnica – ramo de Energia

Orientadores:

Equiparado a Professor Adjunto Rui Pestana

Professor Adjunto Pedro Fonte

Júri:

Presidente:

Doutor Jorge Alberto Mendes de Sousa

Vogais:

Doutor Victor Manuel Fernandes Mendes

Equiparado a Professor Adjunto Rui Pestana

Professor Adjunto Pedro Fonte

Setembro de 2014



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Sistemas de Potência e Automação



**Despacho económico de unidades com funções de custo
não convexas usando enxames**

JOÃO CARLOS BENTO PONTES

(Licenciado em Engenharia Electrotécnica)

Dissertação para a obtenção do grau de Mestre em Engenharia Electrotécnica – ramo de Energia

Orientadores:

Equiparado a Professor Adjunto Rui Pestana

Professor Adjunto Pedro Fonte

Júri:

Presidente:

Doutor Jorge Alberto Mendes de Sousa

Vogais:

Doutor Victor Manuel Fernandes Mendes

Equiparado a Professor Adjunto Rui Pestana

Professor Adjunto Pedro Fonte

Setembro de 2014

Dissertação realizada sob a orientação de:

Equiparado a Professor Adjunto Rui Pestana

Professor Adjunto Pedro Fonte

Instituto Superior de Engenharia de Lisboa

Aos meus avós

Agradecimentos

Esta dissertação foi um trabalho longo e por vezes complicado, pelo que sem a ajuda que recebi, a dissertação provavelmente ficaria mais pobre. Assim, agradeço em primeiro lugar ao meu orientador, eng^o Rui Pestana e ao meu co-orientador, eng^o Pedro Fonte, pela disponibilidade e prontidão com que me guiaram ao longo destes meses de trabalho e pelas reuniões a três que fomos fazendo, nas quais aprendi imenso. Creio que contribuíram positivamente para o meu desenvolvimento pessoal, académico e profissional.

Reconheço que estes anos de formação, que agora acabam, foram um processo contínuo de aprendizagem pessoal e académica e por vezes o final pode ter parecido impossível. Nestas situações, reconheço que sem colegas e amigos ao meu lado a ajudarem-me, teria sido bastante mais difícil chegar aqui. Espero ter estado ao seu lado quando precisaram.

Agradeço à minha namorada que acompanhou o meu percurso todo nesta dissertação.

Por último, agradeço à minha base: o meu irmão, Luís, a minha mãe, o meu pai, a minha avó e o meu avô.

Resumo

Nesta dissertação, o problema de despacho económico é estudado, especificamente a sua optimização através de métodos convencionais, *i.e.* métodos com base no gradiente e métodos baseados em heurísticas. Métodos diferentes são usados para otimizar funções de teste e casos estudados de despacho económico, sendo os resultados comparados entre si.

As funções de teste são as comumente utilizadas em literatura para validar a capacidade de busca, tempo de processamento e robustez de algoritmos de optimização.

Os casos estudados são 4 e de despacho económico, sendo a diferença entre os mesmos a existência ou não de restrições, funções de custo convexas ou não convexas e número grupos.

Nesta dissertação é proposta uma nova meta-heurística, que é um híbrido do *Particle Swarm Optimization* e do *Sensing Cloud Optimization*. A esta meta-heurística é dado o nome de *Particle Cloud Optimization*.

Palavras-chave:

Despacho económico, Particle Swarm Optimization, Bee Colony Optimization, Cockroach Swarm Optimization, Sensing Cloud Optimizadon, Particle Cloud Optimization, meta-heurísticas, Lambda iteration.

Abstract

This dissertation deals with the economic dispatch problem regarding its optimization by conventional methods, *i.e.* methods based upon gradient and methods based upon heuristics. Different methods optimize test functions and case studies of economic dispatch, and the results are compared.

The test functions are usually used in literature to validate search ability, processing time and robustness of optimization algorithms.

The case studies are 4 and of economic dispatch. They differ on the existence or not of restrictions, convex or non convex functions and number of groups.

In this dissertation it is proposed a new meta-heuristic, which is a hybrid between Particle Swarm Optimization and Sensing Cloud Optimization. To this meta-heuristic is given the name of Particle Cloud Optimization.

Keywords:

Economic dispatch, Particle Swarm Optimization, Bee Colony Optimization, Cockroach Swarm Optimization, Sensing Cloud Optimizadon, Particle Cloud Optimization, meta-heuristics, Lambda iteration.

Lista de símbolos

a_i, b_i, c_i	Coeficientes da função de custo do grupo i	
\mathbf{B}_0^T	Matriz de susceptâncias transposta	
\mathbf{B}_{00}	Constante da equação de perdas	
\mathbf{B}_{ij}	Matriz da susceptância nodal	S
DR_i	Rampa de descida (Down ramp) de potência de um grupo i	MW/h
e_i, f_i	Coeficientes para o efeito válvula do grupo i	
$F_{i,t}(P_{i,t})$	Função de custo no intervalo t por grupo e $P_{i,t}$ a geração no	€
F_T	Função de custo total da geração eléctrica	€
\mathbf{G}_{ij}	Matriz da condutância nodal	S
NP_i	Número de zonas proibidas do grupo i	
P_C	Potência activa consumida	MW
P_G	Potência activa gerada	MW
P_i	Potência gerada pelo grupo i	MW
P_L	Potência activa de perdas	MW
$P_{máx}$	Potência máxima	MW
$P_{mín}$	Potência mínima	MW
P_{perdas}	Perdas por efeito de <i>Joule</i> na rede	MW
P_i^0	Potência activa no intervalo de tempo anterior	MW
P_{ji}^k	Trânsito de potência activa do barramento j para o i pela	MW
$P_{i,j}^{LB}$	Limite inferior de potência da zona proibida j do grupo i	MW
$P_{i,j}^{UB}$	Limite superior de potência da zona proibida j do grupo i	MW
Q_C	Potência reactiva consumida	MVAr
Q_G	Potência reactiva gerada	MVAr
Q_L	Potência reactiva de perdas	MVAr
S_i	Potência aparente do barramento i	MVA
S_i^C	Potência aparente consumida no barramento i	MVA
S_i^G	Potência aparente gerada no barramento i	MVA
SR_i	Reserva girante (<i>spinning reserve</i>)	MW
UR_i	Rampa de subida (Up ramp) de potência de um grupo i	MW/h
θ	Argumento da tensão	Grau
ΔP_i	Erro aceitável de potência activa	MW
ΔQ_i	Erro aceitável de potência reactiva	MVAr
$\$$	Unidade monetária genérica	

Abreviaturas

<i>a priori</i>	Do latim, “de antes” ou “do anterior”
<i>i.e.</i>	Do latim, <i>id est</i> (isto é)
<i>e.g.</i>	Do latim, <i>exempli grati</i> (por exemplo)
ABC	<i>Artificial bee colony</i>
ACO	<i>Ant colony optimization</i>
BCO	<i>Bee colony optimization</i>
BFO	<i>Bacteria foraging optimization</i>
CSO	<i>Cockroach swarm optimization</i>
EP	<i>Evolutionary programming</i>
ES	<i>Evolutionary strategies</i>
FEP	<i>Fast evolutionary programming</i>
FEPSPD	<i>Fast evolutionary programming with swarm direction</i>
GA	<i>Genetic algorithms</i>
IEP	<i>Improved evolutionary programming</i>
IPSO	<i>Improved particle swarm optimization</i>
ITS	<i>Improved taboo search</i>
NN	<i>Neural networks</i>
PCG	Problema de Comissionamento de Grupos
PCO	<i>Particle cloud optimization</i>
PDE	Problema de Despacho Económico
PSO	<i>Particle swarm optimization</i>
PTP	Problema de Trânsito de Potências
SA	<i>Simulated annealing</i>
SCO	<i>Sensing cloud optimization</i>
SEE	Sistema Eléctrico de Energia
TS	<i>Taboo search</i>
UC	<i>Unit commitment</i>

Convenções

1. Sistema de unidades de medida

Neste texto é utilizado, salvo menção expressa em contrário, o Sistema Internacional (S.I.) de unidades de medida, bem como os símbolos e abreviaturas normalmente usadas para os múltiplos e os submúltiplos das diversas unidades. Por ser um sistema de unidades bem conhecido essas abreviaturas não são citadas na secção “Abreviaturas” contida neste trabalho.

2. Utilização de estrangeirismos

Ao longo desta dissertação utilizam-se alguns estrangeirismos fundamentalmente por dois motivos. O primeiro refere-se a palavras que ainda não se encontram definidas na língua portuguesa e o segundo refere-se a palavras em que a mera tradução linguística faz perder algum significado técnico atribuído às mesmas no original.

Todos os estrangeirismos referidos neste ponto, são como tal, assinalados ao longo do texto em itálico.

Índice

1. Introdução	21
1.1. Motivação.....	21
1.2. Enquadramento.....	22
1.3. Objectivos	23
1.4. Estrutura da dissertação.....	23
2. Definição do problema	27
2.1. Introdução	27
2.2. Multi-combustível	29
2.3. Restrições	29
2.3.1. Potência mínima e máxima	29
2.3.2. Restrição de igualdade	30
2.3.3. Zonas proibidas	30
2.3.4. Rampa de incremento e decréscimo de potência	31
2.3.5. Reserva girante.....	32
2.4. Formulação do problema.....	33
2.5. Trânsito de potências e perdas nas linhas.....	33
2.5.1. Equações do trânsito de potências.....	34
2.5.1.1. Tipos de barramentos	35
2.5.1.2. Resolução de trânsito de potências.....	36
2.5.1.3. Resolução de trânsito de potências pelo modelo DC	36
2.5.2. Definição matemática de função convexa.....	38
2.5.2.1. Condições de convexidade	39
2.6. Não convexidade das funções de custo	39
2.6.1. Efeito das válvulas de vapor.....	40
3. Visão global (<i>Overview</i>).....	47
3.1. Introdução	47
3.2. Breve evolução do problema de despacho económico.....	47
3.3. Teorema de <i>Karush Kuhn-Tucker</i>	48

3.4.	Métodos de otimização convencionais.....	48
3.4.1.	Método do gradiente.....	49
3.4.2.	Lambda iteration	49
3.5.	Métodos não convencionais	51
3.5.1.	Meta-heurísticas	51
3.5.2.	Origem do estudo da Inteligência de Enxames	51
3.5.3.	Comportamento de enxame.....	52
3.5.4.	<i>Particle Swarm Optimization</i>	54
3.5.5.	<i>Bee Colony Optimization</i>	56
3.5.6.	<i>Cockroach Swarm Optimization</i>	59
3.5.7.	<i>Sensing Cloud Optimization</i>	61
3.5.8.	<i>Particle Cloud Optimization</i>	67
4.	Otimização através de Inteligência de enxames.....	73
4.1.	Introdução	73
4.2.	Funções de teste	74
4.2.1.	Resultados	76
5.	Caso estudado.....	81
5.1.	Introdução	81
5.2.	Penalização de soluções não óptimas.....	81
5.3.	Caso estudado 1 – 3 grupos.....	82
5.4.	Caso estudado 2 – 10 grupos.....	85
5.5.	Caso estudado 3.....	86
5.6.	Caso estudado 4.....	87
6.	Conclusão e futuros desenvolvimentos	93
7.	Bibliografia	95
	Anexo 1 – Pseudo-código do Particle Swarm Optimization.....	103
	Anexo 2 – Pseudo-código do Bee Colony Optimization	105
	Anexo 3 – Pseudo-código do Cockroach Swarm Optimization.....	107
	Anexo 4 – Fluxograma do Sensing Cloud Optimization	109

Anexo 5 – Pseudo-código do Particle Cloud Optimization	111
Anexo 6 – Evolução de cada algoritmo para a função <i>De Jong</i> com 5 dimensões.....	113
Anexo 7 – Evolução de cada algoritmo para a função <i>De Jong</i> com 10 dimensões.....	115
Anexo 8 – Evolução de cada algoritmo para a função <i>Rosenbrock</i> com 5 dimensões	117
Anexo 9 – Evolução de cada algoritmo para a função <i>Rosenbrock</i> com 10 dimensões.....	119
Anexo 10 – Evolução de cada algoritmo para a função <i>Schaffer</i> com 2 dimensões	121
Anexo 11 – Evolução de cada algoritmo para a função <i>Griewank</i> com 5 dimensões.....	123
Anexo 12 – Evolução de cada algoritmo para a função <i>Griewank</i> com 10 dimensões.....	125
Anexo 13 – Evolução de cada algoritmo para a função <i>Rastrigin</i> com 5 dimensões.....	127
Anexo 14 – Evolução de cada algoritmo para a função <i>Rastrigin</i> com 10 dimensões.....	129
Anexo 15 – Evolução da potência por grupo para o caso estudado 1 com o PSO, BCO, CSO, SCO e PCO	131
Anexo 16 – Evolução da potência por grupo para o caso estudado 1 com todos os algoritmos	133

Índice de figuras

Figura 1-1 – Repartição da produção eléctrica em Portugal, 2012 [2].....	22
Figura 2-1 – Esquema exemplificativo de um SEE.	27
Figura 2-2 – Fluxograma da combinação dinâmica no PDE.....	28
Figura 2-3 - Função de custo com zonas proibidas.	31
Figura 2-4 – Esquema unifilar de um sistema com 2 barramentos [12].....	35
Figura 2-5 – Conjunto não convexo [18].	38
Figura 2-6 – Definição geométrica de função convexa.....	38
Figura 2-7 – Representação da expressão (34) [19].	39
Figura 2-8 – Representação de uma função convexa [19].	40
Figura 2-9 – Efeito <i>Joule-Thomson</i>	41
Figura 2-10 – Função de custo (azul) com legenda à esquerda e sua derivada (verde) com eixo à direita [22].....	42
Figura 2-11 – Progressão combinatória do seccionamento de funções de custo.....	43
Figura 2-12 – Aproximação linear da função de custo com efeito de válvula.	43
Figura 3-1 – Exemplo de <i>lambda iteration</i>	50
Figura 3-2 – Estigmergia em formigas.....	53
Figura 3-3 – Velocidade imposta a uma dada partícula.	55
Figura 3-4 – Evolução do enxame do PSO na busca pela melhor solução.	56
Figura 3-5 – Representação da dança da abelha para comunicar.....	57
Figura 3-6 – Evolução do enxame do BCO na busca pela melhor solução.	58
Figura 3-7 – Comportamento de uma partícula no CSO.....	60
Figura 3-8 - Evolução do algoritmo CSO na busca pela melhor solução.	60
Figura 3-9 – Contour plot da partícula $\mathbf{Xq}(1)$ [41].	62
Figura 3-10 – Regressão polinomial para a dimensão $\mathbf{X}(2)$ [41].	63
Figura 3-11 – Regressão polinomial para a dimensão $\mathbf{x}(1)$ [41].	63
Figura 3-12 – Cálculo da distância de cada partícula à central [41].	64
Figura 3-13 – Valores de fitness em função da distância à partícula central.	65
Figura 3-14 – Característica de $\mathbf{Fs1}$ [41].	66
Figura 3-15 – Característica de $\mathbf{Fs2}$ [41].	66
Figura 3-16 – Evolução do PSO (vermelho) e do SCO (verde).	67
Figura 3-17 – Representação do PCO face ao PSO e SCO.	68
Figura 4-1 – a) Evolução da melhor simulação e b) Evolução média de cada algoritmo para a função <i>De Jong</i> com 5 dimensões.....	76
Figura 4-2 – Representação gráfica do desempenho de cada algoritmo.	78

Figura 5-1 – a) Função de custo do caso 1 e b) derivada da função de custo do caso 1.	83
Figura 5-2 – Resultado do caso 1 com o PSO.....	83
Figura 5-3 – Resultado do caso 1 com o PSO comparado com uma repartição igualitária e com uma maximização sucessiva.....	84
Figura 7-1 – Fluxograma do SCO [41].	109
Figura 7-2 – Evolução da melhor simulação para a função De Jong com 5 dimensões.	113
Figura 7-3 – Evolução média para a função De Jong com 5 dimensões.....	113
Figura 7-4 – Evolução da melhor simulação para a função De Jong com 10 dimensões.	115
Figura 7-5 – Evolução média para a função De Jong com 10 dimensões.....	115
Figura 7-6 – Evolução da melhor simulação para a função Rosenbrock com 5 dimensões.....	117
Figura 7-7 – Evolução média para a função Rosenbrock com 5 dimensões.	117
Figura 7-8 – Evolução da melhor simulação para a função Rosenbrock com 10 dimensões....	119
Figura 7-9 – Evolução média para a função Rosenbrock com 10 dimensões.	119
Figura 7-10 – Evolução da melhor simulação para a função Schaffer com 2 dimensões.	121
Figura 7-11 – Evolução média para a função Schaffer com 2 dimensões.....	121
Figura 7-12 – Evolução da melhor simulação para a função Griewank com 5 dimensões.....	123
Figura 7-13 – Evolução média para a função Griewank com 5 dimensões.	123
Figura 7-14 – Evolução da melhor simulação para a função Griewank com 10 dimensões....	125
Figura 7-15 – Evolução média para a função Griewank com 10 dimensões.	125
Figura 7-16 – Evolução da melhor simulação para a função Rastrigin com 5 dimensões.....	127
Figura 7-17 – Evolução média para a função Rastrigin com 5 dimensões.	127
Figura 7-18 – Evolução da melhor simulação para a função Rastrigin com 10 dimensões.....	129
Figura 7-19 – Evolução média para a função Rastrigin com 10 dimensões.	129
Figura 7-20 – Caso estudado 1 com PSO.....	131
Figura 7-21 – Caso estudado 1 com BCO.....	131
Figura 7-22 – Caso estudado 1 com CSO.	131
Figura 7-23 – Caso estudado 1 com SCO.	132
Figura 7-24 – Caso estudado 1 com PCO.	132

Índice de tabelas

Tabela 2-1 – Variáveis do problema	34
Tabela 2-2 – Tipos de barramentos [12]	36
Tabela 3-1 – Funções de custo do caso estudado em [24]	50
Tabela 3-2 – Lambda iteration (iterações e valores de potência)	50
Tabela 4-1 – Parâmetros do PSO	73
Tabela 4-2 – Parâmetros do BCO	73
Tabela 4-3 – Parâmetros do CSO	74
Tabela 4-4 – Parâmetros do SCO	74
Tabela 4-5 – Funções de teste utilizadas	75
Tabela 4-6 – Parâmetros das simulações	76
Tabela 4-7 – Resultados de optimização das funções de teste. *tempo necessário para chegar ao melhor valor de fitness.	77
Tabela 5-1 – Parâmetros das simulações	81
Tabela 5-2 – Resultados para o caso estudado 2	85
Tabela 5-3 – Valor de produção de cada grupo para o caso estudado 2	85
Tabela 5-4 – Resultados para o caso estudado 3	86
Tabela 5-5 – Valor de produção de cada grupo para o caso estudado 3	87
Tabela 5-6 – Resultados para o caso estudado 4	88
Tabela 5-7 – Valor de produção de cada grupo para o caso estudado 4	89

Capítulo I

Introdução

Resumo

Neste capítulo são descritas as motivações e objectivos desta dissertação, bem como o seu enquadramento. A estrutura dos capítulos é também apresentada.

1. Introdução

Despacho económico é o problema de alocação de potência por grupo, respeitando as restrições que existam. O seu objectivo é o de minimizar o custo de produção. A sua optimização pode ser feita por métodos convencionais, baseados no método do gradiente, mas por vezes as funções de custo dos grupos não são próprias para a aplicação de métodos convencionais, por terem mínimos locais. A aplicação de meta-heurísticas em despacho económico é uma das formas de evitar a dificuldade de aplicar métodos convencionais em certas funções de custo, que por características técnicas do grupo gerador ou limitações de operação da rede poderão não ser diferenciáveis. Assim, uma distinção existe à partida entre métodos convencionais e meta-heurísticas, em que por método convencional se entende aquele que se baseia apenas em pressupostos matemáticos sem incorporar heurísticas. Aqui se incluem o método do gradiente, *lambda iteration*, programação dinâmica, método de Newton, programação linear e *interior point method* entre outras. Por outro lado, uma heurística é uma abordagem empírica a um problema, sendo que a solução obtida poderá não ser a ideal. Uma meta-heurística é um método aplicado que se baseia no conceito de heurística.

Heurísticas podem ser utilizadas na resolução de qualquer problema matemático, mas a sua mais-valia será encontrada em problemas de elevada complexidade e com necessidade de rapidez de solução, como por exemplo o problema de despacho económico.

1.1. Motivação

A motivação para esta dissertação pode ser dividida em duas: económica e técnica. A motivação económica prende-se com a vontade óbvia de garantir o despacho económico, minimizando ao máximo o seu custo. Sabendo que o custo está associado aos recursos utilizados, é de salientar que os recursos fósseis ainda são uma fatia considerável de cerca de 36% (figura 1-1) do consumo de Portugal, excluindo a contribuição da importação. Além das preocupações ambientalistas, os recursos fósseis também são susceptíveis às variações no preço dos combustíveis nos mercados, o que pode ser dispendioso financeiramente na operação de uma central, caso se verifique uma subida no preço dos combustíveis. Portanto, do ponto de vista económico, faz todo

o sentido que os grupos operem de tal forma que o custo global de geração seja o mais reduzido.

A motivação técnica surge da dificuldade em aplicar métodos convencionais em funções de custo não convexas. Com a aplicação de meta-heurísticas, esta dificuldade é contornada, visto que estas têm uma forma diferente de atingir o ponto óptimo. As meta-heurísticas implementadas baseiam-se em inteligência de enxames e encontram no comportamento animal formas de atingir um resultado, mesmo que o espaço de busca seja altamente restrito, pelo que a sua aplicação ao problema de despacho económico com restrições, em que as funções de custo sejam não convexas, poderá ser proveitoso.

1.2. Enquadramento

Em qualquer sistema eléctrico de energia (SEE) existe um *mix* energético, uma combinação de tecnologias de produção eléctrica para fornecer energia à carga. Este *mix* energético é ajustado de acordo com as circunstâncias económicas, técnicas e ambientais. Economicamente e tecnicamente, o decisor operacional decide fundamentando-se no despacho económico, que é o problema de alocação de potência gerada, sendo a solução pretendida o valor de produção eléctrica por cada grupo para que as restrições sejam satisfeitas da forma mais económica possível [1]. No caso português, a produção eléctrica em 2012 baseou-se em 36% de combustíveis fósseis; uma fatia decrescente em relação a 2011 veio de fontes renováveis e de uma crescente importação. O mix energético português para 2012 pode ser observado na figura 1-1.

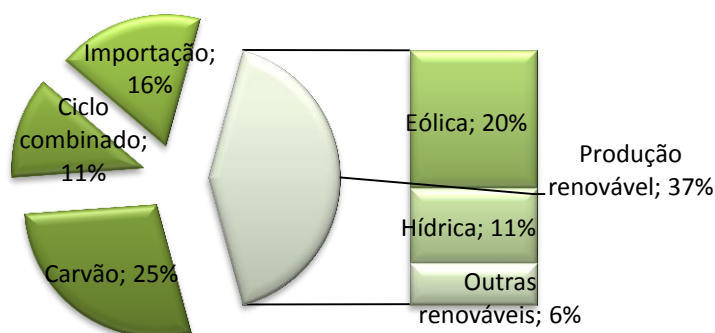


Figura 1-1 – Repartição da produção eléctrica em Portugal, 2012 [2].

O problema de despacho económico (PDE) faz parte do problema maior de *scheduling*, que engloba o problema de comissionamento de grupos (PCG) (*unit*

commitment), a análise de segurança restrita (*security constrained analysis*), avaliação de reserva (*reserve assessment*), avaliação de fiabilidade (*reliability assessment*), previsão (*forecasting*), *market clearing* e análise de risco (*risk analysis*). Destes segmentos, o PDE é um sub-segmento do PCG. Este é responsável por definir o estado de cada grupo como ligado/desligado para cada hora, tendo em conta a previsão de carga para as horas seguintes, os tempos e custos de arranque de grupo, tempos mínimo e máximo de operação/paragem de grupo [3, pp. 1–19]. Os algoritmos de PCG optimizam a produção de energia através de um sistema de *ranking* e não de alocação de geração como é feito pelos algoritmos de PDE [3, pp. 1–19]. O comissionamento de grupos cria um planeamento (*schedule*), que habitualmente é para as 4 a 48 horas seguintes, com intervalos de 5 minutos a 1 hora para actualizar o planeamento [3, pp. 1–19].

1.3.Objectivos

Esta dissertação pretende estudar diferentes meta-heurísticas aplicadas ao problema de despacho económico e comparar os seus desempenhos. Assim, os objectivos são:

- Estudar o comportamento de enxames na resolução de problemas matemáticos convexos e não convexos.
- Formular o problema completo de despacho económico com a sua função objectivo e todas as restrições.
- Contribuir com um novo modelo híbrido de meta-heurística.
- Analisar o comportamento das meta-heurísticas na resolução de problemas de despacho económico.
- Comparação com outros métodos tradicionais.
- Concluir acerca dos resultados.

1.4.Estrutura da dissertação

Seis capítulos compõem esta dissertação. No primeiro capítulo são descritas as motivações e objectivos desta dissertação, bem como o seu enquadramento. A estrutura dos capítulos é também apresentada.

No segundo capítulo é enunciado o problema de despacho económico e explicadas as restrições inerentes ao mesmo. É abordado o problema de trânsito de potências e o cálculo de perdas numa rede eléctrica.

No terceiro capítulo é feita uma visão global do problema de despacho económico e dos métodos convencionais e meta-heurísticas utilizadas para o resolver.

No quarto capítulo é apresentado um conjunto de funções de teste de referência para a aferição das performances de algoritmos de optimização. São também validadas as performances das heurísticas apresentadas na resolução das funções de referência.

No quinto capítulo são resolvidos os casos estudados de despacho económico. Os casos estudados são quatro. O primeiro tem 3 grupos de função de custo igual, não convexa. O segundo tem 10 grupos, com funções de custo diferentes, não convexas e com *multi-fuels*. O terceiro tem 15 grupos, com funções de custo convexas e presença de perdas. O quarto tem 40 grupos, com funções de custo não convexas.

No sexto capítulo são apresentadas as conclusões desta dissertação e propostos futuros desenvolvimentos.

Capítulo II

Definição do problema

Resumo

Neste capítulo é enunciado o problema de despacho económico e explicadas as restrições inerentes ao mesmo. É abordado o problema de trânsito de potências e o cálculo de perdas numa rede eléctrica.

2. Definição do problema

O problema de despacho económico é uma área importante na operação de um SEE. Fundamentalmente, o objectivo da sua resolução prende-se com a avaliação da potência que cada grupo gerador ligado à rede deve produzir de modo a minimizar o custo de produção respeitando as restrições técnicas e de carga. Na figura 2-1 está representado um esquema exemplificativo de um sistema eléctrico de energia com respectiva geração, carga e linhas de transmissão de energia eléctrica.

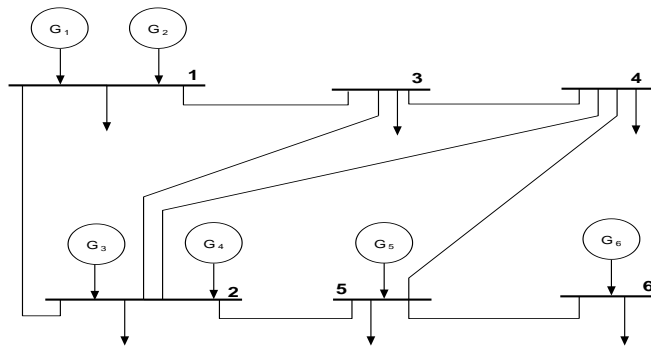


Figura 2-1 – Esquema exemplificativo de um SEE.

2.1. Introdução

O problema de despacho económico centra-se na minimização de custos relacionados com a geração de energia eléctrica, principalmente os custos das unidades térmicas que consomem combustíveis fósseis. A sua formulação básica está indicada em (1).

$$\min F_T = \min \sum_{t=1}^T \sum_{i=1}^N F_{i,t}(P_{i,t}) \quad (1)$$

Sendo F_T o custo total para o intervalo T , N o número total de grupos, $F_{i,t}$ a função de custo no intervalo t por grupo i e $P_{i,t}$ a geração no intervalo t de cada grupo i . Tradicionalmente, a função de custo de cada grupo pode ser definida por (2):

$$F_i = c_i P_i^2 + b_i P_i + a_i \quad (2)$$

Em que a , b e c são os coeficientes da função de custo do grupo i .

A sequência do problema de despacho económico pode ser temporal. Neste caso, o óptimo de um intervalo t , poderá não ser o óptimo do conjunto, ou seja, o óptimo de um certo intervalo temporal não é necessariamente a soma dos óptimos de cada unidade temporal.

Na figura 2-2 está representado um fluxograma sobre o PCG para um dado PDE.

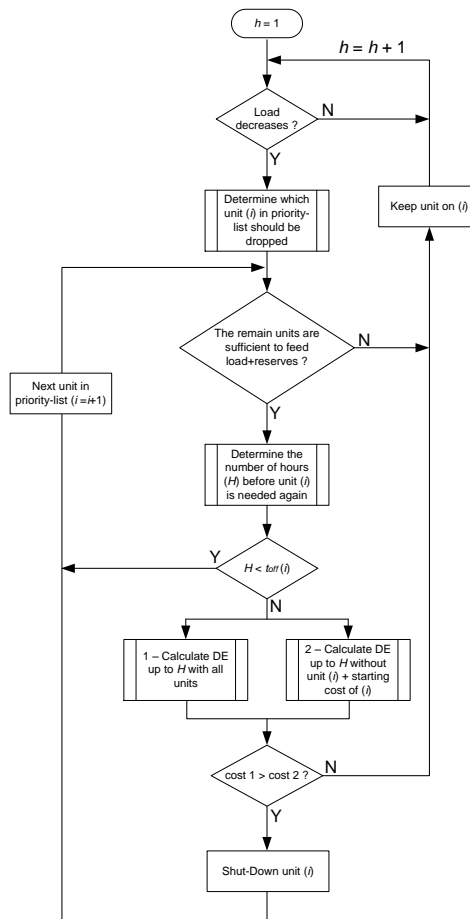


Figura 2-2 – Fluxograma da combinação dinâmica no PDE.

Como se pode ver, a cada unidade de tempo é realizada uma análise à carga. O comissionamento de grupos irá determinar quantos e quais os grupos que se mantêm em operação. Um dos métodos empregues para determinar quais os grupos que entram e os que saem de geração é o da lista prioritária [4]. Uma lista é criada com um critério que estabeleça quais os grupos que devem sair em primeiro lugar. Por norma, este critério é o custo médio de operação à potência máxima.

outros [6]. Abaixo de P_{min} , o controlo poderá ser instável. A expressão matemática desta restrição é dada por (4).

$$P_{i,min} \leq P_i \leq P_{i,máx} \quad (4)$$

2.3.2. Restrição de igualdade

Num sistema eléctrico regulado existe a obrigatoriedade de gerar potência eléctrica suficiente para alimentar a carga e compensar as perdas no trânsito de potência. Nesse sentido, resulta uma restrição de igualdade que pode ser definida por (5).

$$\sum_{i=1}^N P_i = P_C + P_{Perdas} \quad (5)$$

Com esta restrição pretende-se manter a frequência constante. Quando a potência produzida fica aquém da consumida mais a perdida, a frequência do SEE desce. Por outro lado, caso a potência produzida seja maior que a consumida mais as perdas, a frequência aumentará.

2.3.3. Zonas proibidas

Em algumas centrais existem grupos geradores em que é desaconselhável a operação dentro de certos intervalos de potência. Estas restrições operacionais podem estar ligadas a vibrações nos veios ou, particularmente em grupos a carvão nas zonas de funcionamento em que existe intermitência na operação dos moinhos de carvão. Nesse sentido consideram-se existir zonas de funcionamento que são de evitar em regime permanente pelo que se consideram como zonas proibidas de funcionamento. Com as zonas proibidas, a função de custo deixa de ser contínua, passando a ser definida por ramos (6). Na figura 2-3 está ilustrada uma possível função de custo com zonas proibidas [7].

Consideremos a disjunção de desigualdades duplas (6)

$$\left\{ \begin{array}{l} P_i^{min} \leq P_{i,t} \leq P_{i,1}^{LBj} \\ P_{i,j-1}^{UBj} \leq P_{i,t} \leq P_{i,j}^{LBj}, \quad j = 2,3, \dots, NP_i \\ P_{i,NP_i}^{UBj} \leq P_{i,t} \leq P_i^{max} \end{array} \right. \quad (6)$$

definidas para i e j . Em que j é o índice da zona proibida, i o índice do grupo, LB_j o limite inferior e UB_j o superior da respectiva zona proibida e NP_i o número de zonas proibidas.

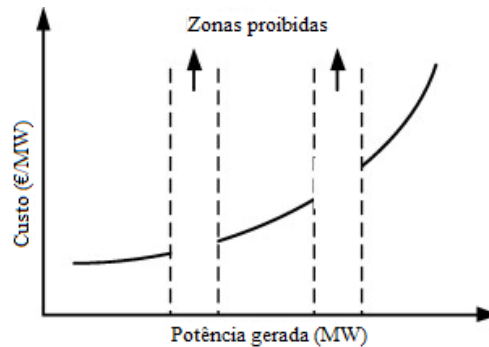


Figura 2-3 - Função de custo com zonas proibidas.

2.3.4. Rampa de incremento e decremento de potência

Devido à dinâmica da produção de vapor ser relativamente lenta, as variações na produção de potência são limitadas. Por exemplo, para aumentar a potência activa produzida num grupo térmico é necessário aumentar a pressão de vapor de água, logo é também necessário aumentar a queima de combustível para aumentar o vapor na caldeira. Consequentemente terá de existir um gradiente de temperatura na caldeira, que será tanto maior quanto a rampa de potência produzida. Grandes diferenças de temperatura na caldeira sujeitam-na a *stress* térmico, algo que pode ser entendido pela dilatação e contracção do material da caldeira a cada aumento ou diminuição de potência produzida. Se o limite de variação de temperatura não for respeitado, o material poderá ser danificado, o que resultará num aumento de custo de manutenção. Geralmente, o aumento de temperatura na caldeira está limitado a 93°C/h , ou no máximo a 204°C/h [8].

Nesse sentido são definidas rampas de incremento e decremento num grupo gerador que representam a variação máxima de potência num dado intervalo de tempo, tanto para redução como para aumento de produção. Habitualmente são representados por UR_i e DR_i , rampa de subida e de descida, respectivamente. Isto leva a que grupos de produção térmica tenham habitualmente rampas mais reduzidas, enquanto grupos hídricos têm elevada flexibilidade na variação da sua potência.

As rampas acabam por ser uma limitação à operação de centrais térmicas. Idealmente, toda a potência térmica instalada poderia entrar instantaneamente na rede, em caso de contingência. Na realidade, a operação de um SEE tem de levar em conta os vários tempos de arranque das térmicas (*hot start*, *warm start*, *cold start*) e rampas. Em regra, grupos com rampas mais pequenas, *i.e.*, menos flexíveis, são colocados a operar na base do diagrama de carga, enquanto os grupos mais flexíveis acompanham as pequenas variações na carga.

$$\begin{aligned} P_i - P_i^0 &\leq UR_i \\ P_i^0 - P_i &\leq DR_i \end{aligned} \quad (7)$$

Em (7) [9] está representada a formulação desta restrição, sendo UR_i a rampa de subida, DR_i a rampa de descida, P_i a potência actual e P_i^0 a potência no intervalo de tempo anterior. A expressão (7) em conjunto com (4), resulta na expressão (8) [10].

$$\max(P_{i,min}, P_i^0 - DR_i) \leq P_i \leq \min(P_{i,max}, P_i^0 + UR_i) \quad (8)$$

2.3.5. Reserva girante

O PDE deve ter em conta anomalias no funcionamento da rede. O objectivo mais importante da rede é conseguir garantir a continuidade de serviço, mesmo em caso de contingência, que pode resultar de um aumento inesperado da carga, avaria de subestação, mau funcionamento de um grupo, linha de transporte cortada, entre outras causas. Assim, reserva girante é uma estratégia utilizada no SEE, que tenta minimizar o efeito das contingências [11]. Esta estratégia requer que a todo o momento exista uma reserva na potência produzida por cada central. Isto é, um dado grupo produzirá no máximo a $1-\alpha\%$ de capacidade, para assim poder contribuir os restantes $\alpha\%$ em caso de contingência. Diferentes redes eléctricas adoptam diferentes valores de α , dadas as características técnicas de cada uma. Em (9) é indicada a expressão matemática desta restrição.

$$\sum_{i=1}^N [\min(P_i^{max} - P_i, UR_i)] \geq SR_i \quad (9)$$

2.4. Formulação do problema

A formulação completa do problema do despacho económico está indicada nas expressões (10) a (16).

$$\min F_T \quad (10)$$

s.a

$$\sum_{i=1}^N P_i = P_C + P_{perdas} \quad (11)$$

$$P_{i,t} \in \begin{cases} P_i^{min} \leq P_{i,t} \leq P_{i,1}^{LBj} \\ P_{i,j-1}^{UBj} \leq P_{i,t} \leq P_{i,j}^{LBj}, \quad j = 2,3, \dots, NP_i \\ P_{i, NP_i}^{UBj} \leq P_{i,t} \leq P_i^{max} \end{cases} \quad (12)$$

$$P_i - P_i^0 \leq UR_i \quad (13)$$

$$|P_i - P_i^0| \geq DR_i \quad (14)$$

$$P_i \leq (1 - \alpha) \cdot P_{max} \quad (15)$$

$$\sum_{j=1}^T \sum_{i=1}^N [\min(P_i^{max} - P_{i,t}, UR_i)] \geq SR_i \quad (16)$$

No caso de ter *multi-fuels* a função de custo tomará a forma de (3).

2.5. Trânsito de potências e perdas nas linhas

Entende-se por trânsito de potências “a solução em estado estacionário de um SEE compreendendo os elementos que o compõem” [12]. Considera-se como componentes do SEE os grupos geradores, transformadores, barramentos, linhas de transporte, etc. O problema de trânsito de potências (PTP) torna-se útil no planeamento e exploração de SEE [13], pelo que para existir rigor na resolução do PDE, o PTP deve ser calculado, englobando-se desta forma no problema as perdas. A influência das perdas no PDE resulta numa variável acrescida, que depende da potência gerada e das características das linhas de transmissão e acaba por afectar a produção total.

Os métodos de cálculo do trânsito de potências podem ser divididos entre determinísticos e probabilísticos [14]. Por métodos determinísticos entendem-se aqueles

que requerem que sejam especificadas algumas variáveis de entrada do SEE, nomeadamente [12]:

Tabela 2-1 – Variáveis do problema

P_G - potência activa gerada	θ - Argumento das tensões
Q_G - potência reactiva gerada	P_C - potência activa consumida
V - amplitudes das tensões	Q_C - potência reactiva consumida

A resolução determinística pode ser encarada como incapaz de lidar com as incertezas presentes em problemas de trânsito de potências. As metodologias que têm em conta as incertezas visam solucionar estas últimas, sendo que este ramo do PTP subdivide-se em 2 modelos de resolução: probabilístico e difuso. O modelo probabilístico trata as variáveis de entrada e os resultados como distribuições de probabilidade. O modelo difuso toma em consideração a lógica *Fuzzy* e trata as variáveis como distribuições de possibilidade [14]. Esta dissertação apenas explora a resolução determinística do PTP.

2.5.1. Equações do trânsito de potências

O conceito de potência injectada num barramento é definido como a diferença entre potência produzida e potência consumida, como se pode ver em (17) de acordo com [13].

$$S_i = S_i^G - S_i^C \quad (17)$$

As potências injectadas por barramento i são formuladas por (18) e (19), segundo [12], onde G_{ij} é a condutância nodal e B_{ij} a susceptância nodal. A figura 2-4 representa as grandezas envolvidas no PTP.

$$P_i = P_{Gi} - P_{Ci} = \sum_{j=1}^n V_i V_j [G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j)], i = 1, \dots, n \quad (18)$$

$$Q_i = Q_{Gi} - Q_{Ci} = \sum_{j=1}^n V_i V_j [G_{ij} \sin(\theta_i - \theta_j) - B_{ij} \cos(\theta_i - \theta_j)], i = 1, \dots, n \quad (19)$$

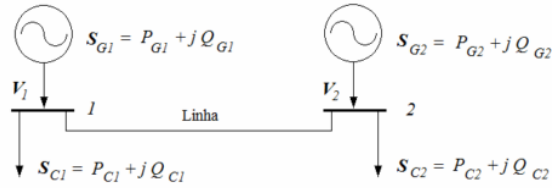


Figura 2-4 – Esquema unifilar de um sistema com 2 barramentos [12].

O trânsito de potências por linha é dado por (20) e (21), onde P_{ji}^k - trânsito de potência activa do barramento j para o i pela linha k e a mesma lógica para equação seguinte [12].

$$P_{ji}^k = G_k V_j^2 - V_i V_j [G_k \cos(\theta_j - \theta_i) + B_k \sin(\theta_j - \theta_i)] \quad (20)$$

$$Q_{ji}^k = -(B_k + B'_k) V_j^2 - V_i V_j [G_k \sin(\theta_j - \theta_i) - B_k \cos(\theta_j - \theta_i)] \quad (21)$$

E assim, as perdas nos ramos são obtidas pela soma de potências que cruzam o mesmo ramo em sentidos opostos.

$$P_L = (V_i^2 + V_j^2 - 2V_i V_j \cos(\theta_i - \theta_j)) G_k \quad (22)$$

$$Q_L = -(B_k + B'_k)(V_i^2 + V_j^2) + 2B_k V_i V_j \cos(\theta_i - \theta_j) \quad (23)$$

O PTP é por si só um problema não-linear, logo, com mais do que uma solução. Nesse sentido é difícil a resolução analítica simbólica, sendo na prática utilizados métodos numéricos, que reduzem o tempo para obter uma solução e toleram que esta esteja dentro de um certo erro.

2.5.1.1. Tipos de barramentos

Três tipos de barramentos são considerados num SEE: PV, PQ e de balanço. Os barramentos PQ representam habitualmente os barramentos de carga; os barramentos PV representam os barramentos de geração e o barramento de balanço representa o barramento de equilíbrio, *i.e.*, a sua geração acompanha a carga. Na tabela 2-2 está representada a relação entre os diferentes tipos de barramento e respectivas variáveis num PTP conhecidas, impostas e calculadas no PTP.

Tabela 2-2 – Tipos de barramentos [12]

Tipo de barramento	Variáveis conhecidas		Variáveis impostas		Variáveis calculadas	
Balanço	P_C	Q_C	θ	V	P_G	Q_G
PQ	P_C	Q_C	P_G	Q_G	θ	V
PV	P_C	Q_C	P_G	V	θ	Q_G

2.5.1.2. Resolução de trânsito de potências

A resolução do trânsito passa por resolver as equações (17) a (23) em função das variáveis desconhecidas e impondo valores às restantes. Estes cálculos são habitualmente realizados em p.u. Diferentes tipos de barramentos têm diferentes variáveis impostas, como é indicado na tabela 2-2. Em seguida, a resolução do PTP passa por inserir os valores das tensões, num processo iterativo até que as potências especificadas sejam iguais às calculadas ou estejam dentro de um intervalo aceitável tal como formulado em (24) e (25) [14].

$$\Delta P_i = |P_i^{esp} - P_i^{cal}| \leq 1e^{-4} \quad (24)$$

$$\Delta Q_i = |Q_i^{esp} - Q_i^{cal}| \leq 1e^{-4} \quad (25)$$

Os métodos numéricos (*Gauss-Seidel*, *Newton-Raphson* e *Fast Decoupled Power Flow*) aplicados na resolução deste problema são extensamente divulgados na literatura, [12]–[15] logo não se toma por vantajoso expor analiticamente cada método mais uma vez.

2.5.1.3. Resolução de trânsito de potências pelo modelo DC

Este modelo simplifica as equações do PTP com os seguintes pressupostos, de acordo com [13]:

- Componente resistiva das linhas desprezada
- Admitância transversal desprezada
- Tensões próximas do nominal
- Desfasamentos pequenos entre barramentos contíguos

As equações do modelo DC são obtidas através do desenvolvimento do modelo nodal.

$$P_i = \sum_{k \neq i} P_{ik} = \sum_{k \neq i} \frac{\theta_i - \theta_k}{X_{ik}}, \text{ para } i = 1 \dots n \quad (26)$$

$$P_i = \theta_i \sum_{k \neq i} \frac{1}{X_{ik}} + \theta_k \sum_{k \neq i} -\frac{1}{X_{ik}} \quad (27)$$

$$B_{ik} = -\frac{1}{X_{ik}}, i \neq k \quad (28)$$

$$P_i = \sum_{k \neq i} B_{ik} \theta_k, \text{ para } i = 1 \dots n \quad (29)$$

Que matricialmente aparecem na seguinte forma:

$$[P_i] = [B]'[\theta] \quad (30)$$

Aplicando a regra de *Cramer* para resolver sistemas de matrizes, o determinante da matriz dos coeficientes $[B]$ terá de ser não nulo, para que o sistema tenha uma solução. Sabendo que o determinante de $[B]$ é nulo, a expressão (30) é indeterminada. A forma de contornar esta indeterminação passa por assumir uma origem das fases, θ_{ref} , e eliminar o barramento de referência da expressão (30) [13]. Em seguida, a expressão (30) é calculada em função das fases das tensões, com as quais são depois calculados os valores de potência nas linhas, através de (31).

$$P_{ik} = \frac{\theta_i - \theta_k}{X_{ik}} \quad (31)$$

Estas simplificações transformam as equações não-lineares em lineares. Desta forma deixa de haver a necessidade de aplicar um método iterativo na resolução do problema e assim o mesmo poderá ser resolvido mais rapidamente.

Existe uma outra simplificação que permite calcular as perdas no SEE apenas através das potências geradas e da matriz \mathbf{B} . É, a par do modelo DC, amplamente divulgada literatura. Esta permite o cálculo das perdas no trânsito de potências de forma mais célere, embora sujeita a simplificações, através da equação (32), [16].

$$P_{perdas} = P^T [B] P + B_0^T P + B_{00} \quad (32)$$

O modelo DC tem duas limitações, nomeadamente o facto de os módulos das tensões serem assumidos e de não ter em conta o trânsito de potência reactiva. No entanto, é

também o método aplicado na maioria da bibliografia e assim, de forma a ser possível validar os resultados obtidos, será também o método utilizado nesta dissertação.

2.5.2. Definição matemática de função convexa

Uma função $f: C \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ num conjunto convexo é convexa, se

$$\forall x, y \in C, f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \text{ para } 0 \leq \lambda \leq 1 \quad (33)$$

Para que um conjunto $C \subseteq \mathbb{R}^n$ seja convexo, $x_1, x_2 \in C, [x_1, x_2] \subseteq C$ [17]. Logo, para que um conjunto seja convexo, quaisquer dois pontos terão de poder ser ligados por uma linha recta, cujos seus pontos também terão de se encontrar dentro do conjunto. Existem simplificações para analisar a convexidade de uma função, tal como é exposto em 2.5.2.1. Na figura 2-5 está exemplificado um conjunto não convexo.

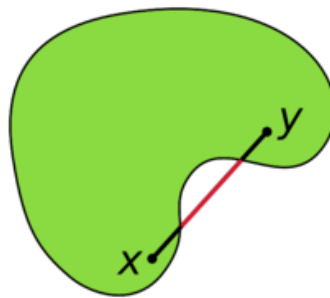


Figura 2-5 – Conjunto não convexo [18].

Uma função convexa poderá ser, por exemplo, qualquer função linear ou polinomial de segunda ordem, que respeite a expressão (33). A figura 2-6 representa a definição de função convexa, descrita em (33).

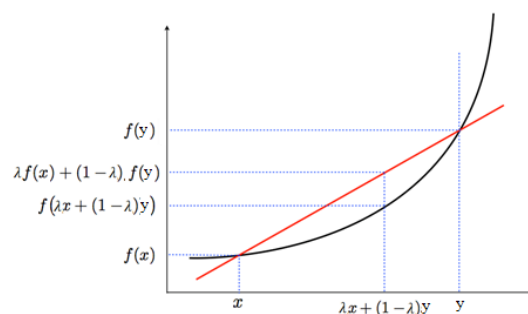


Figura 2-6 – Definição geométrica de função convexa.

2.5.2.1. Condições de convexidade

A condição de convexidade de primeira ordem diz que, se a função f for diferenciável no seu domínio f , então a mesma é convexa sse o domínio de f for convexo e a expressão (34) se cumprir [19]. Na figura 2-7 encontra-se representada a expressão (34).

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad (34)$$

A condição de convexidade de segunda ordem é cumprida quando a Hessiana de f , $\nabla^2 f(x)$ exista no domínio de f . Se tal se observar, f é uma função convexa sse o domínio de f for convexo e a sua Hessiana seja semidefinida positiva [19].

$$\nabla^2 f(x) \geq 0 \quad (35)$$

Esta condição garante que a derivada da função f é não decrescente e que a curvatura de f é positiva.

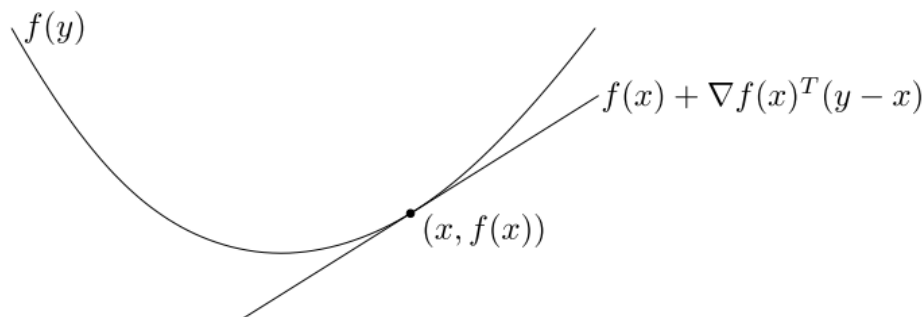


Figura 2-7 – Representação da expressão (34) [19].

2.6. Não convexidade das funções de custo

Geometricamente, uma função f é convexa quando a linha que une quaisquer dois pontos se encontra acima da própria função f , tal como se pode ver na figura 2-8.

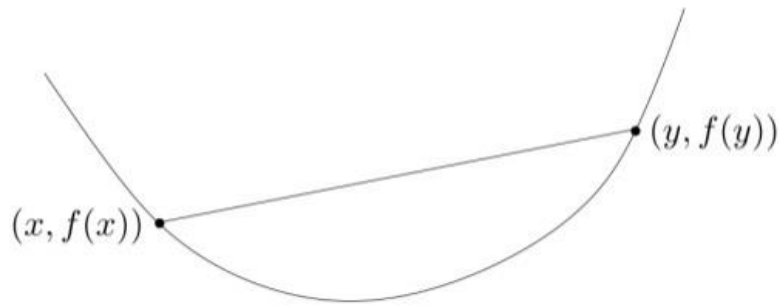


Figura 2-8 – Representação de uma função convexa [19].

“Uma propriedade fundamental de problemas de optimização convexos é que qualquer mínimo local é igualmente o óptimo global” [19]. Assim, se a função de custo for convexa, as restrições de desigualdade forem convexas e as de igualdade forem quase-convexas [19], então um método de optimização baseado no método de gradiente consegue analiticamente encontrar o mínimo global.

Funções de custo não convexas são um entrave à aplicação de métodos convencionais em PDE e, embora existam técnicas para contornar esta dificuldade (aplicação da regressão à função de custo ou seccioná-la por troços, por exemplo), a solução final poderá não ser suficientemente precisa. O efeito das válvulas de vapor é um dos responsáveis por tornar as funções de custo quadráticas e convexas em não convexas.

2.6.1. Efeito das válvulas de vapor

A geração de energia eléctrica recorrendo a centrais térmicas recorre a vapor de água pressurizado para girar uma turbina. Esta injeção de vapor de água pode ser utilizada para controlar a potência gerada através da regulação da abertura e/ou fecho de válvulas. A abertura das válvulas poderá causar um efeito *ripple* no funcionamento da turbina [20] e, conseqüentemente, influenciar a função de custo.

Ao abrir as válvulas, crescem as perdas por estrangulamento e assim a taxa de calor (*heat rate*) aumenta. Em termodinâmica, o aumento de perdas pode ser explicado pelo efeito *Joule-Thomson*. Este descreve o comportamento de um gás pressurizado ao passar por uma válvula ou qualquer mecanismo de controlo à sua passagem, com a

entalpia¹ constante, *i.e.*, o sistema é estanque, não existindo trocas energéticas com o exterior. O vapor de água comprimido ao atravessar a válvula passará para um ambiente menos pressurizado e, segundo a *Lei dos gases ideais*², irá expandir. Esta expansão arrefecerá o gás e reduzirá a sua pressão, pelo que será preciso mais combustível para recuperar a pressão (com conseqüente aumento do custo de produção), aumentando a taxa de calor (*heat rate*). Isto tem como conseqüência um aumento do custo de produção da energia eléctrica como se pode verificar na figura 2-10, ou seja, para gerar a mesma unidade de energia eléctrica, será necessário mais combustível [21]. Antes da abertura da válvula, a função de custo da figura 2-10 tende a estabilizar e o custo por unidade de tempo de geração fica estável, quase constante. Esta estabilização é o resultado de a câmara que liga a válvula à turbina estar já a chegar ao seu limite, logo as pressões na câmara pré-válvula e pós-válvula aproximam-se e assim o efeito *Joule-Thomson* é diminuído, como é ilustrado na figura 2-9.

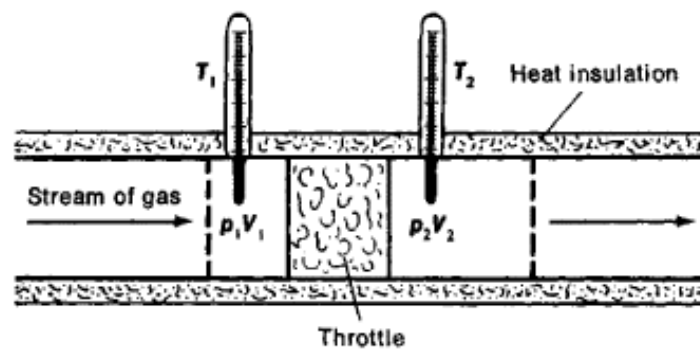


Figura 2-9 – Efeito *Joule-Thomson*.

O efeito válvula de vapor é um dos responsáveis por afectar a função de custo e torná-la não convexa e não diferenciável em alguns pontos do seu domínio, como se vê na figura 2-10. Isto dificulta a aplicação de métodos de optimização convencionais, tais como o método do gradiente, visto estes necessitarem de funções diferenciáveis em todo o domínio.

Para se modelizar matematicamente o efeito das válvulas, a expressão (2) é modificada tomando a forma de (36) onde e_i e f_i são coeficientes constantes e $P_{i,min}$ o limite mínimo de potência para o grupo i .

¹ Energia existente num sistema termodinâmico capaz de ser extraída na forma de calor.

² A lei dos gases ideais é a equação de estado de um gás ideal hipotético.

$$F_i(P_i) = c_i P_i^2 + b_i P_i + a_i + |e_i \times \text{sen}[f_i \times (P_{i,\min} - P_i)]| \quad (36)$$

Na figura 2-10 está representada a evolução da expressão (36) (em azul e eixo à direita), bem como a sua derivada (em verde e eixo à esquerda) [22]. Verifica-se assim que funções de custo como a indicada na expressão (36) não são diferenciáveis em todo o seu domínio. Nesse sentido, não é possível aplicar directamente métodos baseados no gradiente.

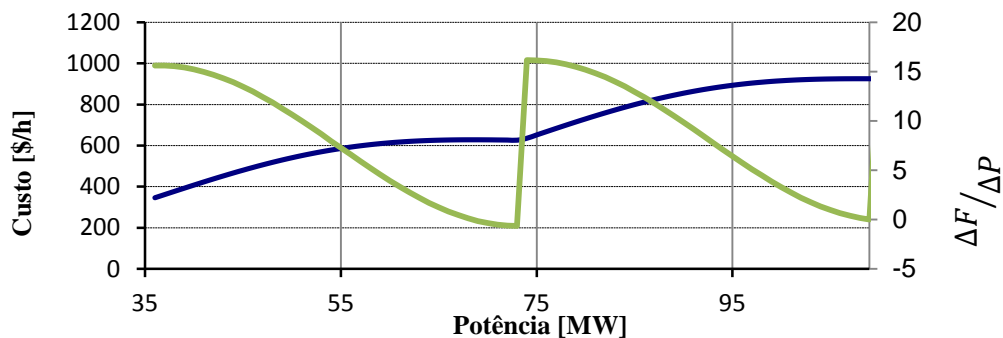


Figura 2-10 – Função de custo (azul) com legenda à esquerda e sua derivada (verde) com eixo à direita [22].

Uma das formas de ultrapassar esta não linearidade, para poder aplicar um método convencional, é seccionar a função em vários intervalos. Na prática isto resulta em dividir a função de custo inicial em n funções de custo para n intervalos. Num PDE com vários grupos e várias centrais, este método de superar a descontinuidade acaba por criar um problema com o aumento do número de cálculos necessários, requerendo uma capacidade de processamento considerável. Por exemplo, caso a função de custo de dois grupos com efeito de válvula seja seccionada em 3 intervalos (3 válvulas de admissão), existirão 9 combinações e necessariamente serão necessários 9 cálculos. A progressão combinatória é dada por $\prod_{i=1}^N C_1^s$, sendo s o número de válvulas de admissão por grupo e N o número de grupos. Caso o número de válvulas de admissão seja igual em todos os grupos, a expressão poderá ser simplificada para C_1^{sN} . Esta notação tem em conta a impossibilidade de um grupo operar em mais do que um estado, ou seja, embora um grupo com 3 válvulas seja tratado como 3 grupos separados, estes não podem ser combinados entre si. Assim, para um PDE de 3 centrais, com 9 grupos no total e cada grupo com 3 válvulas de admissão, o PDE teria de ser calculado como se o número de

grupos fosse $C_1^{3^9}$, resultando num PDE com 19 683 grupos. Na figura 2-11 está representada a evolução do número de combinações possíveis em função do número de grupos, assumindo 3 válvulas.

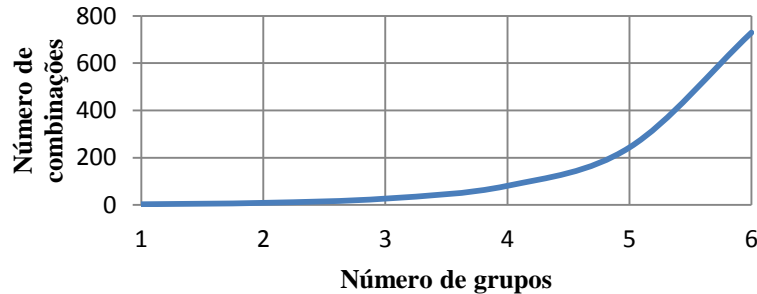


Figura 2-11 – Progressão combinatória do seccionamento de funções de custo.

Outro dos métodos passa por aproximar a função de custo através de uma regressão polinomial de primeira ou segunda ordem, tolerando o erro resultante. Na figura 2-12 é representada uma possível regressão polinomial de segunda ordem.

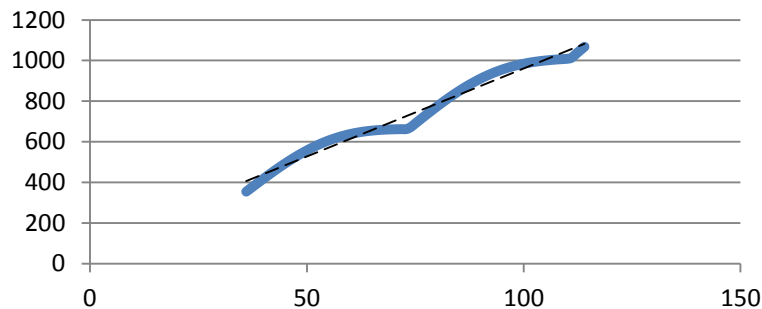


Figura 2-12 – Aproximação linear da função de custo com efeito de válvula.

Com esta aproximação é possível otimizar a função de custo resultante com métodos convencionais visto ser diferenciável. No entanto, o erro entre a função real e a aproximada pode resultar num acréscimo significativo do custo de operação.

Pelo exposto torna-se necessário explorar outros métodos para solucionar o problema de modo a diminuir o número de combinações de formulações do problema ou evitar erros que possam advir de algum tipo de simplificação. Nesse sentido, alguns métodos heurísticos são propostos com resultados bastante encorajadores.

Capítulo III

Visão global (*Overview*)

Resumo

Neste capítulo é apresentada uma visão global do problema de despacho económico e dos métodos convencionais e meta-heurísticas utilizados para o resolver.

3. Visão global (*Overview*)

3.1. Introdução

Diversos métodos foram e são empregues para resolver o PDE. Recentemente foram propostos métodos baseados em heurísticas, que poderão ser mais práticos face às funções de custo não convexas encontradas em PDE. Os métodos aplicados ditos tradicionais ou convencionais são o método do gradiente, lagrangeano, *lambda iteration*, programação dinâmica, método de Newton, programação linear e método do ponto interior (*interior point method*) entre outros [23]. No entanto, algumas funções de custo de grupos térmicos não são contínuas, nem convexas, nem diferenciáveis devido ao efeito das válvulas de vapor, multi-fuels ou ainda existência de zonas proibidas. Por este motivo, o PDE torna-se num problema não convexo com restrições, que não pode ser resolvido directamente por alguns dos métodos convencionais. Programação dinâmica pode resolver este tipo de problema, mas sofrerá com o número de dimensões e o tempo computacional requerido [22]. Por outro lado, os pacotes de optimização convencionais usados em SEE apenas suportam funções convexas, sendo as funções de custo não convexas manipuladas para passarem a convexas.

3.2. Breve evolução do problema de despacho económico

O problema de despacho económico gere diferentes objectivos, que poderão não ser independentes entre si. Basicamente, o PDE serve para garantir que a produção iguala a carga mais as perdas, do modo mais económico possível. Por outro lado, a mesma produção deverá ter em conta o seu impacto ambiental e reduzi-lo ao máximo. Apesar disso, a segurança de pessoas e equipamentos deverá sobrepor-se aos objectivos anteriores. A solução do PDE é assim influenciada por objectivos conflitantes que interagem entre si [24]. Este tem vindo a ser estudado desde o início do século XX, mais precisamente desde o princípio dos anos 20 desse século, segundo “*Optimal Power Dispatch – A Comprehensive Survey*” de Happ, que publicou em 1977 este registo histórico da evolução do problema de despacho. No princípio foi optimizado com recurso ao método de *base load*, que atribui a certos grupos uma potência constante independente das variações na carga. Grupos de operação menos eficiente serão

utilizados para acompanhar a diferença entre a potência constante produzida pelos grupos mais eficientes e a carga. Outro método, *best point loading*, coloca os grupos a produzir na potência com o menor *heat rate*. Os *heat rates* são comparados e a carga é satisfeita desde o mais eficiente (menor *heat rate*) até ao menos eficiente (maior *heat rate*) [24] [25]. O *incremental method* (método de incrementação) foi proposto por *Steinberg* e *Smith* em 1934 e mais tarde exposto no livro que escreveram em 1943 denominado *Economy Loading of Power Plants and Electric Systems*. Esta metodologia indica que o incremento de carga deve ser gerado pelo grupo com o menor custo incremental ou custo marginal.

3.3. Teorema de *Karush Kuhn-Tucker*

Este teorema é uma “condição necessária, mas não suficiente a menos que sejam verificadas certas condições de suficiência, para a identificação de um ponto óptimo de um problema de programação matemática” [17]. O teorema de *Karush Kuhn-Tucker* (KKT) diz que seja a função objectivo e suas funções restricionais continuamente diferenciáveis no ponto da solução óptima, χ^* , que satisfaz à condição de qualificação restricional, então se:

$$\chi^* \in G \quad (37)$$

$$\exists \lambda_j \geq 0 \text{ tal que } \lambda_j g_j(\chi^*) = 0 \text{ para } j=1, 2, \dots, m \quad (38)$$

$$\nabla f(\chi^*) + \sum_j^m \lambda_j \nabla g_j(\chi^*) = 0 \quad (39)$$

A solução diz-se ponto crítico ou de estacionaridade e não solução óptima. Para que seja óptima, a mesma deverá cumprir as condições de suficiência de KKT [17].

3.4. Métodos de optimização convencionais

Os métodos convencionais são métodos numéricos, que possibilitam a descoberta da solução exacta e precisa de determinado PDE, desde que não apresente efeito das válvulas de vapor, nem zonas proibidas, nem *multi-fuels*. Assim, para PDE sem estas restrições, poderá ser mais vantajoso aplicar um método convencional que uma meta-

heurística, pois a solução final desta pode não ser a óptima. De 3.4.1 a 3.5.8 são analisados alguns métodos de optimização convencionais.

3.4.1. Método do gradiente

A busca do óptimo pelo método do gradiente assume que é possível atingir o óptimo através de uma série de iterações, em que o gradiente, expressão (40), indica a inclinação da descida [26]. Portanto, é um conceito iterativo simples, que pode ser melhor observado pela expressão (41).

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \quad (40)$$

$$x_1 = x_0 - \nabla f \alpha \quad (41)$$

Em que x_1 é a nova solução, x_0 é a solução actual, ∇f é o gradiente da função f e α é um escalar que garante a convergência do método. O gradiente em si direcciona para o máximo numa função, daí ser passado ao simétrico na expressão (41).

3.4.2. Lambda iteration

Lambda iteration é um método iterativo de resolução de PDE, “sendo a sua convergência bastante rápida neste tipo de problema” [26], [27]. Este método analisa o custo marginal dos grupos e, calculando a potência gerada respectiva, compara-os com a potência da carga. Graficamente, como é demonstrado na figura 3-1, o *lambda iteration* pode ser executado traçando uma recta num gráfico que englobe todas as derivadas das funções de custo, *i.e.*, funções de custo marginal. Esta recta corresponde a um custo marginal, que por sua vez ao cruzar numa dada função irá corresponder a uma certa potência gerada. Manualmente, uma recta é traçada onde se considera ser o custo marginal; de seguida é deslocada para cima e para baixo, até que a soma de todas as potências geradas resulte na potência da carga, atendendo às restrições do PDE em questão.

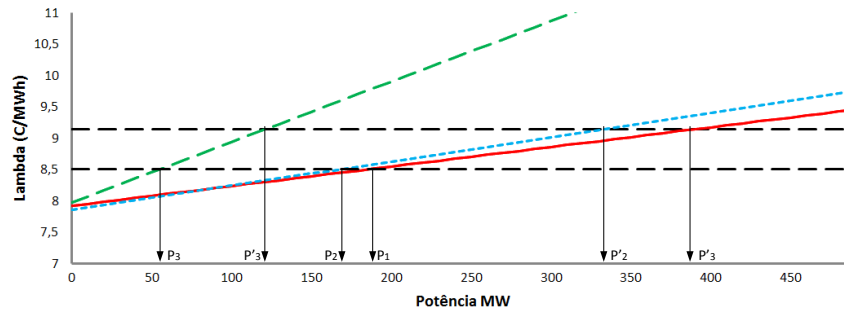


Figura 3-1 – Exemplo de *lambda iteration*.

A demonstração que se segue do *lambda iteration* usa dados do caso estudado em [26], que aqui se resumem para mais fácil verificação:

Tabela 3-1 – Funções de custo do caso estudado em [24]

P_{Carga}	850 [MW]
Função de custo 1	$F_1(P_1) = 561 + 7,92 \cdot P_1 + 0,001562P_1^2$
Função de custo 2	$F_2(P_2) = 310 + 7,85 \cdot P_2 + 0,00194P_2^2$
Função de custo 3	$F_3(P_3) = 78 + 7,97 \cdot P_3 + 0,00482P_3^2$

Nota: Os custos marginais são representados por λ s.

Na tabela 3-2 estão indicados os valores resultantes do *lambda iteration* ao longo de 7 iterações.

Tabela 3-2 – Lambda iteration (iterações e valores de potência)

Iteração	1 ^a	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	7 ^a
Lambda	0	7,335	7,983	8,630	9,045	9,128	9,144
P_1 [MW]	150	150	150	227,1	360,3	386,7	391,9
P_2 [MW]	100	100	100	200,9	308,1	329,4	333,6
P_3 [MW]	50	50	50	68,4	111,6	120,1	121,8
P_{Total} [MW]	300	300	300	496,5	780	836,1	847,2

Analisando os resultados, verifica-se que ao fim de 7 iterações o valor de potência produzida já se aproxima bastante da carga (850 MW), o que demonstra a rapidez de convergência do método.

Dois critérios de paragem são habitualmente empregues no *lambda iteration*, sendo que um compara o resultado do lambda com a tolerância introduzida e o outro limita o número de iterações [26], [27]. Para o primeiro critério é necessário introduzir um valor de tolerância considerado bom (tolerância grande e o resultado irá ter um erro grande, tolerância pequena, são necessárias mais iterações), a partir do qual é feita a comparação entre $P_{Total}^{iteração\ i}$ e P_{carga} , sendo que caso a razão $\frac{P_{Total}^{iteração\ i}}{P_{carga}} \times 100$ não esteja dentro de do intervalo de tolerância $[-\varepsilon, \varepsilon]\%$, o *lambda iteration* não converge.

3.5. Métodos não convencionais

3.5.1. Meta-heurísticas

Dadas as restrições encontradas em funções de custo não-convexas, diversas meta-heurísticas³ foram propostas para a resolução do PDE, tais como algoritmos genéticos (GA), *Simulated annealing* (SA), *Taboo search* (TS), *Evolutionary programming* (EP), *Evolutionary strategies* (ES), *Particle swarm optimization* (PSO), *Bacteria foraging optimization* (BFO), *Ant colony optimization* (ACO), Redes neuronais (NN), *Artificial bee colony* (ABC) e híbridos [28]. A partir dos algoritmos básicos, foram propostas melhorias e híbridos, tais como o *Improved Taboo Search* (ITS) [22], *Fast Evolutionary Programming* (FEP) e *Improved Evolutionary Programming* (IEP) [29], *Improved Particle Swarm Optimization* (IPSO) [28] e híbridos como o PSO com EP [30] e *Fast Evolutionary Programming with Swarm Direction* [7]. Muitas destas meta-heurísticas têm em comum o facto de se basearem em inteligência de enxames, um ramo da inteligência artificial fundamentado no comportamento animal.

3.5.2. Origem do estudo da Inteligência de Enxames

Em 1989, G. Beni e J. Wang cunharam a expressão *Swarm Intelligence*, no seu trabalho em *Robotic Swarm* [31]. O estudo do reino animal aprofundou-se no estudo comportamental e possibilitou o melhor entendimento de como cooperam indivíduos dentro de um grupo e quais os mecanismos usados para controlar o enxame e

³ Uma meta-heurística é um método fundamentado numa heurística.

condicionar o indivíduo, tais como a estigmergia. Por enxame, pode entender-se manada, alcateia, bando, colónia, entre outras designações conforme o animal e, a partir daqui, qualquer referência a um grupo de agentes passa a ser feita por enxame, *e.g.*, um enxame de pássaros. Os 5 princípios da inteligência de enxame segundo [32], são:

- Proximidade – A população deve ser capaz de executar tarefas simples.
- Qualidade – A população deve ser capaz de responder a mudanças no ambiente.
- Diversidade – A população não deve focar-se excessivamente.
- Estabilidade – A população não deve alterar o seu comportamento a cada alteração no ambiente.
- Adaptabilidade – A população deve ser capaz de alterar o seu comportamento quando valha o esforço computacional.

3.5.3. Comportamento de enxame

Um animal, quando agrupado com mais membros da sua espécie, poderá exibir um comportamento diferente de quando isolado. Um bando de pássaros em vôo tem um comportamento de enxame, isto é, os pássaros seus constituintes agem em prol do grupo, em vez de si mesmos, não respeitando um comando centralizado. Este comportamento pode ser observado em diversos animais, desde os insectos (formigas, gafanhotos, térmitas, abelhas, etc), até aos mamíferos (morcegos, gnus, ovelhas, humanos, etc). Assim, uma definição generalista dirá que este comportamento se dá sempre que um grupo de indivíduos se agrupa e age socialmente, tentando atingir um objectivo e que os indivíduos num enxame possuem sensibilidade local, efectuando acções simples e desconhecendo o estado global do enxame, bem como o seu objectivo comum. Este objectivo comum é quase tão diverso quanto o número de espécies que se comportam em enxame. As abelhas actuam conjuntamente na busca de alimento, na protecção da sua colmeia e seu desenvolvimento; as andorinhas voam em formação para assim reduzirem o atrito aerodinâmico, algo que também pode ser observado em pilotos de aviões caça e ciclistas; manadas de gnus juntam-se em números avolumados para protecção contra predadores, acasalamento e até calor.

Sobre o comportamento de enxames, estigmergia é um termo usado para o explicar. Vem do grego, com a junção das palavras gregas *στίγμα* (estigma, sinal) e *ἔργον*

(projecto, acção). É um termo cunhado pelo biólogo francês, Pierre-Paul Grassé, em 1959 para caracterizar o comportamento de térmitas. Basicamente, estigmergia é o comportamento que leva um agente a reagir à acção de outro ou de si próprio. Trata-se de um condicionalismo que estimula certa resposta a certa acção, reforçando a acção inicial e permitindo assim uma acção complexa realizada por agentes simples, sem inteligência. O exemplo habitualmente dado para a estigmergia é o das formigas representadas na figura 3-2. Sozinhas, as formigas são agentes simples, sem inteligência, mas em enxame demonstram a inteligência de perceber qual o caminho mais curto entre o formigueiro e a comida. As formigas libertam químicos, feromonas, no seu rasto, que qualquer formiga no seu encalço sentirá e decidirá se segue o caminho das feromonas ou se diverge e cria um caminho próprio. A probabilidade de as formigas seguirem o caminho já trilhado é maior.

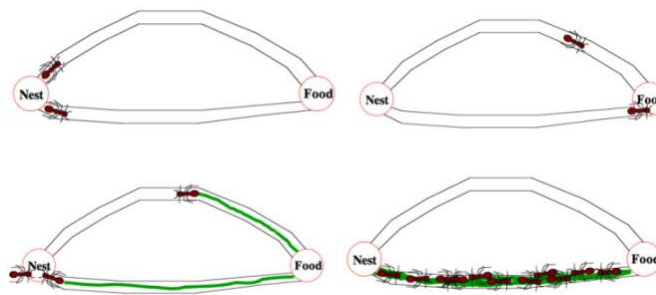


Figura 3-2 – Estigmergia em formigas

Na figura 3-2, duas formigas partem em busca de alimento, tendo a formiga de baixo demorado menos tempo até regressar ao formigueiro por o seu caminho ser mais curto. Este caminho foi percorrido duas vezes (ida e volta), logo terá maior intensidade de feromonas que o caminho de cima. Esta maior intensidade levará uma maior quantidade de formigas a escolherem o caminho de baixo, enquanto a intensidade do de cima tenderá a desaparecer, pois as feromonas evaporam.

3.5.4. Particle Swarm Optimization

Em 1995 foi apresentado um artigo da autoria de J. Kennedy e R. Eberhart, no qual foi introduzido um método de optimização para funções não lineares, denominado *Particle Swarm Optimization* (PSO) [33].

O PSO é baseado no comportamento de um bando de pássaros em busca de alimento. O bando aumenta a probabilidade de encontrar alimento quando coopera na sua procura, interagindo entre si com a informação que cada pássaro retira do meio ambiente. Cada pássaro poderá decidir entre tomar um rumo egoísta, em que valoriza o seu próprio conhecimento e segue para a melhor posição que já encontrou até ao momento ou decidir tomar um rumo social, em que segue a direcção do melhor pássaro do bando (considera-se como melhor pássaro aquele que mais perto se encontra da posição da comida). Esta é a base biológica em que se fundamenta a formulação matemática do PSO, tal como foi inicialmente enunciada em [33].

Na implementação do algoritmo em problemas de optimização, o meio ambiente representa o espaço de busca e cada pássaro é considerado uma partícula, que representa uma possível solução para o problema considerado. Cada partícula interage com todas as outras na busca por uma solução melhor através de três parâmetros, nomeadamente a sua inércia e comportamentos cognitivo e de sociabilização. O comportamento cognitivo dá um peso ao melhor valor de *fitness* de cada partícula obtida até ao momento e o de sociabilização dá um peso à partícula que alcançou o melhor valor de *fitness* do enxame até ao momento. A inércia estabelece limites à velocidade de cada partícula [34]. Cada pássaro tem a capacidade de avaliar a sua posição, devolvendo um valor de *fitness* para a sua posição. Este *fitness* permite que as partículas sejam comparadas entre si e determinem para onde se devem dirigir. Em suma, cada partícula é definida pela sua posição, pela sua velocidade e pelo seu valor de *fitness*. A inércia é calculada por (42), a velocidade de cada partícula por (43) e a nova posição de cada partícula por (44) [22].

$$\omega = \omega_{m\acute{a}x} - \frac{\omega_{m\acute{a}x} - \omega_{m\acute{i}n}}{it\acute{e}ra\acute{c}\tilde{o}es_{m\acute{a}x\acute{i}m\acute{o}}} \times it\acute{e}ra\acute{c}\tilde{o}es_{actual} \quad (42)$$

$$v_i(t + 1) = \omega \times v_i(t) + c_1 \times (x_{best} - x_i(t)) + c_2 \times (g_{best} - x_i(t)) \quad (43)$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (44)$$

Em (42), ω é a inércia, $\omega_{máx}$ a inércia máxima e ω_{min} a inércia mínima, sendo estes dois últimos definidos *a priori*. Em (43), v_i é a velocidade de cada partícula i , c_1 é o coeficiente cognitivo, c_2 é o coeficiente social, x_{best} representa o melhor valor atingido por cada partícula até ao momento e g_{best} o melhor valor atingido pelo enxame até ao momento. Em (44), $x_i(t + 1)$ é a nova posição e $x_i(t)$ é a posição actual. A equação (43) é ilustrada na figura 3-3, onde está representado esquematicamente o processo que define a nova posição de cada partícula i .

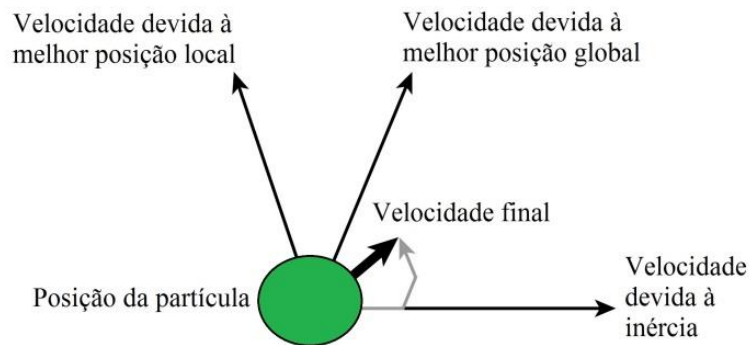


Figura 3-3 – Velocidade imposta a uma dada partícula.

Os coeficientes c_1 e c_2 são parâmetros positivos e constantes. Na implementação original do PSO, apenas um parâmetro, c , era aplicado, sendo então chamado de factor de aprendizagem (*learning factor*) ou constante de aceleração (*acceleration constant*) e geralmente igual a dois, *i.e.*, $c_1 = c_2 = 2$. Para controlar o comportamento do enxame, versões recentes do PSO têm vindo a ajustar estes parâmetros no intervalo [0,4]. Ajustando estes parâmetros, os seguintes comportamentos são obtidos [31]:

- Valores elevados de c_1 e c_2 habilitam o enxame a obter novas partículas em regiões distantes, o que garante uma melhor busca global, mas por outro lado poderão levar o enxame a divergir.
- Valores reduzidos de c_1 e c_2 limitam o movimento do enxame, podendo assim restringir a busca global a uma busca mais local. Esta opção poderá ser vantajosa quando se pretende uma busca localizada, mais refinada, com poucas variações à volta dos melhores valores obtidos.
- Se o coeficiente cognitivo for maior que o de sociabilização, o enxame favorecerá as melhores experiências locais de cada partícula.

- Se o coeficiente de sociabilização for maior que o cognitivo, o enxame favorecerá a melhor experiência global do enxame.

Na figura 3-4 está representada a evolução das partículas do enxame em função do número de iterações para a função *De Jong* (função 1, [35]) para duas dimensões, 100 partículas, $c_1 = c_2 = 2$, $\omega_{máx} = 0,9$ e $\omega_{min} = 0,3$. A função *De Jong* tem o seu valor mínimo no ponto $f(x_1, x_2) = (0,0)$, verificando-se claramente a tendência do deslocamento das partículas para as proximidades desse valor com o evoluir do número de iterações.

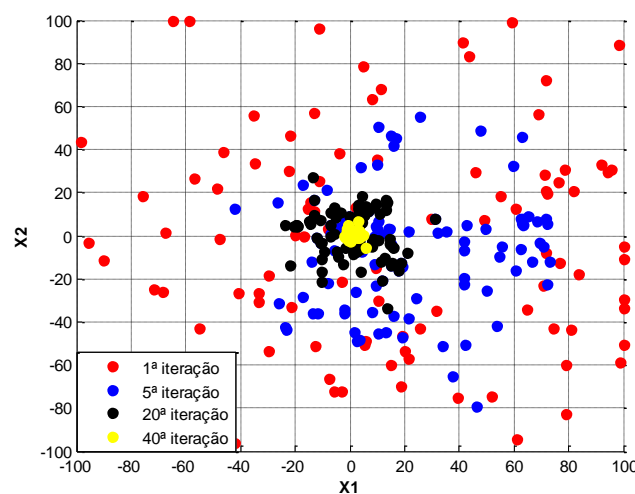


Figura 3-4 – Evolução do enxame do PSO na busca pela melhor solução.

Apesar da sua simplicidade, o PSO é um algoritmo rápido a correr completamente e eficaz, sendo utilizado nos mais variados problemas. No anexo 1 pode ser encontrado um possível pseudo-código para a implementação do algoritmo [22], [31] e [36].

3.5.5. *Bee Colony Optimization*

O *Bee Colony Optimization* (BCO) é caracterizado por reproduzir o comportamento das abelhas na busca por pólen nas flores. A simulação de colónia de abelhas pode ser encontrada em [34], em que é denominada de *Artificial Bee Colony*.

As abelhas batedoras (*scouts*) partem da colmeia em busca de flores, com a maior quantidade de pólen. No regresso à colmeia, as abelhas comunicarão com as restantes

abelhas de modo a cativá-las para as acompanhar. Esta comunicação indica qualidade da flor, distância relativamente à colmeia e direcção relativamente ao sol e é conseguida através de uma dança executada pela abelha. A direcção (parecida com uma trajectória em forma de oito) em que a abelha dança informa qual a posição da flor relativamente ao sol; a distância da flor relativamente à colmeia é codificada na dança pelo comprimento que a abelha percorre; a qualidade da flor é transmitida pela frequência com que a abelha treme, *e.g.*, quanto maior a frequência, mais excitada a abelha se demonstra estar sobre a qualidade da flor. Uma representação esquemática da dança está ilustrada na figura 3-5.



Figura 3-5 – Representação da dança da abelha para comunicar.

Na implementação do BCO, são criados três diferentes tipos de abelhas: batedoras (*scouts*), seguidoras (*followers*) e escudeiras (*onlookers*). Na primeira fase, as *scouts* partem da colmeia em busca de flores num percurso aleatório. De seguida, regressam à colmeia e apresentam a sua melhor flor encontrada às *followers*. Estas acompanharão com maior probabilidade a *scout* com melhor *fitness*. Se nenhuma melhoria no valor de *fitness* for encontrada pela *scout* num determinado intervalo de iterações, esta abandona a flor e passa a *onlooker*, que irá procurar outra flor de forma aleatória [31], [34].

A equação (45) demonstra como é calculada a probabilidade de cada flor encontrada pelas *scouts* ser explorada pelas *followers*. Na equação (46) está representada a forma como as *followers* circulam à volta da posição encontrada pela *scout*.

$$P_i = \frac{fit(x_i)}{\sum_{n=1}^{SN} fit(x_n)} \quad (45)$$

$$v_{ij} = x_{ij} + \varphi_{ij} \times (x_{ij} - x_{kj}) \quad (46)$$

Em (45), P_i é a probabilidade de cada flor ser escolhida pelas *followers*, $fit(x_i)$ é o valor de *fitness* da posição x_i e SN é o número de alimentos, *i.e.*, número de *scouts*. Em (46), v_{ij} é a posição da *follower* na vizinhança da flor encontrada pela *scout*, x_{ij} é a posição da flor encontrada pela *scout*, φ_{ij} é um valor aleatório no intervalo $[-1, 1]$ e x_{kj} é a posição de outra *scout*, com a qual é feita a comparação.

Na figura 3-6 está representada a evolução das partículas (abelhas) do enxame em função do número de iterações para a função *De Jong* para duas dimensões, 100 partículas.

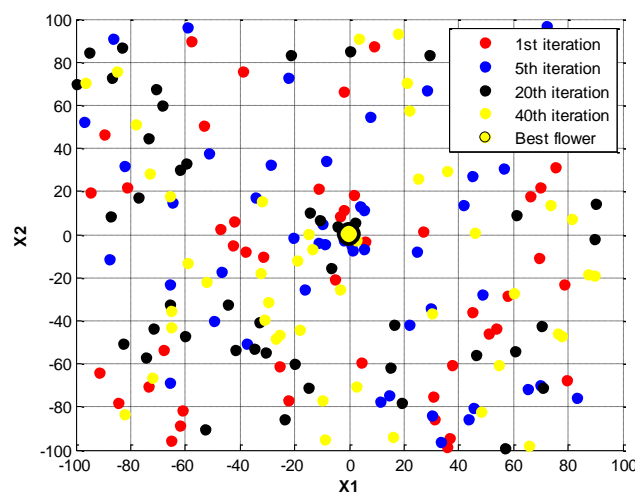


Figura 3-6 – Evolução do enxame do BCO na busca pela melhor solução.

Na figura 3-6 pode ver-se que o BCO e o PSO convergem de forma diferente. Enquanto que no PSO as partículas vão convergindo para o mínimo global quando este é encontrado, no BCO esta convergência não acontece. Na 1ª iteração já o algoritmo encontrou a proximidade do óptimo em (0,0), mas as partículas continuam a procurar o espaço de busca inteiro (como se pode ver na 40ª iteração, em que há partículas nos extremos) de forma aparentemente aleatória, embora estejam limitadas pela posição das restantes abelhas. Tal característica resulta num algoritmo com uma elevada capacidade de busca e facilidade em não ficar preso em mínimos locais. No reverso da medalha, esta característica também atrasa o algoritmo, tendo o BCO uma velocidade de convergência baixa.

No anexo 2 pode ser encontrado um possível pseudo-código para a implementação do algoritmo [34].

3.5.6. Cockroach Swarm Optimization

O *Cockroach Swarm Optimization* (CSO) é inspirado no comportamento social das baratas. Diversos algoritmos foram criados com base no comportamento das baratas, por exemplo o *Cockroach Swarm Optimization* e o *Roach Infestation Optimization* (RIO) que podem ser analisados em [37] e [38], respectivamente.

Um estudo do comportamento das baratas em [39], conclui que “*a decisão colectiva emerge das interacções entre indivíduos iguais, que inicialmente possuem pouca informação sobre o seu ambiente*”. As baratas exibem um comportamento de perseguição (*chase swarming*), pelo qual as baratas deixam um trilho de matéria fecal no seu caminho, para que outras baratas possam seguir. Para este comportamento também contribui a emissão de feromonas para o ar. Outro comportamento, dispersão (*dispersing*), permite às baratas dispersarem-se repentinamente na presença de perigo ou quando há falta de recursos no local. Por último, os seus enxames apresentam um comportamento de implacabilidade (*ruthless*), *i.e.*, quando há falta de alimento, as baratas eliminar-se-ão mutuamente [37], [40].

A formulação matemática do CSO pretende simular os comportamentos dos enxames de baratas, através das três características já apontadas. Um enxame de baratas é inicializado de forma aleatória e as suas partículas são avaliadas, sendo o melhor indivíduo, *i.e.*, o melhor valor de *fitness* associado a uma dada posição, no seu campo visual. Caso exista uma melhor barata no seu campo visual, ela irá para lá segundo (47), formando um *cluster*. Se a barata for a melhor no seu campo visual ou *cluster*, então irá para a posição da melhor barata de todo o enxame segundo (48), sendo seguida pelas restantes baratas que fazem parte do seu *cluster*.

$$x'(i) = x(i) + step \times rand \times (P(i) - x(i)), \quad x(i) \neq P(i) \quad (47)$$

$$x'(i) = x(i) + step \times rand \times (P_g - x(i)), \quad x(i) = P(i) \quad (48)$$

Em (47), $x'(i)$ é a nova posição da barata i , $x(i)$ é a posição actual, $rand$ é um valor aleatório uniformemente distribuído e $P(i)$ a posição associada ao melhor valor de *fitness* do cluster. P_g é a posição associada ao melhor valor de *fitness* do enxame em (48). Na figura 3-7 está representada a evolução da posição de uma barata tal como em (47).

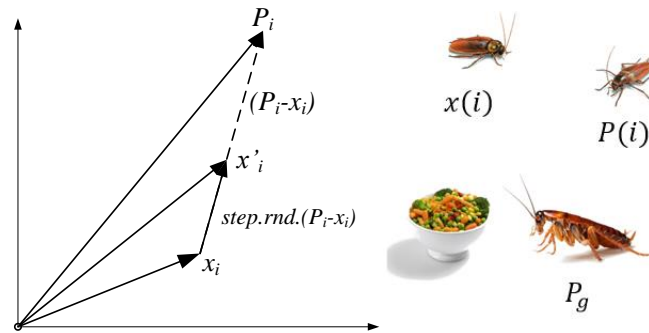


Figura 3-7 – Comportamento de uma partícula no CSO.

Na figura 3-8 está representada a evolução das baratas do enxame em função do número de iterações para a função *De Jong* para duas dimensões, 10 partículas.

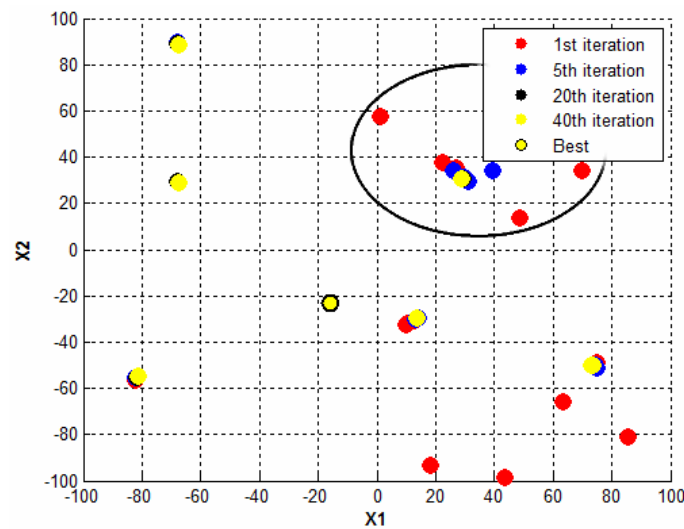


Figura 3-8 - Evolução do algoritmo CSO na busca pela melhor solução.

Verifica-se o *chase swarming* característico do CSO, bem como o deslocamento das baratas para as proximidades do valor mínimo da função com o evoluir do número de iterações. No CSO, as baratas criam aglomerados (*clusters*), com vários *clusters* de

baratas a convergir para a melhor posição encontrada até ao momento pelo enxame, tal como está representado na figura 3-8 com o círculo.

Um possível pseudo-código é apresentado no anexo 3 [37].

3.5.7. *Sensing Cloud Optimization*

O *Sensing Cloud Optimization* (SCO) é uma técnica heurística baseada em nuvens de partículas proposta em [7] e [39] não se baseando em qualquer comportamento biológico. Cada partícula da nuvem tem como função analisar o espaço de busca de modo a ajudar na definição do posicionamento da partícula central da nuvem, que é a única candidata a óptimo. A posição da partícula central é definida através de uma média pesada entre a posição da melhor partícula da nuvem e o vértice de regressão polinomial de segunda ordem. A dimensão da nuvem, definida dinamicamente, é obtida através de uma regressão polinomial de primeira ordem da distância entre a partícula central e as restantes. O SCO opera em dois estados complementares, em que o primeiro consiste na avaliação do *fitness* da nuvem de partículas e o segundo numa análise estatística com dois objectivos: o de direccionar a mesma nuvem e de definir a sua expansão. Uma das particularidades do SCO encontra-se no segundo objectivo, o qual capacita este algoritmo de controlar o seu próprio tamanho no espaço de busca. O controlo da expansão e contracção da nuvem serve para acelerar o tempo de convergência e evitar mínimos locais.

A partícula central é inicialmente criada a partir de (49), na qual i representa o número de dimensões do problema, garantindo-se logo que a mesma se encontra dentro do espaço de busca.

$$X_{q(i)} = Rnd(0,1). (X_{\max(i)} - X_{\min(i)}) + X_{\min(i)} \quad (49)$$

A criação da nuvem é feita através de uma distribuição gaussiana com média definida pela posição da partícula central dispersa segundo um desvio padrão inicial.

Após a criação da nuvem, todas as partículas são avaliadas, e com os seus valores de *fitness*, define-se uma regressão polinomial de segunda ordem para cada dimensão do espaço de busca. Estas regressões permitem ter uma ideia acerca da forma do espaço de

busca sob a nuvem. No caso de as regressões serem convexas pode-se definir quais os seus valores mínimos $t_{p(i)}$ e assim ter uma indicação para onde a partícula central se deverá mover. Assim a posição da partícula central resulta da soma ponderada entre o valor de $t_{p(i)}$ e o melhor de *fitness* encontrado pela nuvem, $X_{b(i)}$, como é demonstrado em (50) e representado na figura 3-9. Os pesos da soma ponderada são obtidos através do coeficiente de correlação da regressão polinomial de segunda ordem, $R_{q(i)}^2$.

$$X_{q(i)} = R_{q(i)}^2(t_{p(i)}) + (1 - R_{q(i)}^2) \cdot X_{b(i)} \quad (50)$$

Na figura 3-9 está representada uma superfície de busca característica da expressão *De Jong* com duas dimensões cujo mínimo se encontra na posição (0,0). Está também representada uma nuvem de partículas com o respectivo ponto central (*old queen*) e as coordenadas do *trend point*, bem como a nova posição do ponto central.

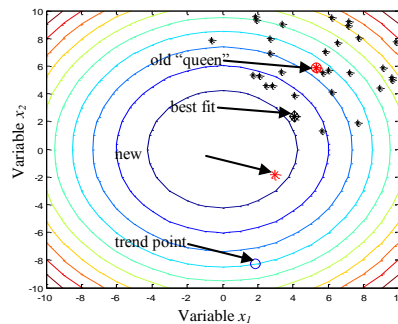


Figura 3-9 – Contour plot da partícula $X_{q(1)}$ [41].

Para uma melhor compreensão do funcionamento do algoritmo, na figura 3-10 e figura 3-11 estão representadas as nuvens de partículas segundo cada uma das dimensões, indicando em cada uma delas a regressão de segunda ordem com o respectivo $t_{p(i)}$ e o coeficiente de correlação. Analisando as figura 3-11 e figura 3-12 pode ver-se o $t_{p(i)}$ que resulta da regressão segundo cada variável. Se a posição da nova partícula central fosse definida apenas pelo valor de $t_{p(i)}$, a solução afastar-se-ia para um valor de *fitness* superior ao da iteração anterior, como se pode ver na figura 3-9. Neste caso, o peso da posição da melhor partícula do enxame, conforme definido em (50), ajuda a que a nova partícula central caminhe para uma solução com menor valor de *fitness*. Isto ajuda também a uma maior estabilidade do algoritmo diminuindo as oscilações da solução.

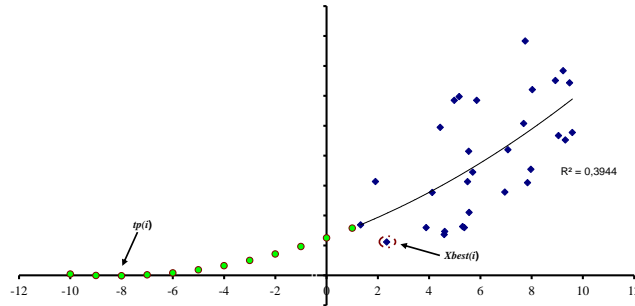


Figura 3-10 – Regressão polinomial para a dimensão $X_{(2)}$ [41].

Por exemplo, no caso da dimensão x_1 a regressão apresenta um coeficiente de correlação de cerca de 0,5, pelo que a nova posição da partícula central terá um peso igual para o $t_{p(i)}$ e para o $X_{b(i)}$.

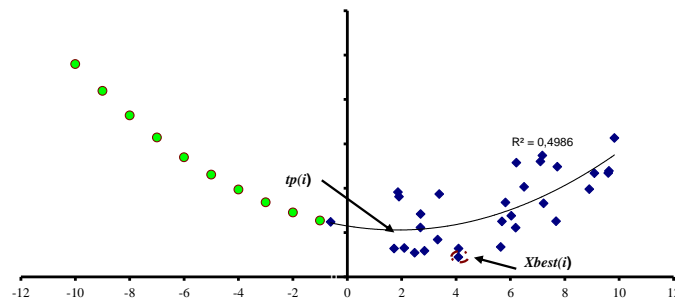


Figura 3-11 – Regressão polinomial para a dimensão $x_{(1)}$ [41].

Após a nova partícula central ser criada é necessário definir a nova nuvem que se criará à sua volta. A posição das partículas é novamente definida por uma distribuição Gaussiana centrada na partícula central e variância definida dinamicamente pela expressão (51).

$$\sigma_{(i)}^{2(k+1)} = \sigma_{(i)}^{2(k)} \cdot F_{S_1(i)}^{(k)} \cdot F_{S_2(i)}^{(k)} \quad (51)$$

Na expressão, k representa o número de iterações e F_{S_1} e F_{S_2} são dois coeficientes que resultam da seguinte análise. Paralelamente ao cálculo da regressão calcula-se a

distância euclidiana entre cada uma das partículas da nuvem e a partícula central através de (52) e representado na figura 3-12.

$$\varphi_{(j)} = \left\| X_{q(j)} - x_{(j)} \right\| \quad (52)$$

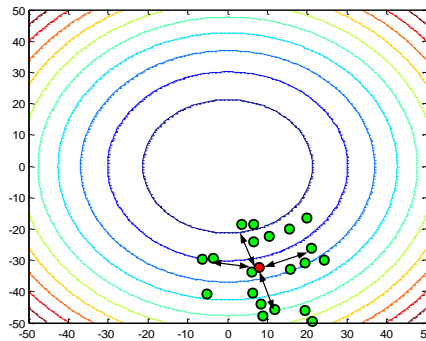


Figura 3-12 – Cálculo da distância de cada partícula à central [41].

Com os resultados obtidos podem-se obter dois gráficos exemplificativos como os da figura 3-13 a) e b).

No da figura 3-13 a), partículas com diferentes distâncias à partícula central apresentam valores de *fitness* muito próximos significando que a superfície de busca coberta pela nuvem é bastante plana. Sendo assim é necessário alargar a busca aumentando a área da nuvem. No caso da figura 3-13 b), partículas com maior distância ao ponto central apresentam maiores valores de *fitness*, o que significa que a nuvem se encontra sobre um mínimo pelo que deverá encolher de modo a refinar a busca.

Traçando uma regressão linear e derivando obtêm-se as expressões (53) e (54). No caso de a derivada ser muito acentuada significa que a nuvem se encontra sobre um vale estreito e profundo, pelo que tendencialmente a dispersão da nuvem deve ser bastante diminuída.

$$\nabla \varphi = \frac{df(\varphi_{(j)})}{d\varphi_{(j)}} \left(\frac{\sigma_{\varphi_{(j)}}}{\sigma_{fitness_{(j)}}} \right) \quad (53)$$

$$\Delta \varphi = \frac{1}{1 + h \cdot \nabla \varphi} \quad (54)$$

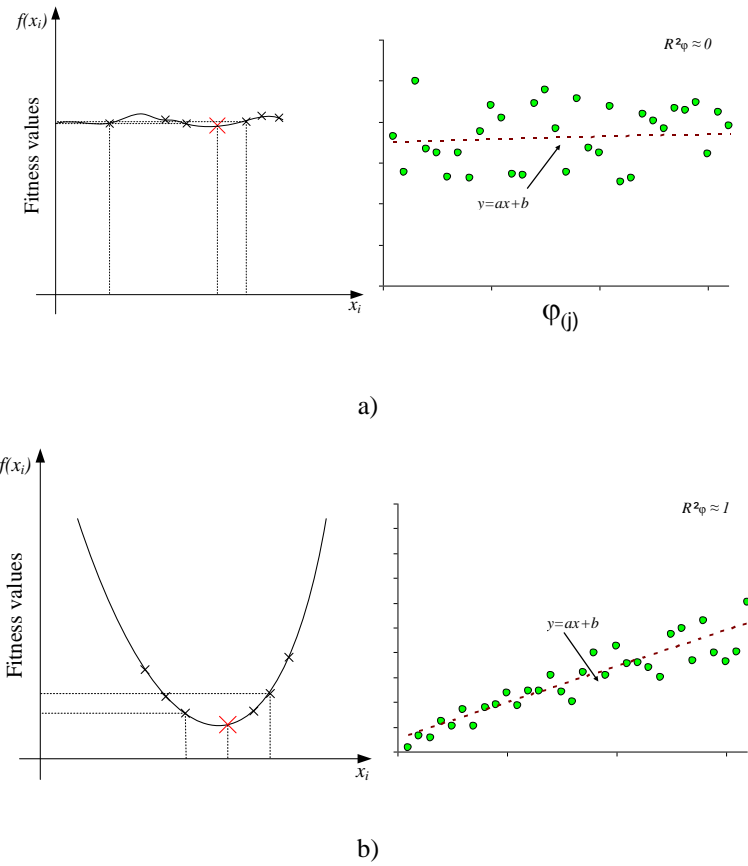


Figura 3-13 – Valores de fitness em função da distância à partícula central.

No caso de a derivada ser pouco acentuada, tendencialmente a dispersão da nuvem não deverá diminuir. Este processo é regulado automaticamente através das expressões (55) e (56) das quais resultam os gráficos da figura 3-14 e figura 3-15.

$$F_{s1} = 1 + \frac{\Delta\phi - 1}{e^{\left(\frac{-8(R_\phi^2 - t_c)}{t_s}\right)}} \quad (55)$$

$$F_{s2(i)} = 1 - \frac{(K - 1)}{1 + e^{\left(-t_m R_{d(i)}^2 + t_n\right)}} \quad (56)$$

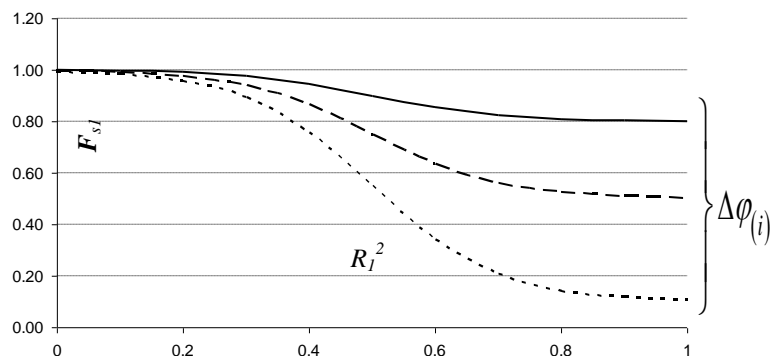


Figura 3-14 – Característica de F_{s1} [41].

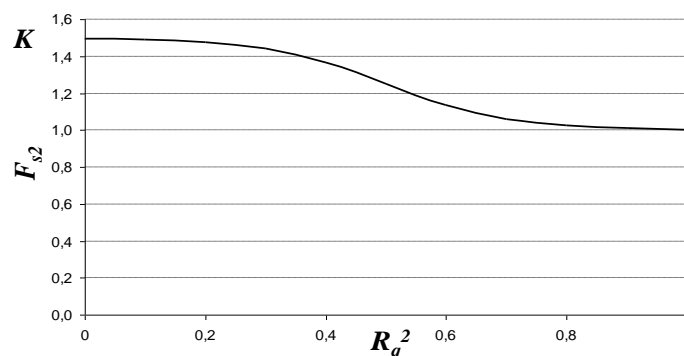


Figura 3-15 – Característica de F_{s2} [41].

Analisando-se a figura 3-13 e figura 3-14 conclui-se que no caso do coeficiente de correlação linear, R_l^2 , da regressão da figura 3-13 ser muito elevado, a nuvem deverá diminuir a sua dispersão segundo (55), resultando F_{s1} . No caso de a derivada ser muito elevada a escolha da curva dependerá do resultado da expressão (54). Se o coeficiente de correlação for muito baixo como é o caso da figura 3-13 a), o valor de F_{s1} dever-se-á manter próximo da unidade. Fazendo a mesma análise aos coeficientes de correlação, R_q^2 das figura 3-10 e figura 3-11, poder-se-á concluir que a nuvem deverá aumentar a sua dispersão através de F_{s2} . Este parâmetro depende do coeficiente de correlação da regressão de segunda ordem R_q^2 . Em caso de ser muito baixo significa que a regressão não consegue caracterizar de forma satisfatória a superfície de busca pelo que a nuvem deverá aumentar a sua dimensão. Nesse sentido, a variância da distribuição das partículas dependerá do valor da iteração anterior e do produto entre F_{s1} e F_{s2} tal como indicado em (51).

É apresentado um possível fluxograma do SCO no anexo 4.

3.5.8. Particle Cloud Optimization

O *Particle Cloud Optimization* (PCO) é uma técnica híbrida entre o PSO e o SCO e é proposta como contribuição original nesta dissertação. O objectivo deste híbrido é o de aproveitar os pontos fortes do PSO e do SCO e assim atingir um algoritmo mais completo. Os pontos fortes em vista são a capacidade de busca, e velocidade de convergência, respectivamente do PSO e do SCO. Na figura 3-16 estão representadas as evoluções do PSO e do SCO para a função *De Jong* com 2 dimensões, 30 partículas e 500 gerações.

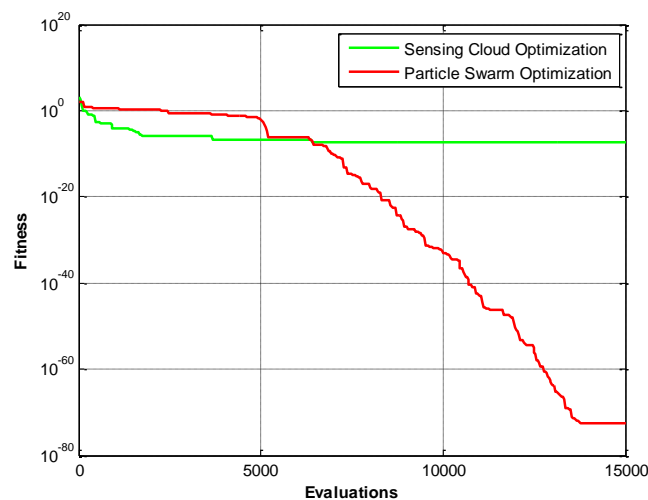


Figura 3-16 – Evolução do PSO (vermelho) e do SCO (verde).

Como se pode ver, o PSO atinge um valor de *fitness* cerca de $1e^{60}$ vezes menor que o atingido pelo SCO. Outro ponto relevante acontece quando os algoritmos chegam às 5000 avaliações e o valor atingido pelo PSO alcança o atingido pelo SCO há cerca de 3000 avaliações. Ou seja, o PSO necessita de muito mais avaliações para obter o valor atingido pelo SCO, mas depois prossegue na sua busca, enquanto o SCO estagna.

O PCO será assim constituído por duas fases: SCO e depois PSO. Na primeira fase, comporta-se como SCO até atingir um patamar de estagnação a partir do qual iniciará o PSO. A fase PSO irá assim começar a sua busca a partir de uma posição já otimizada e poderá usar as avaliações restantes para obter um valor de *fitness* melhor que o do PSO.

A implementação do PCO é feita de forma modular, ligando o PSO (explicado em 3.5.4) ao SCO (explicado em 3.5.8). Os parâmetros do PCO são os mesmos que os

indicados para o PSO e o SCO separadamente. O único parâmetro exclusivo do PCO tem a ver com o início da sua segunda fase, ou seja, o começo do PSO. Este parâmetro reflecte o critério de entrada do PSO, que se escolhe como sendo o número limite de iterações sem melhoria por parte do SCO.

Na figura 3-17 estão representados a evolução dos valores de *fitness* para os algoritmos PSO, SCO e o modelo híbrido proposto (PCO), que resulta do PSO e SCO. A simulação é feita com 30 partículas em cada algoritmo e 3000 gerações.

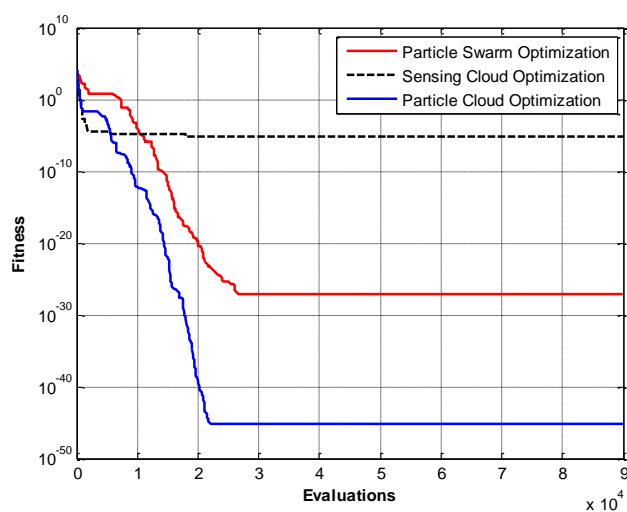


Figura 3-17 – Representação do PCO face ao PSO e SCO.

Além de o PCO ser computacionalmente mais rápido que os algoritmos em que se baseia, demonstra também uma maior capacidade de busca atingindo um mínimo melhor que o do PSO e o do SCO.

O patamar de estagnação do PCO deve-se a dois factores. O primeiro tem a ver com o momento em que o PSO entra em funcionamento, ao fim de um determinado número de iterações sem melhoria no SCO. Este número de iterações sem melhoria poderá ser reduzido ou aumentado, mas ambos os extremos têm defeitos. Se o número for demasiado pequeno, então o PCO não recolhe os benefícios totais do SCO. Se o número for demasiado elevado, o PCO prolonga o seu patamar de estagnação, o que prejudica a sua velocidade de convergência e capacidade de busca. Neste caso da figura 3-17, o número de iterações é 5, ou seja 150 avaliações. O segundo factor a considerar sobre o patamar de estagnação diz respeito à fase PSO do PCO. Dado que o PSO começa a sua busca a partir de uma posição com algum grau de optimização, os seus parâmetros não

podem ser iguais aos que tem o PSO isolado. Pelo contrário, a velocidade inicial, v , das partículas tem de ser reduzida, visto que não é vantajoso voltar a procurar no espaço de busca todo. Os coeficientes c_1 e c_2 têm também de ser afinados, bem como o ω , pela mesma razão que a da velocidade inicial. O desafio reside no elevado número de parâmetros que carecem de ser determinados.

Um possível pseudo-código para o PCO está descrito no anexo 5.

Capítulo IV

Optimização através de Inteligência de enxames

Resumo

Neste capítulo é apresentado um conjunto de funções de teste de referência para a aferição das performances de algoritmos de optimização. São também validadas as performances das heurísticas apresentadas na resolução das funções de referência.

4. Optimização através de Inteligência de enxames

4.1. Introdução

No interesse da validação das meta-heurísticas implementadas, o seu desempenho num conjunto de funções de teste é comparado com os desempenhos observados em [34], [42] e discutido. O sucesso é medido pela média dos valores de *fitness* atingidos, a robustez através do desvio-padrão e a velocidade de convergência pelo tempo de computação até ser alcançado o melhor valor de *fitness*, tal como feito em [37]. Para cada execução do programa em *Matlab*[®], são registados diversos dados, embora somente sejam utilizados três (\bar{X}, σ, t) para discussão de resultados.

As simulações são feitas com 3 parâmetros em comum nos 5 algoritmos, que são o número de partículas, número de gerações e número de corridas. O número de partículas indica de quantas partículas dispõem os algoritmos para espalhar pelo espaço de busca, o número de gerações indica quantas iterações o algoritmo pode correr no máximo e o número de simulações indica o número de vezes que o algoritmo é corrido.

O programa no qual os algoritmos estão implementados é executado num computador de secretária, com processador de 2,40GHz e 2GB de memória RAM.

Na tabela 4-1 são indicados os parâmetros do PSO nas simulações executadas com o mesmo.

Tabela 4-1 – Parâmetros do PSO

	c_1	c_2	$\omega_{m\acute{a}x}$	$\omega_{m\acute{i}n}$
PSO	2	2	0,9	0,01

Na tabela 4-2 são indicados os parâmetros do BCO nas simulações executadas com o mesmo.

Tabela 4-2 – Parâmetros do BCO

	Limite tentativas
BCO	70

Na tabela 4-3 são indicados os parâmetros do CSO nas simulações executadas com o mesmo.

Tabela 4-3 – Parâmetros do CSO

	Iteração de dispersão	Porcentagem de dispersão	Campo visual	Step
CSO	50	100%	100	2

Na tabela 4-4 são indicados os parâmetros do SCO nas simulações executadas com o mesmo.

Tabela 4-4 – Parâmetros do SCO

	h	t_m	t_n	t_c	t_s	σ^2	K
SCO	1	50	1	0,5	1	$\frac{P_{m\acute{a}x} - P_{min}}{2}$	1,5

Os parâmetros do PCO são idênticos aos utilizados para o PSO e SCO.

4.2. Funções de teste

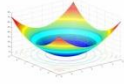
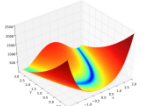
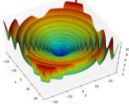
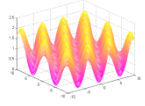
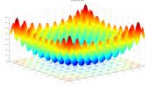
Existem funções padrão (*standard*) utilizadas para validar métodos de optimização que, pela sua complexidade, número de soluções óptimas e convexidade, possibilitam depreender a *performance* de cada método, quais as suas vantagens e quais as suas desvantagens em relação aos demais. Estas funções podem ser categorizadas pelo seu número de mínimos, em que uma função com apenas um mínimo é denominada de unimodal e uma com vários mínimos (um global e vários locais) é denominada de multimodal. Funções unimodais conseguem indicar a velocidade de convergência, enquanto que funções multimodais conseguem indicar a capacidade do algoritmo não ficar retido num mínimo local [43].

De forma a que os diferentes algoritmos possam ser comparados, os seus parâmetros foram uniformizados. Deste modo, o número de partículas, número de gerações e número de simulações é igual em todos os algoritmos, assim como o número de

dimensões da função optimizada. Para as funções f_1 (*De Jong*) e f_3 (*Schaffer*) consideraram-se 3000 iterações e para as funções f_2 (*Rosenbrock*), f_4 (*Griewank*) e f_5 (*Rastrigin*) 5000 iterações, o número de partículas é 30 e 20 simulações são feitas para cada optimização. As funções são optimizadas para 5 e 10 dimensões.

Relativamente ao número de partículas, o CSO começa com o mesmo número que os demais algoritmos e é estabelecido 10 como limite mínimo de baratas sobreviventes ao comportamento de implacabilidade, em que as mesmas comem as mais fracas.

Tabela 4-5 – Funções de teste utilizadas

Função	Expressão matemática
 <i>De Jong</i>	$f_1(\vec{x}) = \sum_{i=1}^n x_i^2 \quad (57)$
 <i>Rosenbrock</i>	$f_2(\vec{x}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \quad (58)$
 <i>Schaffer</i>	$f_3(\vec{x}) = 0,5 + \frac{\text{sen}^2(\sqrt{x_1^2 + x_2^2}) - 0,5}{(1 + 0,001(x_1^2 + x_2^2))^2} \quad (59)$
 <i>Griewank</i>	$f_4(\vec{x}) = \frac{1}{4000} \left(\sum_{i=1}^n (x_i - 100)^2 \right) - \left(\prod_{i=1}^n \left(\frac{x_i - 100}{\sqrt{i}} \right) \right) + 1 \quad (60)$
 <i>Rastrigin</i>	$f_5(\vec{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (61)$

A função *De Jong* (f_1) é contínua, convexa e unimodal, *i.e.* sem mínimos locais, excepto o global em $x_i = 0$, onde $f(x_i) = 0$. A função *Rosenbrock* (f_2) é não-convexa, com um mínimo global dentro de um vale longo, estreito e com forma parabólica e plana. Encontrar o vale é trivial, mas convergir para o mínimo global não é, mesmo sendo uma função unimodal. O mínimo global encontra-se em $x_i = 0$ onde $f(x_i) = 0$.

A função *Schaffer* (f_3) é bidimensional, apresentando diversos mínimos locais. Existe um mínimo global em $(x_1, x_2) = (0,0)$, onde $f(x_i) = 0$. A função *Griewank* (f_4) é amplamente utilizada para testar algoritmos de optimização. Num intervalo de $[-600, 600]$ tem 191 mínimos locais, com o global em $x_i = 0$, onde $f(x_i) = 0$. A função *Rastrigin* (f_5) é multimodal e o seu número de mínimos locais aumenta exponencialmente. Existe um mínimo global em $x_i = 0$ onde $f(x_i) = 0$ [44].

Na tabela 4-6 estão indicados os parâmetros aplicados em todos os algoritmos.

Tabela 4-6 – Parâmetros das simulações

Número de partículas	30				
Simulações	20				
Gerações	f_1	f_2	f_3	f_4	f_5
	3000	5000	3000	5000	5000

4.2.1. Resultados

Na figura 4-1 encontra-se representada a evolução de cada algoritmo até atingirem o respectivo valor mínimo na optimização da função *De Jong*, com 5 dimensões.

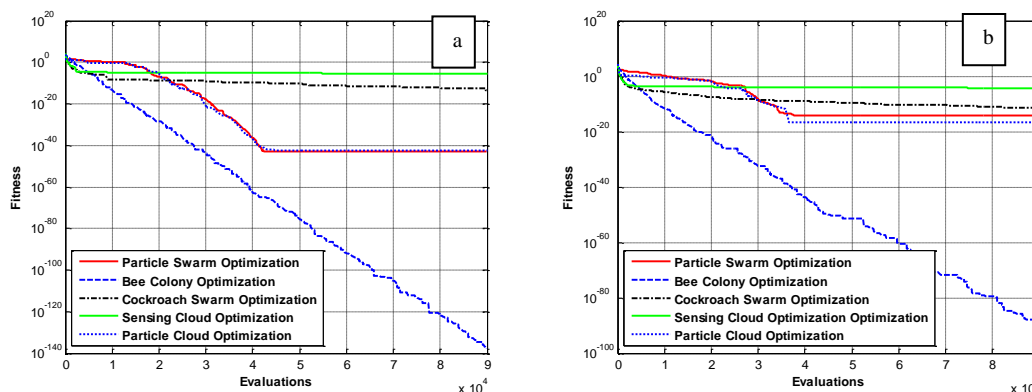


Figura 4-1 – a) Evolução da melhor simulação e b) Evolução média de cada algoritmo para a função *De Jong* com 5 dimensões.

Analisando a figura 4-1 verifica-se que todas as técnicas experimentadas conseguem obter boas aproximações ao mínimo da função *De Jong* tanto em termos absolutos (o mínimo de todas as corridas) como em termos médios (média de todas as corridas). Apesar de todas as técnicas terem obtidos resultados satisfatórios, pode-se destacar a performance do BCO em comparação com as restantes técnicas. Com o aumento de

dimensões, verifica-se que qualquer uma das técnicas consegue obter valores satisfatórios, embora atingindo valores mínimos superiores (por exemplo, o BCO consegue um mínimo médio cerca de 2 vezes maior) em relação ao caso com 5 dimensões. É também de salientar os comportamentos das diferentes técnicas em termos médios e, em ambas as dimensões, o BCO continua a ter uma maior capacidade de busca que as restantes.

Na tabela 4-7 são indicados os resultados dos 5 algoritmos para as 5 funções diferentes e para 5 e 10 dimensões. Do anexo 6 ao anexo 14 encontram-se as curvas de evolução dos 5 algoritmos para as 5 funções com 5 e 10 dimensões.

Tabela 4-7 – Resultados de optimização das funções de teste. *tempo necessário para chegar ao melhor valor de fitness.

Heurística $f_n(x)$	Min		Média		Máx		Desvio		t(s)		Aval.		
	5	10	5	10	5	10	5	10	5	10	5	10	
PSO	f_1	9,82 e ⁻⁴⁴	3,8 e ⁻³	5,71 e ⁻¹⁵	5,31 e ⁻¹	1,13 e ⁻¹³	3,37	2,53 e ⁻¹⁴	8,9 e ⁻¹	2,71	3,50	51960	26970
	f_2	1,51 e ⁻³	5,75	5,51 e ⁴	1,3 e ⁴	1 e ⁶	2,5 e ⁵	2,23 e ⁵	5,57 e ⁴	2,62	13,1	25770	149160
	f_3	0		3,89e ⁻³		9,72e ⁻³		4,88e ⁻³		0,90			19350
	f_4	1,93 e ⁻¹²	0,27	6,40 e ⁻⁴	4,86	6,57 e ⁻³	91,01	1,56 e ⁻³	20,28	5,27	10,8	90930	110910
	f_5	0	3,99	3,91	18,3	2,69 e ¹	40,9	5,74	9,91	1,88	6,86	22200	40110
BCO	f_1	1,79 e⁻¹³⁸	2,98 e⁻³⁵	5,10 e⁻⁸⁹	1,85 e⁻³⁰	1,02 e⁻⁸⁷	2,28 e⁻²⁹	2,28 e⁻⁸⁸	5,68 e⁻³⁰	12,7	11,7	89940	44880
	f_2	3,2 e ⁻²	2,15 e⁻¹	3,01	3,12	11,0	16,2	3,5	3,76	8,36	9,87	35970	37410
	f_3	0		1,97 e ⁻⁵		1,82 e ⁻⁴		4,78 e ⁻⁵		4,98			24840
	f_4	7,89 e ⁻²	0,14	7,02 e ⁻²	0,34	0,27	0,62	5,33 e ⁻²	0,12	40,2	76,1	31860	37020
	f_5	0	0	0	8,74 e⁻¹⁴	0	12,7 e⁻¹²	0	3,81 e⁻¹³	2,88	13,1	14370	31740
CSO	f_1	4,73 e ⁻¹⁴	7,61 e ⁻¹⁷	3,56 e ⁻¹²	9,87 e ⁻¹⁵	2,86 e ⁻¹¹	4,79 e ⁻¹⁴	7,50 e ⁻¹²	1,35 e ⁻¹⁴	1,65	29,4	89790	89970
	f_2	1,29 e ⁻²	0,72	37,34	25,11	627,18	205,18	139,14	54,43	62,2	48,0	149910	149910
	f_3	0		0		0		0		11,2			37140
	f_4	2,85 e ⁻⁶	5,17 e⁻⁹	3,77 e⁻⁵	3,91 e⁻⁵	4,96 e⁻⁴	7,78 e⁻⁴	1,15 e⁻⁴	1,74 e⁻⁴	36,4	35,1	76800	148950
	f_5	0,99	7,96	4,03	22,6	9,95	40,8	2,42	9,83	48,8	43,9	140100	86340
SCO	f_1	2,32 e ⁻⁶	5,49 e ⁻⁴	4,90 e ⁻⁵	1,25 e ⁻³	2,27 e ⁻⁴	2,46 e ⁻³	4,88 e ⁻⁵	5,49 e ⁻⁴	6,26	3,17	54780	19800
	f_2	8,53 e ⁻²	3,52	23,16	44,76	133,76	356,13	40,06	89,80	7,95	14,5	58290	86250
	f_3		3,15 e ⁻⁶		5,04 e ⁻⁵		2,55 e ⁻⁴		6,16 e ⁻⁵		6,05		68100
	f_4	5,11 e ⁻⁶	3,19 e ⁻³	6,89 e ⁻⁴	9,23 e ⁻³	3,63 e ⁻³	2,14 e ⁻²	1,29 e ⁻³	5,47 e ⁻³	5,46	14,1	33210	55740
	f_5	0,40	8,40	2,27	21,45	4,04	30,81	1,06	5,05	13,37	3,11	87900	19050
PCO	f_1	6,03 e ⁻⁵²	1,04 e ⁻¹⁴	4,76 e ⁻²⁵	4,59 e ⁻¹¹	9,48 e ⁻²⁴	2,33 e ⁻¹⁰	2,12 e ⁻²⁴	6,33 e ⁻¹¹	2,86	5,79	28800	25140
	f_2	2,48 e⁻³⁰	6,64	2,19	335,82	12,11	4180,86	2,90	941,81	0,82	12,83	18660	58890
	f_3		5,55 e ⁻¹⁷		5,34 e ⁻³		9,72 e ⁻³		4,96 e ⁻³		0,19		10230
	f_4	2,20 e⁻¹⁴	3,28 e ⁻³	0,68	0,30	0,96	0,93	0,29	0,39	9,95	7,87	150000	3270
	f_5	7,11 e ⁻¹⁵	4,32	2,07	11,08	5,97	27,30	1,79	6,26	1,27	7,45	27810	70020

Em negrito estão assinalados os melhores valores alcançados para cada indicador em relação a cada função, algoritmo e dimensão.

Na figura 4-2 são representados graficamente os resultados da tabela 4-7, discriminados por três critérios de avaliação (capacidade de busca, robustez e velocidade de convergência). A percentagem indica a relação com que dado algoritmo foi melhor que os restantes. Por exemplo, a robustez do BCO é de 56%, o que indica que este algoritmo foi o mais robusto 56% das vezes.

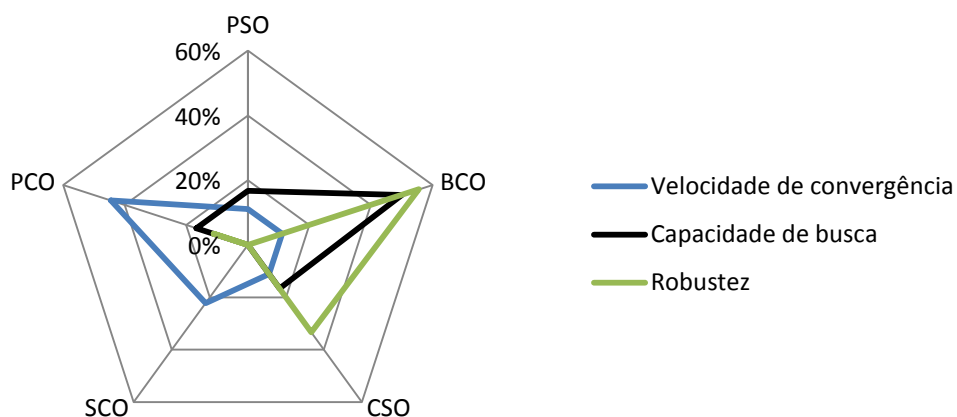


Figura 4-2 – Representação gráfica do desempenho de cada algoritmo.

É possível observar quais os algoritmos mais equilibrados, quais os que demonstram melhor capacidade de busca, velocidade de convergência e robustez. O CSO e o PCO apresentam o maior equilíbrio entre os três critérios avaliados.

Estes resultados são semelhantes aos encontrados em literatura, pelo que a validação dos algoritmos implementados é atingida. No capítulo cinco, estes algoritmos são aplicados a quatro casos estudados de PDE.

Capítulo V

Caso estudado

Resumo

Neste capítulo são resolvidos os casos estudados de despacho económico. Os casos estudados são quatro. O primeiro tem 3 grupos de função de custo igual, não convexa. O segundo tem 10 grupos, com funções de custo diferentes, não convexas e com multi-fuels. O terceiro tem 15 grupos, com funções de custo convexas e presença de perdas. O quarto tem 40 grupos, com funções de custo não convexas.

5. Caso estudado

5.1. Introdução

Para aplicar as meta-heurísticas implementadas à optimização do PDE, quatro casos de PDE são estudados, sendo o objectivo a minimização do custo de produção de potência para satisfazer a carga, bem como as restrições de igualdade e de desigualdade.

As performances de cada técnica são avaliadas pela capacidade de busca, robustez e velocidade de convergência. A capacidade de busca é definida pelo valor mínimo obtido, a robustez pelo valor médio e desvio padrão das diversas corridas. A rapidez é medida pelo tempo de convergência e número de avaliações necessárias para alcançar o mínimo.

Na tabela 5-1 estão indicados os parâmetros aplicados em cada algoritmo.

Tabela 5-1 – Parâmetros das simulações

Número de partículas	20
Simulações	20
Gerações	500

5.2. Penalização de soluções não óptimas

Um factor de penalização pode ser aplicado para condicionar de diversas formas o progresso do algoritmo na sua busca pelo valor óptimo. Uma penalidade pode ser utilizada para limitar as perdas por efeito *Joule* na transmissão de energia [16]; as soluções impossíveis (*unfeasible*), favorecendo as possíveis (*feasible*) [5]; o desvio entre a potência produzida e a potência da carga mais perdas e ainda para penalizar soluções dentro de zona proibida [45]. Um factor de penalização serve também para incorporar as restrições na função objectivo.

Em PDE, esta metodologia pode ser aplicada para os limites técnicos, rampas e zonas proibidas, bem como reserva girante. Habitualmente, a função de fitness tem um factor de penalização, com o objectivo de anular o desvio entre a potência produzida e a consumida [10].

$$F_{penalidade_j^k} = q \times \sum_{i=1}^N P_i^k - D - P_L^k \quad (62)$$

A escolha do parâmetro q pode ser problemática, visto que um valor reduzido poderá implicar um desvio maior entre a produção e o consumo. Ou seja, se a penalidade não for suficiente, o *fitness* associado a uma produção deficitária poderá ser mais baixo que o de uma produção igual ao consumo. Por outro lado, se q for demasiado elevado, tal poderá distorcer o espaço de busca.

Nesse sentido, a função de custo definida em (36) passa a ter a formulação apresentada em (63). Com esta alteração pode-se eliminar a restrição de igualdade.

$$F_i(P_i)_{penalidade} = c_i P_i^2 + b_i P_i + a_i + |e_i \times \text{sen}[f_i \times (P_{i,min} - P_i)]| + q \times \sum_{i=1}^N P_i^k - D - P_L^k \quad (63)$$

5.3.Caso estudado 1 – 3 grupos

O primeiro caso estudado tem 3 grupos com função de custo igual, não convexa, crescente e derivada decrescente no último troço de potência, como se pode ver na figura 5-1. Para simular esta função de custo, a mesma é seccionada por intervalos de potência, tal como está indicado na expressão (64). A metodologia desta simulação passa por aplicar carga gradual, com o objectivo de ver a repartição de carga indicada por cada algoritmo, sendo o resultado comparado com uma repartição igualitária e com uma repartição desigual, em que os grupos serão maximizados sucessivamente, enquanto os restantes são minimizados.

$$F(P) = \begin{cases} 5P + 500, \text{ para } 100 \leq P < 200 \\ 6P + 300, \text{ para } 200 \leq P < 300 \\ 7P, \text{ para } 300 \leq P < 400 \\ 6P + 400, \text{ para } 400 \leq P \leq 500 \end{cases} \quad (64)$$

$$100 \leq P \leq 500 \text{ (MW)} \quad (65)$$

A figura 5-2 é a representação gráfica do resultado obtido pelo PSO no caso 1, encontrando-se os restantes resultados do caso estudado 1 desde o

anexo 15 até ao anexo 16, incluindo o da figura 5-2.

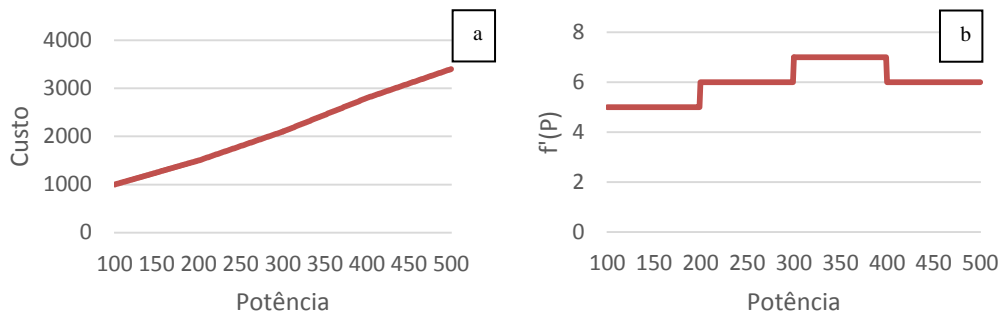


Figura 5-1 – a) Função de custo do caso 1 e b) derivada da função de custo do caso 1.

Como se pode ver pela figura 5-1, a função de custo do caso 1 é monotamente crescente e a derivada é crescente, excepto no último troço, logo a função de custo não é convexa. Na figura 5-2 está representado o resultado do caso estudado 1 com o PSO.

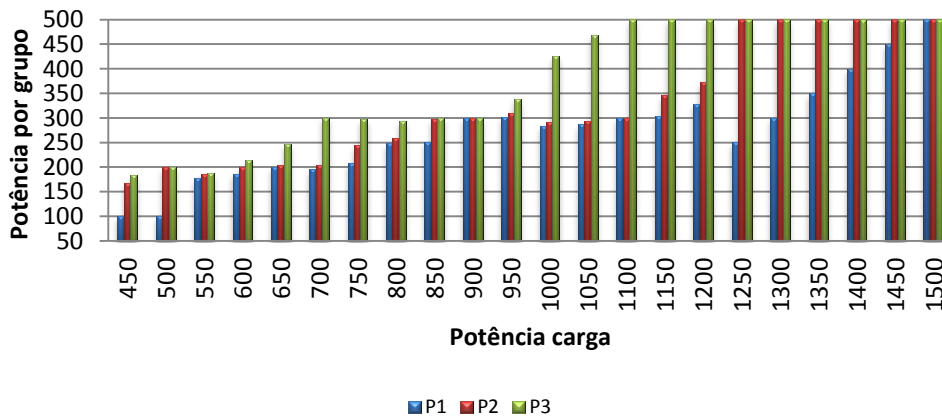


Figura 5-2 – Resultado do caso 1 com o PSO.

Para valores baixos de carga, o valor de potência produzida por cada grupo encontra-se na zona em que a função de custo é convexa, pelo que a repartição de potência por grupo deve ser igual. Quando a carga atinge um nível que permita ter um grupo perto do máximo é que a função de custo do grupo estará na zona não convexa, logo o óptimo passa por colocar um grupo no máximo e os restantes em valores mais baixos. Na figura 5-2 verifica-se que o PSO consegue detectar a não linearidade do último troço e

localizar o óptimo, passando de uma repartição igualitária⁴ a uma maximização sucessiva⁵. Quando o valor de carga é suficiente, o PSO passa a maximizar um grupo, como se pode ver a partir de uma carga de 1050 MW. Outro ponto interessante acontece com uma carga de 1250 MW, em que dois grupos passam a gerar a potência máxima e o outro passa para a potência mínima. Na figura 5-3 é comparado o resultado do PSO com uma repartição igualitária e com uma maximização sucessiva. É possível observar que até uma carga de 1000 MW, é mais barato repartir a carga igualmente pelos três grupos. Já com 1050 MW, a maximização sucessiva é mais barata, atingindo um custo mais baixo. Daí para a frente, a maximização sucessiva é mais económica, excepto com 1100 MW e 1150 MW. O PSO consegue sempre igualar o menor valor entre a repartição igualitária e maximização sucessiva, como se pode ver na figura 5-3.

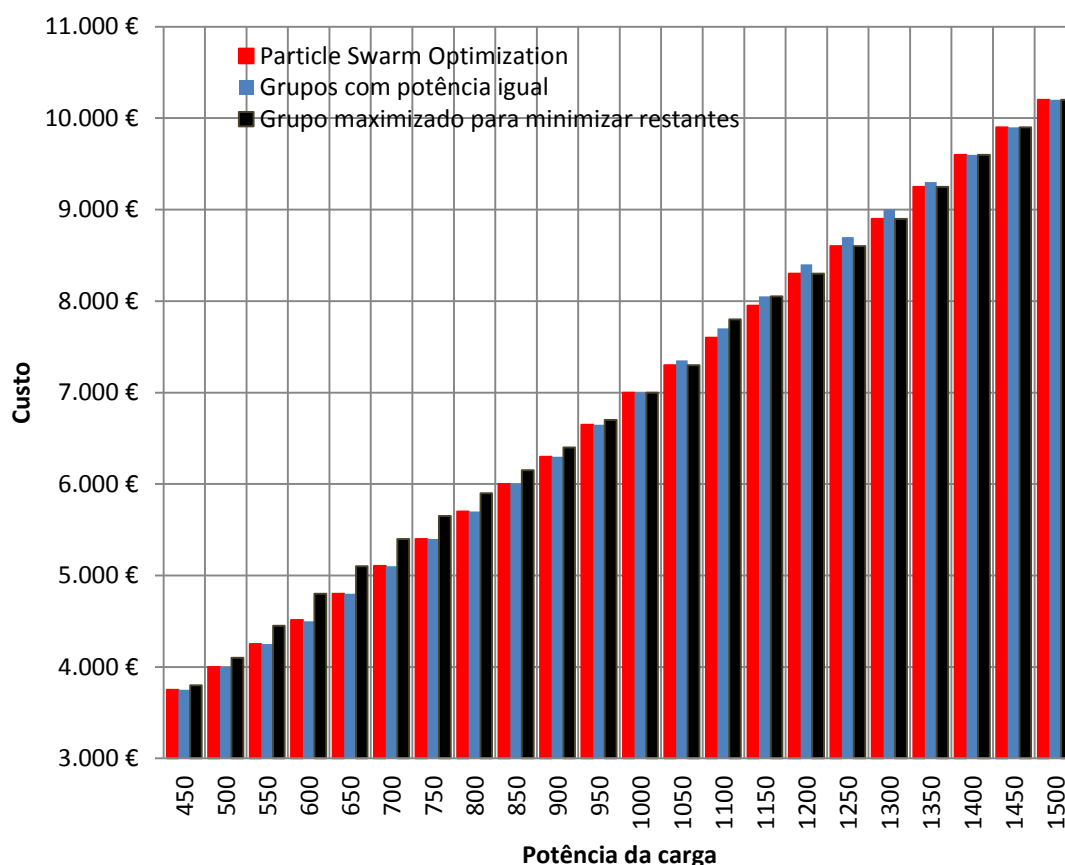


Figura 5-3 – Resultado do caso 1 com o PSO comparado com uma repartição igualitária e com uma maximização sucessiva.

⁴ Variação de potência nos grupos igual.

⁵ O aumento de potência gerada é realizado com a maximização de um ou mais grupos (dependendo da carga), deixando os restantes minimizados.

5.4.Caso estudado 2 – 10 grupos

Este caso estudado é baseado em [46] sendo formulado para 10 grupos com funções de custo diferentes, não convexas, com mais do que um combustível (*multi-fuels*).

Cada um dos grupos é constituído por 3 combustíveis, excepto o primeiro grupo que tem apenas 2.

Com este caso pretende-se mostrar a capacidade de resolução com problemas com funções de custo não contínuas pelo não são contempladas perdas nas linhas, zonas proibidas, rampas ou reserva girante. Considera-se que a carga são 2700 MW.

Na tabela 5-2 e tabela 5-3 estão apresentados os resultados obtidos para cada uma das meta-heurísticas estudadas.

Tabela 5-2 – Resultados para o caso estudado 2

Heurística	Mínimo	Máximo	Médio	Desvio	t (s)	Avaliações
PSO	717,6	853,4	797,6	55,1	11,4	14970
BCO	738,2	993,0	826,1	102,5	27,9	5190
CSO	734,9	1230,5	1021,8	210,8	5,13	14730
SCO	723,2	996,8	789,1	117,1	5,07	10032
PCO	667,9	765,8	716,4	37,7	5,7	14670

Tabela 5-3 – Valor de produção de cada grupo para o caso estudado 2

Grupo	fuel	PSO	fuel	BCO	fuel	CSO	fuel	SCO	fuel	PCO
1	1	236,0	2	250,0	1	135,7	1	192,9	2	250,0
2	2	51,3	1	178,2	1	196,4	1	200,9	1	165,3
3	2	289,4	1	294,4	2	410,8	2	392,1	1	266,5
4	1	119,9	2	147,5	2	165,8	2	150,5	3	201,3
5	2	365,6	1	287,1	1	294,1	3	420,1	2	373,3
6	3	264,7	1	145,8	3	264,9	3	218,9	3	214,3
7	2	356,1	3	392,7	1	244,1	1	239,3	2	376,9
8	3	264,5	1	113,9	1	129,8	1	102,5	2	188,9
9	3	440,0	3	427,6	3	375,3	3	387,6	3	407,8
10	1	312,5	2	462,9	2	483,2	3	395,1	1	255,6
Potência produzida [MW]		2700		2700		2700		2700		2700

Na tabela 5-3 estão apresentadas as melhores soluções de cada técnica. Verifica-se que todas conseguiram resolver o problema respeitando tanto os limites dos grupos bem

como a carga. Pelos resultados apresentados na tabela 5-2, verifica-se que o PCO demonstrou ter uma maior capacidade de busca e robustez que as restantes técnicas com o menor valor de custo, bem como o menor desvio padrão. Em termos de rapidez, não foi o mais rápido a convergir para o seu mínimo, embora demonstre grande velocidade de processamento, uma vez que conseguiu em 5,7 segundos realizar 14670 avaliações da função de custo. Comparativamente, o BCO necessita de muito mais tempo para fazer um menor número de avaliações.

5.5.Caso estudado 3

No caso estudado 3 é formalizado um problema com 15 grupos com funções de custo convexas. Este problema é vulgarmente apresentado na literatura [22] e nele são contempladas perdas nas linhas, restrições de mínima e máxima potência. Todos os grupos apresentam rampas de produção e, alguns, zonas proibidas de funcionamento. A carga é 2630 MW. Este caso estudado pretende avaliar a capacidade destas técnicas na resolução de problemas com elevado número de restrições.

Tabela 5-4 – Resultados para o caso estudado 3

Heurística	Mínimo	Máximo	Médio	Desvio	t (s)	Avaliações
PSO	33034,3	33272,2	33156,5	87	10,1	14970
BCO	32908,4	33112,7	33035,2	75,9	35,5	480
CSO	32992,8	33392,4	33200,7	148,6	8,4	12870
SCO	32992,1	33524,6	33301,1	214,5	7,5	11664
PCO	33090,5	33300,2	33213,5	87,7	13,7	14850

Tabela 5-5 – Valor de produção de cada grupo para o caso estudado 3

Grupo	Limites	PSO	BCO	CSO	SCO	PCO
1	[150, 455]	385,04	455,00	455,00	455,00	452,97
2	[150, 455]	380,00	380,00	278,28	380,00	342,93
3	[20, 130]	119,10	130,00	93,98	130,00	130,00
4	[20, 130]	130,00	130,00	130,00	130,00	119,80
5	[150, 470]	170,00	170,00	170,00	170,00	170,00
6	[135, 460]	460,00	426,96	460,00	460,00	365,00
7	[135, 465]	430,00	407,64	430,00	430,00	385,00
8	[60, 300]	160,00	160,00	128,56	60,00	96,25
9	[25, 162]	162,00	68,70	133,95	114,97	162,00
10	[25, 160]	115,18	60,29	160,00	36,74	122,59
11	[20, 80]	20,00	80,00	80,00	80,00	69,69
12	[20, 80]	55,00	66,04	79,60	20,00	80,00
13	[25, 85]	28,84	36,29	25,00	85,00	72,71
14	[15, 55]	15,00	55,00	15,00	55,00	40,66
15	[15, 55]	40,95	36,48	29,00	52,90	50,86
Potência		2671,11	2662,4	2668,37	2659,61	2660,46
Perdas [MW]		41,11	32,4	38,37	29,70	30,46

Também neste caso se constata que todas as técnicas conseguiram resolver o problema respeitando as restrições. No entanto, pelos resultados da tabela 5-4 verifica-se que ao contrário do caso estudado 2, não foi o PCO que apresentou os melhores resultados mas sim o BCO. Este atingiu o valor mínimo, embora tendo sido o mais lento de todas as técnicas. Quanto à robustez, o SCO demonstrou maior robustez uma vez que obteve o menor desvio padrão. Uma vez mais verifica-se que o BCO apesar de ter boa capacidade de busca é um algoritmo lento. Apesar de ter atingido o valor óptimo após 480 avaliações o tempo de computação foi elevado.

5.6.Caso estudado 4

No caso estudado 4, também ele muito difundido na literatura [22], [28], é apresentado um problema com 40 grupos e serve para testar as performances das meta-heurísticas com o aumento de dimensões do problema. Neste caso as funções de custo são não convexas devido ao efeito das válvulas. No seguimento do proposto em [28], não são contempladas perdas, zonas proibidas ou rampas. Como restrições, apresenta apenas os limites mínimos e máximos de cada grupo bem como uma carga de 10500 MW. Os resultados obtidos são posteriormente comparados com [28].

Como foi indicado em [22], uma das limitações dos métodos “tradicionais” é necessitarem de funções de custo contínuas, monotaneamente crescentes. Tal como foi também indicado em [46], uma das maneiras de obviar este problema é aproximar as funções não convexas a convexas através de uma regressão polinomial, tal como apresentado na figura 2-12 e admitir o erro resultante da aproximação. Neste caso estudado, os resultados obtidos pelas meta-heurísticas serão comparados com o método do *lambda iteration* (após as funções de custo não convexas serem aproximadas a funções convexas). Neste caso pretende-se analisar a relação custo/esforço computacional para compreender se face à diferença de custos as meta-heurísticas trazem uma mais-valia relevante.

Tabela 5-6 – Resultados para o caso estudado 4

Heurística	Mínimo	Máximo	Médio	Desvio	t (s)	Avaliações
PSO	131822,3	148516	142402,9	6537,4	24,1	14970
BCO	134001,3	146736,6	141262,9	4634,6	90	14850
CSO	134142,1	149988	143559,8	6609,2	13,5	11700
SCO	129052,3	134778,9	131307,2	2072	4,34	6324
PCO	135492,4	158713,5	145206	9756,3	47,5	15000
<i>Lambda</i>	124156,3	-	-	-	0,15	-

Tabela 5-7 – Valor de produção de cada grupo para o caso estudado 4

Grupo	Limites	PSO	BCO	CSO	SCO	PCO	<i>Lambda</i>
1	[36, 114]	114,00	51,88	36,00	114,00	114,00	114,00
2	[36, 114]	36,00	66,79	36,00	85,23	114,00	114,00
3	[60, 120]	60,00	103,25	60,00	118,19	117,05	120,00
4	[80, 190]	190,00	186,93	83,34	176,39	190,00	190,00
5	[47, 97]	97,00	85,92	97,00	97,00	97,00	97,00
6	[68, 140]	140,00	85,40	94,72	68,69	106,16	140,00
7	[110, 300]	110,00	298,57	147,46	300,00	233,73	300,00
8	[135, 300]	135,00	169,56	135,00	252,60	285,57	300,00
9	[135, 300]	300,00	300,00	300,00	222,24	202,51	300,00
10	[130, 300]	300,00	300,00	281,80	213,22	300,00	130,00
11	[94, 375]	375,00	375,00	272,22	157,02	307,29	94,00
12	[94, 375]	94,00	324,20	272,95	350,69	170,52	94,00
13	[125, 500]	500,00	250,98	439,68	425,77	399,82	125,00
14	[125, 500]	125,00	298,26	404,82	394,47	349,29	271,67
15	[125, 500]	500,00	324,88	500,00	175,94	356,39	266,66
16	[125, 500]	500,00	451,49	500,00	481,11	438,48	266,66
17	[220, 500]	500,00	470,84	363,04	500,00	408,58	500,00
18	[220, 500]	500,00	457,97	500,00	399,66	500,00	500,00
19	[242, 550]	550,00	548,59	545,53	505,33	356,70	550,00
20	[242, 550]	550,00	550,00	485,81	395,01	549,26	550,00
21	[254, 550]	550,00	528,62	531,96	538,36	415,12	550,00
22	[254, 550]	550,00	548,77	550,00	528,31	489,58	550,00
23	[254, 550]	550,00	256,51	550,00	498,19	341,62	550,00
24	[254, 550]	550,00	537,66	436,01	432,82	328,71	550,00
25	[254, 550]	550,00	439,62	550,00	547,44	550,00	550,00
26	[254, 550]	254,00	504,32	550,00	534,47	509,47	550,00
27	[10, 150]	51,53	41,59	68,27	15,18	66,24	10,00
28	[10, 150]	10,00	76,35	10,00	10,00	57,20	10,00
29	[10, 150]	10,00	42,94	50,59	41,38	88,97	10,00
30	[47, 97]	97,00	97,00	97,00	83,49	97,00	97,00
31	[60, 190]	60,00	142,74	170,06	128,94	190,00	190,00
32	[60, 190]	190,00	190,00	79,04	184,37	179,47	190,00
33	[60, 190]	176,47	177,32	190,00	74,51	187,16	190,00
34	[90, 200]	200,00	162,02	166,05	197,60	200,00	200,00
35	[90, 200]	200,00	138,27	176,93	200,00	197,45	200,00
36	[90, 200]	200,00	172,95	200,00	183,43	171,92	200,00
37	[25, 110]	25,00	84,58	47,44	110,00	110,00	110,00
38	[25, 110]	25,00	108,26	54,91	110,00	109,99	110,00
39	[25, 110]	25,00	73,51	79,40	110,00	103,62	110,00
40	[242, 550]	550,00	476,49	386,99	539,78	510,12	550,00
Potência produzida [MW]		10500	10500	10500	10500,83	10499,9	10500

Observando-se os resultados da tabela 5-6 verifica-se que o menor custo foi obtido pelo algoritmo do *lambda iteration* quando as funções não convexas foram aproximadas a convexas. Verifica-se assim que para o caso estudado 4, o uso de um algoritmo como o do *lambda iteration* revela-se mais oportuno em termos de capacidade de busca, robustez (converge sempre para o mínimo) e tempo de convergência.

Capítulo VI

Conclusão e futuros desenvolvimentos

Resumo

Neste capítulo são apresentadas as conclusões desta dissertação e propostos futuros desenvolvimentos.

6. Conclusão e futuros desenvolvimentos

O principal objectivo desta dissertação foi a formulação do problema de despacho económico, com restrições e diferentes funções de custo e a sua solução através de algoritmos baseados em enxames.

Os algoritmos baseados em inteligência de enxames foram aplicados em problemas matemáticos convexos e não convexos conhecidos em literatura como funções de teste. Nestas funções os cinco algoritmos implementados foram capazes de encontrar soluções possíveis, tanto em espaços de busca convexos, como em não convexos. Daqui é possível concluir que os algoritmos são válidos, dado que atingem resultados semelhantes aos encontrados em literatura. Importa concluir que o aumento da complexidade da formulação não implicou alterações à programação.

Um novo modelo híbrido denominado PCO foi proposto e os resultados são apresentados. O PCO demonstra um grande potencial para recolher os benefícios do PSO e os do SCO e assim ser um híbrido com elevada capacidade de busca e velocidade de convergência. O desempenho do híbrido no capítulo quatro e cinco é igual ao previsto e é expectável que o seu potencial permita obter resultados superiores a todos os algoritmos estudados neste trabalho. Pode ver-se que o PCO é melhor ou pelo menos igual ao PSO e SCO nos três critérios de avaliação, logo o híbrido consegue ser melhor ou pelo menos igual que os algoritmos em que se baseia. No entanto, apenas é o melhor dos cinco algoritmos no critério da velocidade de convergência. Isto poderá ser uma consequência do grande número de parâmetros que existem para regular o PCO, o que o tornará mais susceptível aos mesmos. Ou seja, o PCO poderá necessitar de parâmetros diferentes para cada função que otimizar, pelo que será mais difícil afinar estes parâmetros para o PCO operar da melhor forma possível.

Em primeiro lugar, nas funções de teste do capítulo quatro é possível ver que os algoritmos baseados em enxames implementados conseguiram obter soluções possíveis, sendo assim validados face à literatura. Nenhum algoritmo conseguiu obter os melhores resultados em todas as funções, na verdade cada algoritmo revelou-se melhor nalguma função de teste com certo número de dimensões, pelo que não é possível concluir qual o melhor algoritmo. É possível no entanto classificar cada um relativamente à sua velocidade de convergência, robustez e capacidade de busca, tal como feito em 4.2.1. O

SCO e o PCO são geralmente os algoritmos com maior velocidade de convergência, o BCO o mais robusto (embora tenha habitualmente a menor velocidade de convergência) e o que atinge o maior número de vezes o mínimo das soluções.

No capítulo cinco, o *lambda iteration* (método convencional utilizado no caso estudado quatro) foi comparado com os desempenhos dos enxames. O *lambda iteration* obteve um custo de produção cerca de 3,8% mais baixo que o do melhor algoritmo (SCO). O custo que o *lambda iteration* devolve à partida é uma aproximação, pelo que os valores de potência de cada grupo tiveram de ser aplicados à função de custo original e daí se obteve o custo exposto na tabela 5-6. Assim, é de concluir que embora o *lambda* seja um método que não possa ser aplicado directamente a funções de custo não diferenciáveis, o mesmo poderá ainda assim atingir o óptimo dependendo da função em causa.

Este é um trabalho extenso, que ainda deixa algumas áreas por estudar mais aprofundadamente e aqui se enunciam para futuros desenvolvimentos:

- Uma análise da relação entre número de partículas, tempo computacional e mínimo atingido e averiguar se compensa aumentar infinitamente o número de partículas ou, caso não compense, qual o melhor compromisso.
- Análise de sensibilidades, em que são ajustados os parâmetros de cada enxame e analisar qual o impacto de cada um no comportamento final do enxame. Por exemplo, até que valor deve ser aumentado o coeficiente social do PSO? Este aumento será preferível quando o PSO otimizar funções mais restritas ou menos restritas?
- Estudar o desempenho do híbrido criado nesta dissertação noutros casos estudados de outras áreas de conhecimento, tais como em problemas de Mecânica.

7. Bibliografia

- [1] (Higher Education Commission), “Power Economic Dispatch - A Brief Survey,” Pakistan.
- [2] (REN), “Dados técnicos, 2012.”
- [3] J. P. S. Catalão, *Electric Power Systems: Advanced Forecasting Techniques and Optimal Generation Scheduling*, CRC Press. 2012, p. 2012.
- [4] J. de Sousa, “MEN - Folhas de apoio,” pp. 1–18.
- [5] I. Ciornei and E. Kyriakides, “A GA-API Solution for the Economic Dispatch of Generation in Power System Operation,” vol. 27, no. 1, pp. 233–242, 2012.
- [6] J. Driesen and R. Belmans, “Specific Problems of High-Speed Electrical Drives,” pp. 1–16, 2005.
- [7] Z.-L. G. Z.-L. Gaing and T.-C. O. T.-C. Ou, “Dynamic economic dispatch solution using fast evolutionary programming with swarm direction,” *2009 4th IEEE Conf. Ind. Electron. Appl.*, no. 1, pp. 1538–1544, 2009.
- [8] R. D. (University of P. Brasington, “Integration and operation of post-combustion capture system on coal-fired power generation: load following and peak power,” University of Pittsburgh, 2007.
- [9] S. Hemamalini and S. Simon, “Dynamic Economic Dispatch with Valve-Point Effect Using Maclaurin Series Based Lagrangian Method,” vol. 1, no. 17, pp. 60–67, 2010.
- [10] Fonte, P.M.; Monteiro, C.; Maciel Barbosa, F.P., “Sensing Cloud Optimization Applied to a non-convex and constrained dynamic economical dispatch,” *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pp.2163,2168,10-13 Nov. 2013
- [11] J. Eto, “Demand Response Spinning Reserve Demonstration,” 2007.
- [12] S. Faias and R. Pestana, “Folhas de Apoio às Aulas de Laboratório de Redes de Energia Eléctrica,” 2010.
- [13] M. A. Matos, “Introdução ao Trânsito de potências,” 1999.
- [14] J. P. B. Iria, “Trânsito de potência DC difuso com despacho incorporado,” 2011.
- [15] J. P. Sucena Paiva, *Redes de Energia Eléctrica: uma Análise Sistémica*. 2005.

- [16] J. A. Wood and B. F. Wollenberg, “Power, Generation, Operation and Control - 4th Chapter,” 1984.
- [17] V. Mendes, *Programação Matemática e Engenharia Industrial*. 2009.
- [18] O. Alexandrov, “Convex polygon illustration,” [Public domain] via *Wikimedia Commons*. [Online]. Available: https://en.wikipedia.org/wiki/File:Convex_polygon_illustration2.png.
- [19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.
- [20] S. Y. Lim, M. Montakhab, and H. Nouri, “Economic Dispatch of Power System Using Particle Swarm Optimization with Constriction Factor,” vol. 4, no. 2, pp. 29–34, 2009.
- [21] J. A. Wood and B. F. Wollenberg, “Power, Generation, Operation and Control - 2nd Chapter,” 1984.
- [22] Z.-L. Gaing, “Particle swarm optimization to solving the economic dispatch considering the generator constraints,” *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1187–1195, Aug. 2003.
- [23] J. Hazra and A. Sinha, “Application of soft computing methods for Economic Dispatch in Power Systems,” no. 2, pp. 538–543, 2009.
- [24] G. S. Christensen, *Optimal economic operation of electric power systems*. Academic Press, 1979, p. 277.
- [25] B. H. Chowdhury and S. Rahman, “A review of recent advances in economic dispatch,” *IEEE*, vol. 5, no. 4, pp. 1248–1259, 1990.
- [26] J. A. Wood and B. F. Wollenberg, “Power, Generation, Operation and Control - 3rd Chapter,” 1984.
- [27] D. Yu, Y. Wang, and R. Guo, “A Hybrid Ant Colony Optimization Algorithm Based Lambda-Iteration Method for Unit Commitment Problem,” *2010 Second WRI Glob. Congr. Intell. Syst.*, no. 1, pp. 19–22, Dec. 2010.
- [28] J. Park, Y. Jeong, J. Shin, S. Member, and K. Y. Lee, “An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems,” *Power*, vol. 25, no. 1, pp. 156–166, 2010.
- [29] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, “Evolutionary Programming Techniques for Economic Load Dispatch,” 2003.
- [30] N. Sinha, “PSO embedded evolutionary programming technique for nonconvex economic load dispatch,” *Power Syst. Conf.*, 2004.

-
- [31] H. Ahmed and J. Glasgow, “Swarm Intelligence: Concepts, Models and Applications,” no. February, 2012.
- [32] D. W. Chambers, *Swarm intelligence.*, vol. 78, no. 2. 2011, pp. 34–43.
- [33] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proc. ICNN’95 - Int. Conf. Neural Networks*, vol. 4, pp. 1942–1948, 1995.
- [34] A. B. D. S. Serapião, “Fundamentos de otimização por inteligência de enxames: uma visão geral,” *Sba Control. Automação Soc. Bras. Autom.*, vol. 20, no. 3, pp. 271–304, 2009.
- [35] N. A. Al-madi and A. T. Khader, “De Jong ’ s Sphere Model Test for A Social-Based Genetic Algorithm (SBGA),” vol. 8, no. 3, pp. 179–185, 2008.
- [36] D. Merkle and M. Middendorf, “Swarm Intelligence,” 2001, pp. 401–435.
- [37] C. Zhaohui, “Cockroach Swarm Optimization,” *2010 2nd Int. Conf. Comput. Eng. Technol.*, pp. V6–652–V6–655, 2010.
- [38] T. C. Havens, S. Member, N. G. Salmon, and J. M. Keller, “Roach Infestation Optimization,” 2008.
- [39] J.-M. Amé, J. Halloy, C. Rivault, C. Detrain, and J. L. Deneubourg, “Collegial decision making based on social amplification leads to optimal group formation,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 15, pp. 5835–40, Apr. 2006.
- [40] HowStuffWorks, “The Cockroach Life Cycle and Behavior.” [Online]. Available: <http://science.howstuffworks.com/zoology/insects-arachnids/cockroach2.htm>. [Accessed: 19-Sep-2013].
- [41] P. M. Fonte, C. Monteiro, F. M. Barbosa, and R. R. Frias, “SCO – Sensing Cloud Optimization A Novel optimization technique.”
- [42] E. Elbeltagi, T. Hegazy, and D. Grierson, “Comparison among five evolutionary-based optimization algorithms,” *Adv. Eng. Informatics*, vol. 19, no. 1, pp. 43–53, Jan. 2005.
- [43] A. Štefek, “Benchmarking of Heuristic Optimization Methods.”
- [44] F. M. B. P. M. Fonte, Claudio Monteiro, “Sensing Cloud Optimization to solve ED of Units with valve-point effects and multi-fuels,” no. 1.
- [45] a. I. Selvakumar and K. Thanushkodi, “A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems,” *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 42–51, Feb. 2007.

- [46] C.-L. Chiang, “Improved Genetic Algorithm for Power Economic Dispatch of Units With Valve-Point Effects and Multiple Fuels,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1690–1699, Nov. 2005.

Publicações (no livro de abstracts da CETC – Conference on Electronics, Telecommunications and Computers)

Solving non-Convex and Restricted Problems using Swarms *Economic Dispatch Case*

J. Pontes¹, P. M. Fonte¹, R. Pestana^{1,2}

Instituto Superior de Engenharia de Lisboa (ISEL),

¹Área Departamental de Engenharia de Sistema de Potência e Automação, ISEL, Lisboa, Portugal

²REN – Redes Energéticas Nacionais, Portugal

jpontespontes@gmail.com

pfonte@deea.isel.ipl.pt

rui.pestana@ren.pt

Keywords: Heuristics, Swarm Intelligence, non-Convex Economic Dispatch Problem.

Abstract: In this paper three heuristic techniques based in swarm intelligence are studied, namely its ability to solve non-convex, non-differentiable and highly restricted optimization problems. The performances of Particle Swarm Optimization, Bee Colony Optimization and Cockroach Swarm Optimization solving a set of benchmark functions will be compared. These techniques will also be applied to an electrical engineering problem, to be precise the economic dispatch with non-convex cost functions. The results obtained up to now, have been demonstrating that these techniques are able to reach good results in the benchmark functions as well as in the problem of economic dispatch.

1 INTRODUCTION

Optimization or mathematical programming algorithms are search methods with the purpose to find the best solution to an optimization problem subject to a set of constraints. Although this definition is simple it can hide a number of complex issues, as nonlinear boundaries which may restrict the search area, the search space can be involved with many possible solutions, the characteristics of the problem may change over time, or the quantity being optimized may have conflicting objectives. In all of these situations we seek an optimal solution, or at least an acceptable solution by the most efficient and effective manner [1].

2 SWARM INTELLIGENCE

In a very generic way, swarm intelligence can be characterized as the collective behavior of decentralize and self-organized multi-agent system, displaying some sort of intelligence. No individual takes the role of commander of the swarm and every one influences with greater or lesser weight the course of the swarm. Solutions are emerging, instead of pre-established which means that the solutions are

obtained iteratively, with different inputs and surrounding environment in constant change.

2.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is based in bird flock behavior searching for food. Each individual is considered a particle, which represent possible solutions to the given problem. Each particle interacts with all others to search a better solution using three parameters, namely, its inertia and cognitive and social behaviors. Cognitive behavior is the memory of the previous particle's position. The social behavior connects the entire swarm by remembering its best position so far. Inertial behavior sets limits to the change in velocity of each particle [2].

2.2 Bee Colony Optimization

Bee Colony Optimization (BCO) is characterized by mimicking bee's behavior in search for flowers. BCO creates three different bees: scout, followers and onlookers. The first phase sets the scouts bees in search of flowers in a random path. Next, the scouts return to the hive and present their best found flower to the followers, which will decide probabilistically

what flower to fly to. The final phase is when no improvement happens and onlooker bees set out randomly to another flower [2].

2.3 Cockroach Swarm Optimization

Cockroach Swarm Optimization (CSO) is inspired by the cockroach's social behavior. Its mathematical formulation intends to imitate the behaviors of cockroach swarms as, chase swarming, dispersing and ruthless. These agents can communicate with each other and feel the environment's changes, dispersing randomly when after a certain number of iterations no improvement has been reached[3].

3 SWARM BEHAVIOUR

To demonstrate the performances of swarms a set of benchmark functions are tested, (1) to (5). The success rate is measured by the average of reached minimum values whereas the robustness is measured by the standard deviation of the results [2].

The *Sphere* function (1) is a continuous, convex and unimodal without local minimum except the global one where $f(x_i) = 0$ with $x_i = 0$. The *Rosenbrock* function (2) is non-convex with a global minimum inside a long, narrow and parabolic shaped flat valley. To find the valley is trivial but to converge to the global minimum is not, even being unique and function unimodal. The global minimum is $f(x_i) = 0$ at $x_i = 1$. The *Schaffer* function (3) is bidimensional and multimodal presenting many local optima. The minimum is $f(x_i) = 0$ at $(x_1, x_2) = (0, 0)$, where. The *Griewank* function (4) is widely used to test the optimization algorithms. In a range of $[-600, 600]$ has 191 local minima, with the global minima at $x_i = 0$ where $f(x_i) = 0$. The *Rastrigin* function (5) is a multimodal where the number of local minimum increases exponentially. There is a minimum at $x_i = 0$ where $f(x_i) = 0$.

$$f_1(\vec{x}) = \sum_{i=1}^n x_i^2 \quad (1)$$

$$f_2(\vec{x}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \quad (2)$$

$$f_3(\vec{x}) = 0.5 + \frac{\text{sen}^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} \quad (3)$$

$$f_4(\vec{x}) = \frac{1}{4000} \left(\sum_{i=1}^n (x_i - 100)^2 \right) - \left(\prod_{i=1}^n \left(\frac{x_i - 100}{\sqrt{i}} \right) \right) \quad (4)$$

$$f_5(\vec{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (5)$$

4 ECONOMIC DISPATCH

The Economical Dispatch (ED) problem is an important issue in the power system operation. Fundamentally it is intended to evaluate the value that each on-line power unit should generate with the lowest cost respecting the technical and load demand constrains. Its formulation is shown from (6) to (12) [4][5].

$$\min \sum_{i=1}^{N_G} c_i P_i^2 + b_i P + a_i + |\theta_i \cdot \sin[f_i \cdot (P_{i,\min} - P_i)]| \quad (6)$$

$$\text{s.t.} \quad \sum_{i=1}^{N_G} P_i = P_L + P_{Loss} \quad (7)$$

$$\max(P_i^{\min}, P_i^0 - DR_i) \leq P_i \leq \max(P_i^{\max}, P_i^0 + UR_i) \quad (8)$$

$$P_{Loss} = \sum_{i=1}^{N_G} \sum_{j=1}^{N_G} P_i B_{ij} P_j + \sum_{i=1}^{N_G} B_{oi} P_i + B_{oo} \quad (9)$$

$$\begin{cases} P_i^{\min} \leq P_i(t) \leq P_{i,1}^{LB} \\ P_{i,j}^{LB} \leq P_i(t) \leq P_{i,j}^{UB}, \quad j = 2, 3, \dots, NP_i \\ P_{i, NP_i}^{UB} \leq P_i(t) \leq P_i^{\max} \end{cases} \quad (10)$$

$$SR_i^{\max} \leq (1 - x) \times P_i^{\max} \quad (11)$$

$$\sum_{i=1}^{N_G} [\min(P_i^{\max} - P_i, SR_i^{\max})] \geq SR_i \quad (12)$$

REFERENCES

- [1] Englebretch, A. P., 2007. *Computational Intelligence - An Introduction*, John Wiley & Sons, Inc., LTD.
- [2] Serapião, A, 2009 Fundamentos de otimização por inteligência de enxames: uma visão geral (in portuguese), *Revista Controle & Automação*, Vol. 20, No. 3, p. 271-304.
- [3] ZhaoHui, C. e HaiYan, T., 2010. Cockroach Swarm Optimization, in *Proc.of ICET'10 - 2nd International Conference on Computer Engineering and Technology*, Chengdu, China, Apr. 2010, p. 652-655.
- [4] P. M. Fonte *et al*, 2013. Sensing Cloud Optimization to solve ED of Units with valve-point effects and multi-fuels, in *Proc. of DoCEIS'13 - Doctoral Conference on Computing, Electrical and Industrial Systems*, Costa da Caparica, Portugal, Apr. 2013, p 477-484.
- [5] Ciomei, I., Kyriakides, E., 2012. A GA-API Solution for the Economic Dispatch of Generation in Power System Operation, *IEEE Transaction on Power Systems*, Vol. 27, No. 1, p. 233-242.

Publicações (na conferência CETC – Conference on Electronics, Telecommunications and Computers)



SOLVING NON-CONVEX AND RESTRICTED PROBLEMS USING SWARMS

Economic Dispatch Case

J. Pontes¹, P. M. Fonte¹, R. Pestana^{1,2}

¹Área Departamental de Engenharia de Sistema de Potência e Automação, ISEL, Lisboa, Portugal
²REN – Redes Energéticas Nacionais, Portugal



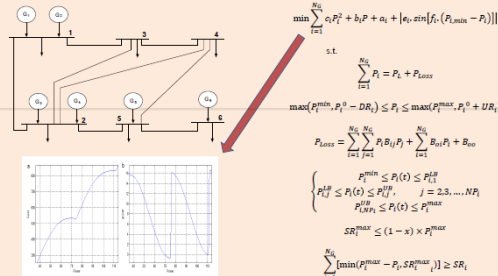
Introduction

Three heuristic techniques based in swarm intelligence are studied, namely its ability to solve non-convex, non-differentiable and highly restricted optimization problems. These techniques were applied to an electrical engineering problem, namely, the economic dispatch with non-convex cost functions.

Commercial tools that are able to solve economical dispatch for thermal units always required convex cost function. This can be because of limitation of the optimizing tool or the need to perform fast and non-convex algorithms tend to be slow.

Economic dispatch

The Economical Dispatch (ED) problem is an important issue in the power system operation. Fundamentally it is intended to evaluate the value that each on-line power unit should generate with the lowest cost respecting the technical and load demand constrains



Keywords: Heuristics; Swarm Intelligence; non-Convex Economic Dispatch Problem.

Swarm intelligence

- Collective behavior
- Decentralized
- Self-organized multi-agent system
- Present some kind of intelligence
- No individual takes the role of commander
- Every particle (element) influences with greater or lesser weight the behavior of the swarm

Particle swarm optimization (PSO)

- Based in bird flock behavior searching for food.
- Each bird (particle) interacts with each other
- Presents cognitive and social behavior and inertia
- Cognitive behavior gives a weight to the best fitness found by each particle (c_1)
- Social behavior gives a weight to the particle with the best fitness of the entire swarm (c_2)
- Inertia sets limits to the velocity of the particles (ω)



$v_i(t+1)$ is the velocity set to particle i and $x_i(t+1)$ its new position

$$v_i(t+1) = \omega \times v_i(t) + c_1 \times (x_{best} - x_i(t)) + c_2 \times (G_{best} - x_i(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Bee Colony Optimization (BCO)

- Mimicking bee's behavior in search for flowers
- Three different bees: scout, followers and onlookers
- First phase sets the scouts bees in search of flowers in a random path
- The scouts return to the hive and present their best found flower to the followers
- Followers decide probabilistically what flower to fly to
- Without any improvement onlooker bees set out randomly to another flower

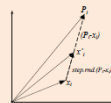
P_i is the probability of each scout's flower to be followed by the followers and v_{ij} the position of the follower in the neighborhood of the scout's position.

$$P_i = \frac{fit(x_i)}{\sum_{n=1}^{SN} fit(x_n)}$$

$$v_{ij} = x_{ij} + \psi_{ij} \times (x_{ij} - x_{kj})$$

Cockroach Swarm Optimization (CSO)

- Inspired by the cockroach's social behavior
- Can communicate with each other, dispersing randomly after a certain number of iterations
- Present chase and ruthless



- $x'(t)$ is the new position and $P(t)$ is the position of the best particle within visual scope

$$x'(t) = x(t) + step.rand.(P(t) - x(t)), \quad x(t) \neq P(t)$$

$$x'(t) = x(t) + step.rand.(P_g - x(t)), \quad x(t) = P(t)$$

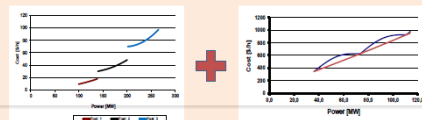
Benchmark functions

Function	Mathematical expression	Illustration
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	
Booth	$f_2(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	
Schaffer	$f_3(x) = 0.5 + \frac{0.5 \sin(\sqrt{0.5(x_1^2 + x_2^2)})}{1 + 0.001(x_1^2 + x_2^2)}$	
Crossin	$f_4(x) = \frac{1}{2000} \left(\sum_{i=1}^n x_i - 100 \right)^2 - \left(\prod_{i=1}^n \left(\frac{x_i - 100}{\sqrt{ x_i }} \right) \right) + 1$	
Beasings	$f_5(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	

Case study - Multi-fuels and valve-point effect

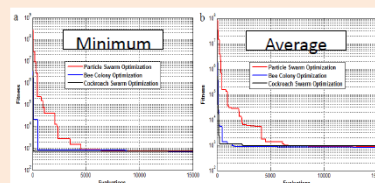
- 10 thermal units
- 2700 MW
- Valve-point effects
- Multi-fuels:
- 30 Particles
- 100 generations

$$F_i(P_i) = \begin{cases} a_i + b_i P_i + c_i P_i^2 + |e_i \times \sin(f_i \times (P_i^{\max} - P_i))| & \text{for fuel 1, } P_i^{\min} \leq P_i \leq P_i^{\max} \\ a_{i2} + b_{i2} P_i + c_{i2} P_i^2 + |e_{i2} \times \sin(f_{i2} \times (P_i^{\max} - P_i))| & \text{for fuel 2, } P_i^{\min} < P_i \leq P_i^{\max} \\ \vdots \\ a_{ik} + b_{ik} P_i + c_{ik} P_i^2 + |e_{ik} \times \sin(f_{ik} \times (P_i^{\max} - P_i))| & \text{for fuel k, } P_{i,k} < P_i \leq P_i^{\max} \end{cases}$$



To solved by commercial 59049 possibilities (a large amount non feasible)

Heuristic	Min.	Max.	Avg.	Std.	t(s)	Eval.
PSO	648,13	1596,18	793,65	156,84	23,9	14640
BCO	667,01	1318,54	803,09	130,91	80,8	6480
CSO	654,63	1594,88	971,18	278,89	9,8	1860



Conclusions

- There isn't an "ideal" heuristic which can solve any kind of problems
- All of the heuristics tested reached acceptable values of minimum cost function
- The results obtained up to now, have been demonstrating that these techniques are able to reach good results in the benchmark functions as well as in the problem of economic dispatch
- It was shown that this kind of heuristics can fill the gaps of commercial software packages used in power systems.

Anexo 1 – Pseudo-código do Particle Swarm Optimization

while (stop condition isn't reached)

for each particle

Initialize randomly the position of particle

Restrain position of particle to the problem's limits

Set initial velocity to particle

Evaluate fitness of particle, $f(x_{best})$

Evaluate fitness of new particle obtained by velocity, $f(x)$

end

for each particle

If $f(x) < f(x_{best})$

Update best local position, x_{best}

end

If $f(x_{best}) < f(g_{best})$

Update best global position, g_{best}

end

for each particle

Update velocity of particles in x

Obtain new position of particles in x

end

end

Anexo 2 – Pseudo-código do Bee Colony Optimization

while (stop condition isn't reached)

for each particle

 Initialize the size of the hive (SH)

 Initialize half of SH to random positions and name this half as scouts

 Restrain position of scouts to the problem's limits

 Evaluate fitness of scouts, $fit(x_i)$

 Determine the probability of the followers going to search around each

 scout

 Evaluate fitness of new positions obtained by the followers, $fit(x'_i)$

end

for each particle

 If $fit(x'_i) < fit(x_i)$

 Update scout position to x'_i

end

 If $fit(x_i) < fit_{best}$

 Update best global position, $best$

end

for each particle

 If $attempts > attempts_{max}$

 Reinitialize randomly scout's position

end

end

Anexo 3 – Pseudo-código do Cockroach Swarm Optimization

while (stop condition isn't reached)

for each particle

Initialize randomly the position of particle

Restrain position of particle to the problem's limits

Evaluate fitness of particle, $f(x_i)$

Compute whether there is any particle better within visual distance

If $f(x_i) > f(\text{visual})$

Search for $P(i)$

Else if $f(x_i) > f(P_g)$

Search for P_g

End

Evaluate fitness of new particle obtained, $f(x_i)$

end

for each particle

If $f(x_i) < f(P_g)$

Update best local position, P_g

end

If $f(x_i) < f(P_i)$

Update best global position, P_i

end

If iterations == disperse_iteration

for preset number of particles

Reinitialize random particle different of P_g

If $f(x_i') < f(x_i)$

Update x_i to x_i'

end

end

end

If variance < preset limit

Erase a particle different of P_g

end

end

Anexo 4 – Fluxograma do Sensing Cloud Optimization

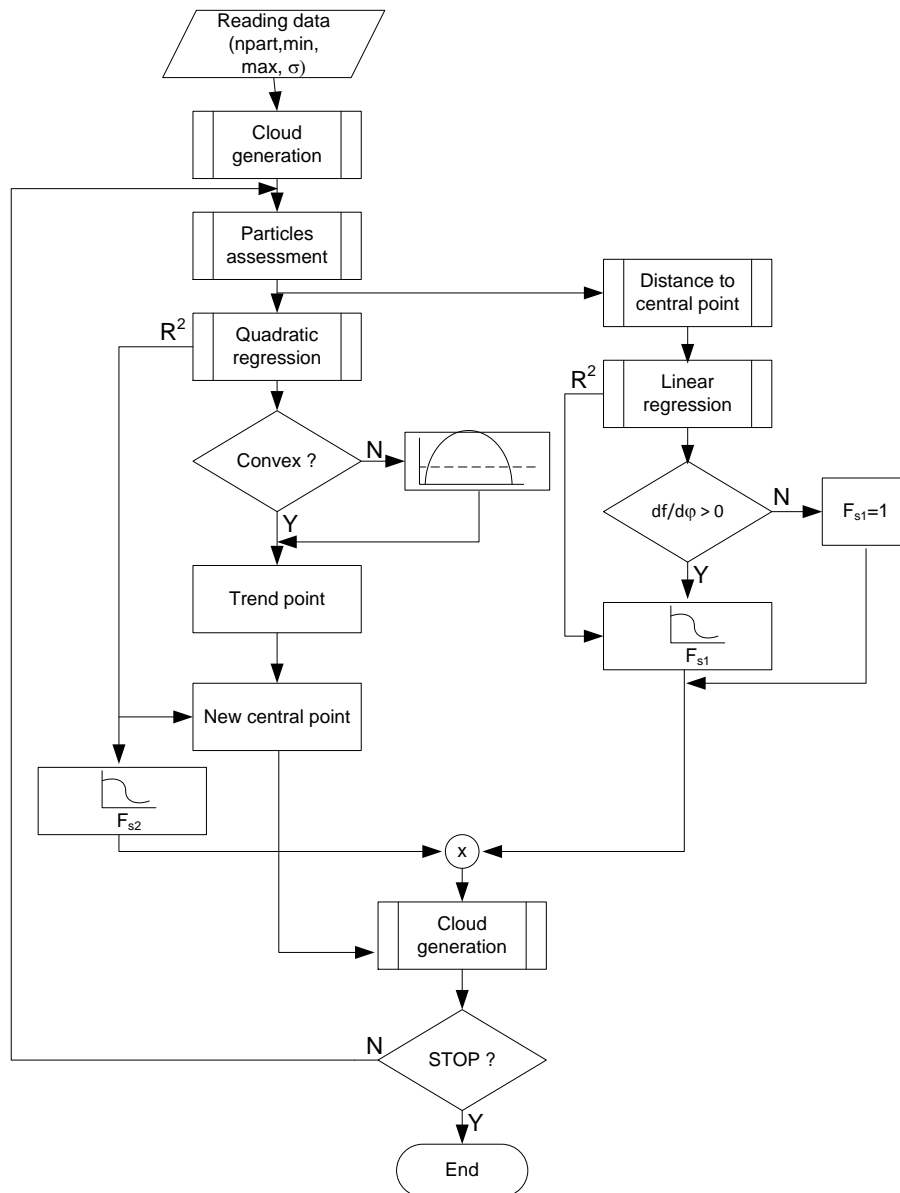


Figura 7-1 – Fluxograma do SCO [41].

Anexo 5 – Pseudo-código do Particle Cloud Optimization

while (stop condition isn't reached)

if number of iterations not improving < number of iterations to enter PSO

run SCO

else

swarm = *cloud*;

run PSO

end

end

Anexo 6 – Evolução de cada algoritmo para a função *De Jong* com 5 dimensões

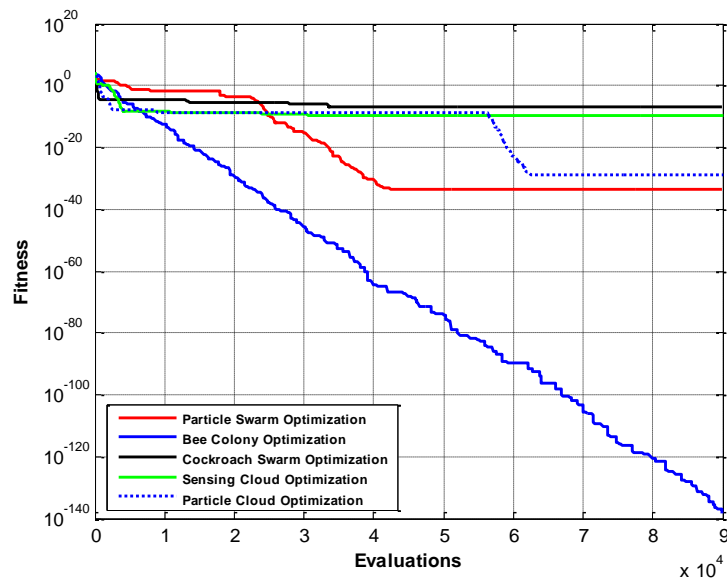


Figura 7-2 – Evolução da melhor simulação para a função De Jong com 5 dimensões.

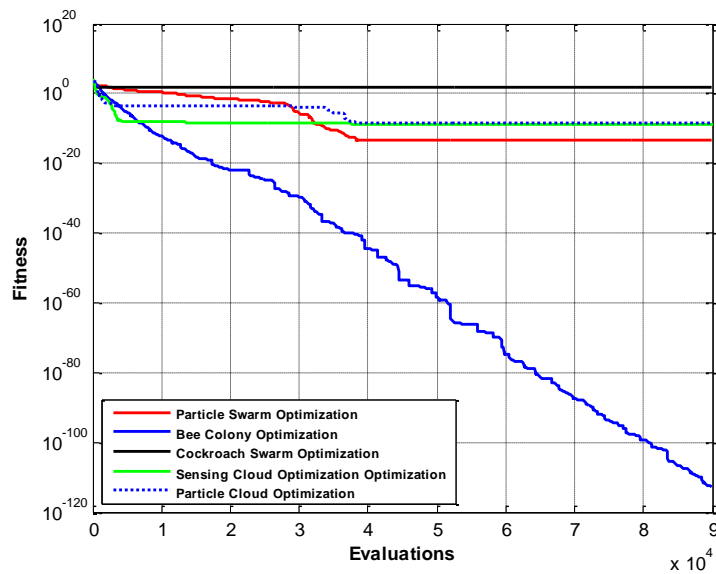


Figura 7-3 – Evolução média para a função De Jong com 5 dimensões.

Anexo 7 – Evolução de cada algoritmo para a função *De Jong* com 10 dimensões

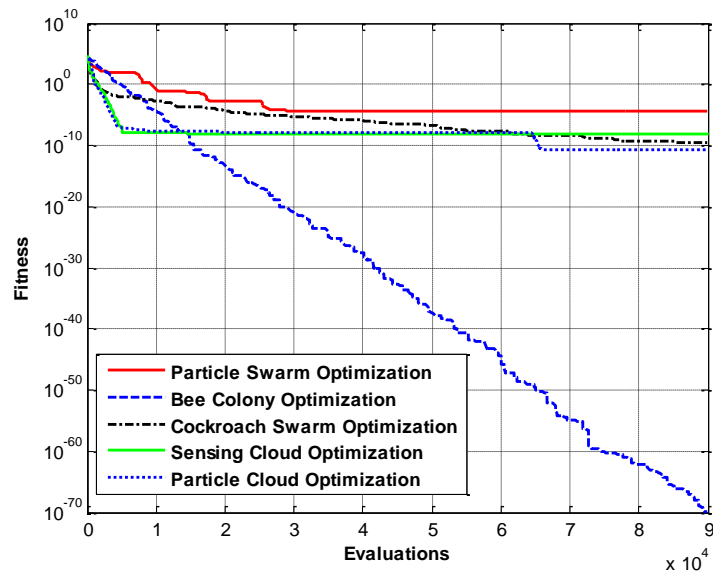


Figura 7-4 – Evolução da melhor simulação para a função De Jong com 10 dimensões.

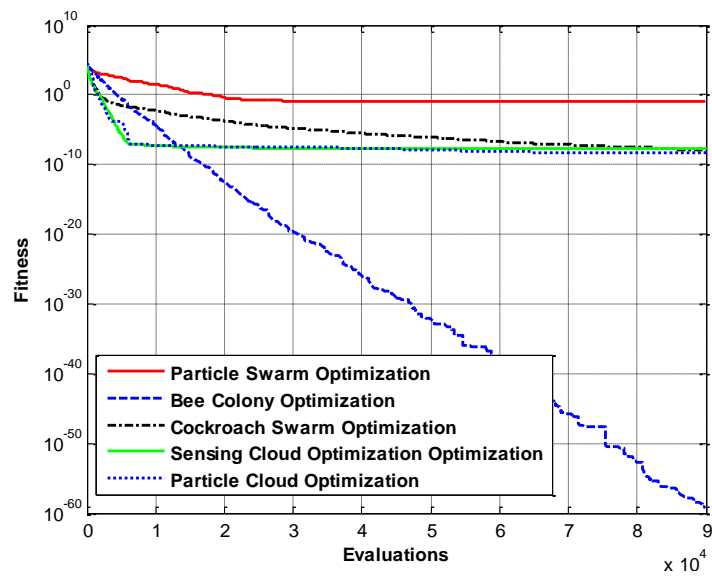


Figura 7-5 – Evolução média para a função De Jong com 10 dimensões.

Anexo 8 – Evolução de cada algoritmo para a função *Rosenbrock* com 5 dimensões

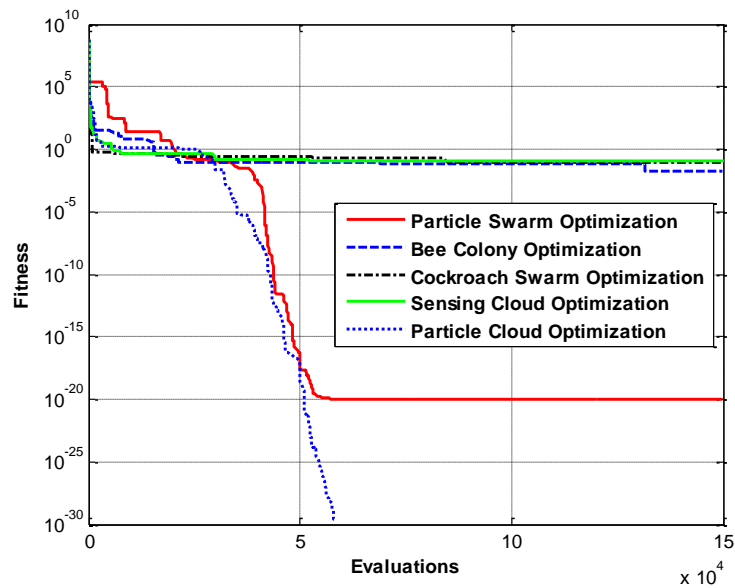


Figura 7-6 – Evolução da melhor simulação para a função *Rosenbrock* com 5 dimensões.

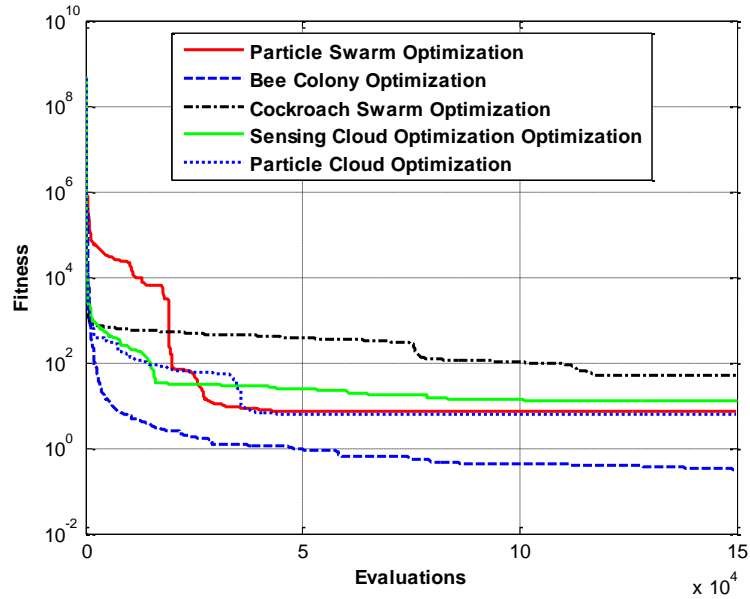


Figura 7-7 – Evolução média para a função *Rosenbrock* com 5 dimensões.

Anexo 9 – Evolução de cada algoritmo para a função *Rosenbrock* com 10 dimensões

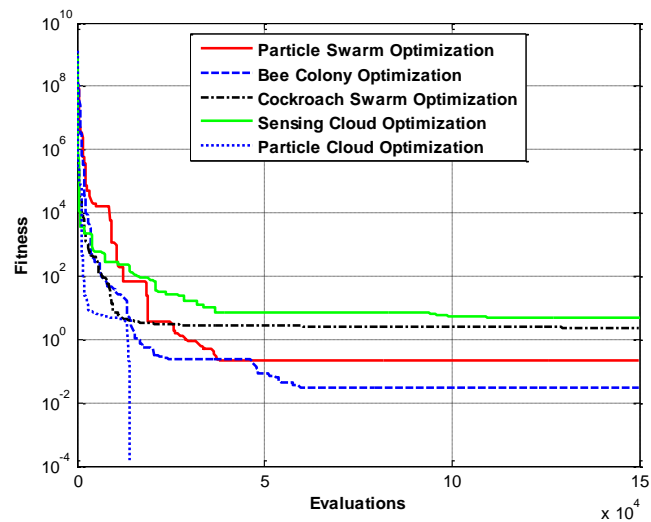


Figura 7-8 – Evolução da melhor simulação para a função Rosenbrock com 10 dimensões.

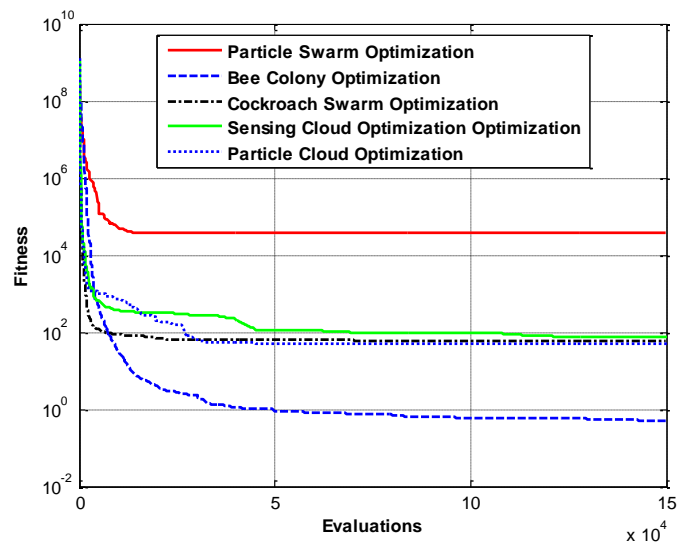


Figura 7-9 – Evolução média para a função Rosenbrock com 10 dimensões.

Anexo 10 – Evolução de cada algoritmo para a função *Schaffer* com 2 dimensões

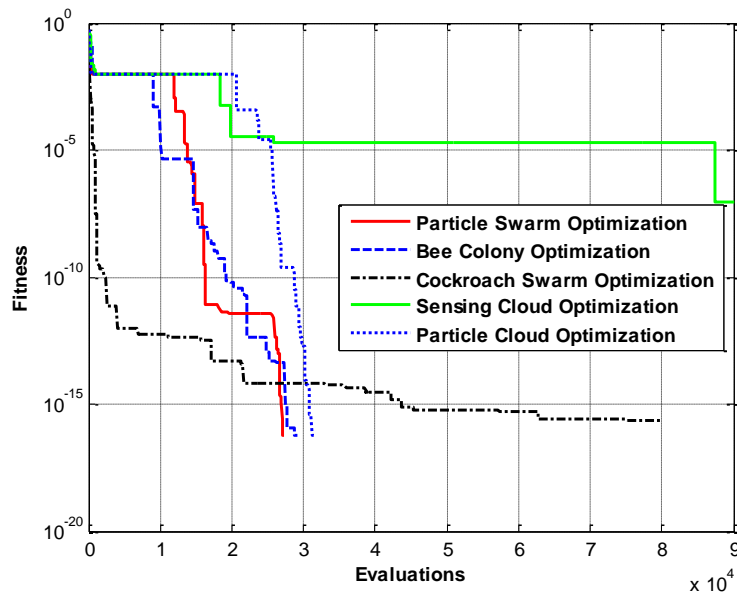


Figura 7-10 – Evolução da melhor simulação para a função Schaffer com 2 dimensões.

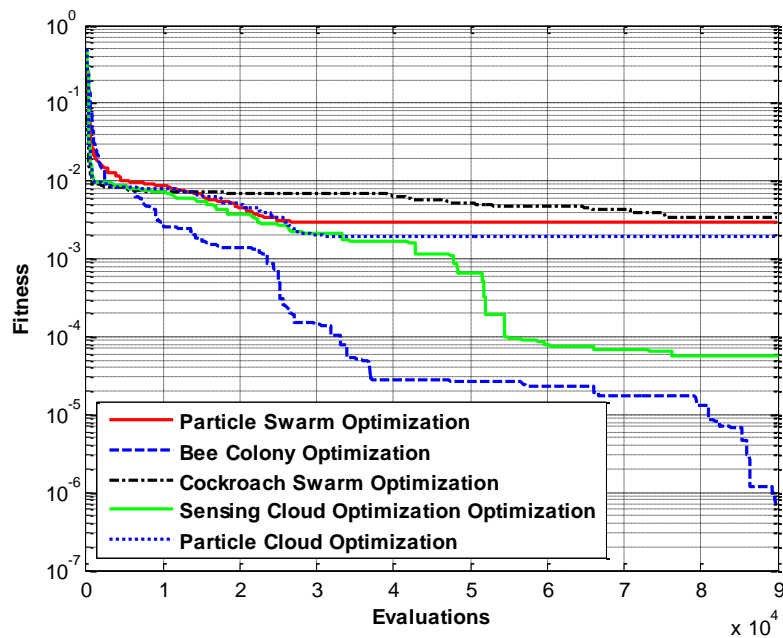


Figura 7-11 – Evolução média para a função Schaffer com 2 dimensões.

Anexo 11 – Evolução de cada algoritmo para a função *Griewank* com 5 dimensões

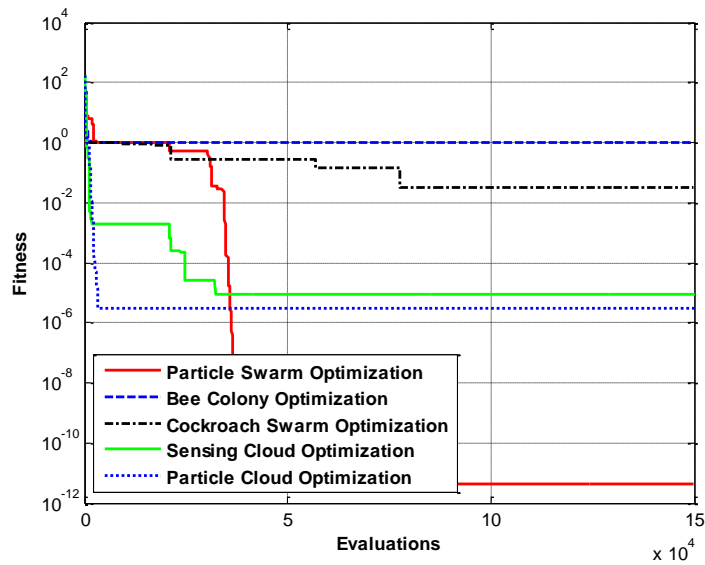


Figura 7-12 – Evolução da melhor simulação para a função Griewank com 5 dimensões.

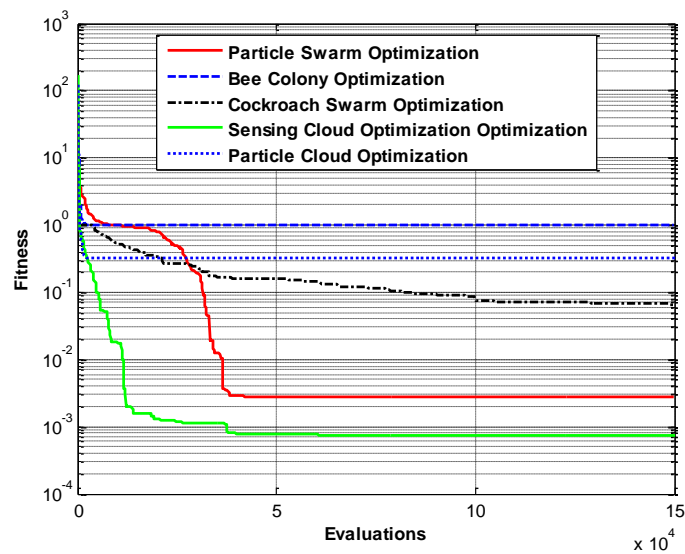


Figura 7-13 – Evolução média para a função Griewank com 5 dimensões.

Anexo 12 – Evolução de cada algoritmo para a função *Griewank* com 10 dimensões

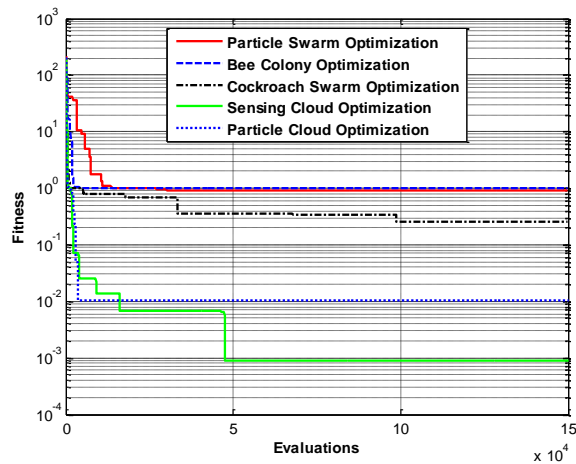


Figura 7-14 – Evolução da melhor simulação para a função Griewank com 10 dimensões.

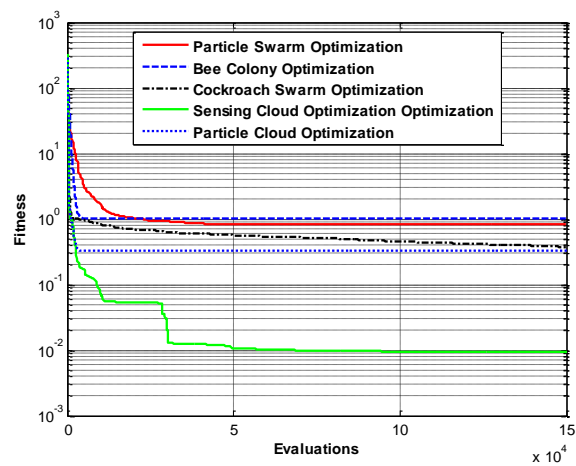


Figura 7-15 – Evolução média para a função Griewank com 10 dimensões.

Anexo 13 – Evolução de cada algoritmo para a função *Rastrigin* com 5 dimensões

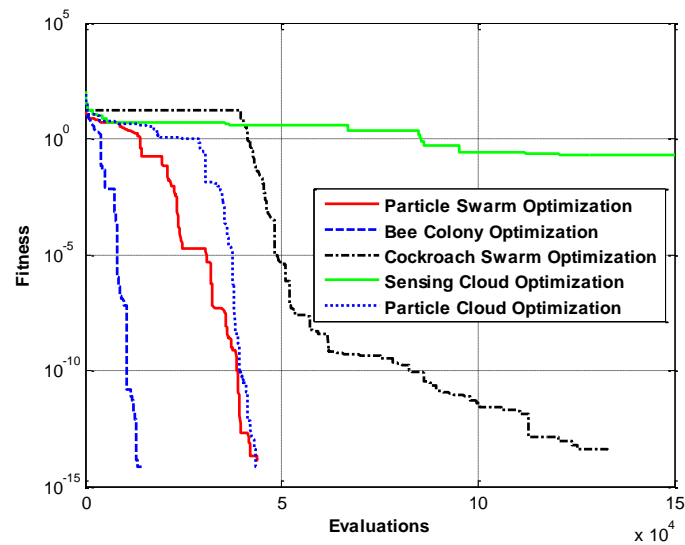


Figura 7-16 – Evolução da melhor simulação para a função Rastrigin com 5 dimensões.

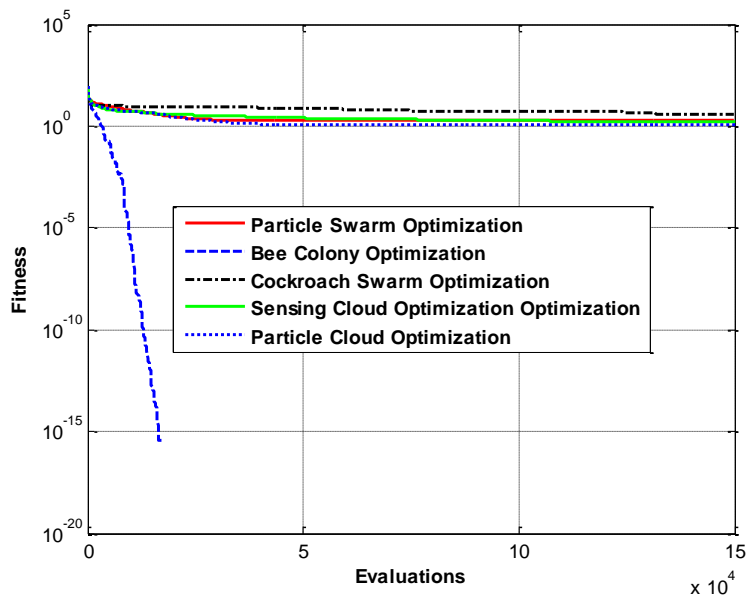


Figura 7-17 – Evolução média para a função Rastrigin com 5 dimensões.

Anexo 14 – Evolução de cada algoritmo para a função *Rastrigin* com 10 dimensões

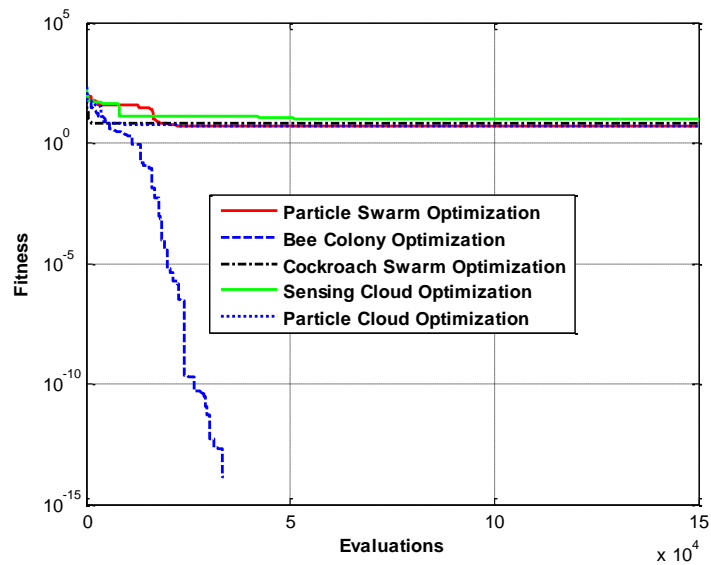


Figura 7-18 – Evolução da melhor simulação para a função Rastrigin com 10 dimensões.

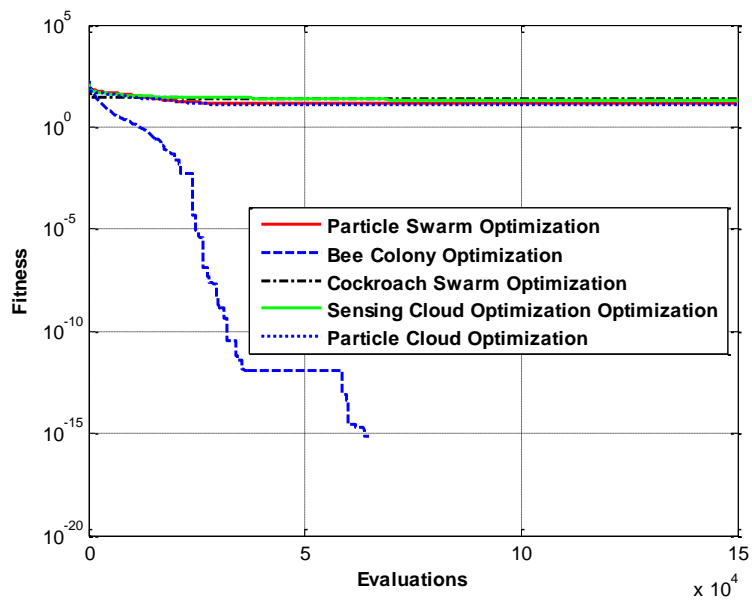


Figura 7-19 – Evolução média para a função Rastrigin com 10 dimensões.

Anexo 15 – Evolução da potência por grupo para o caso estudado 1 com o PSO, BCO, CSO, SCO e PCO

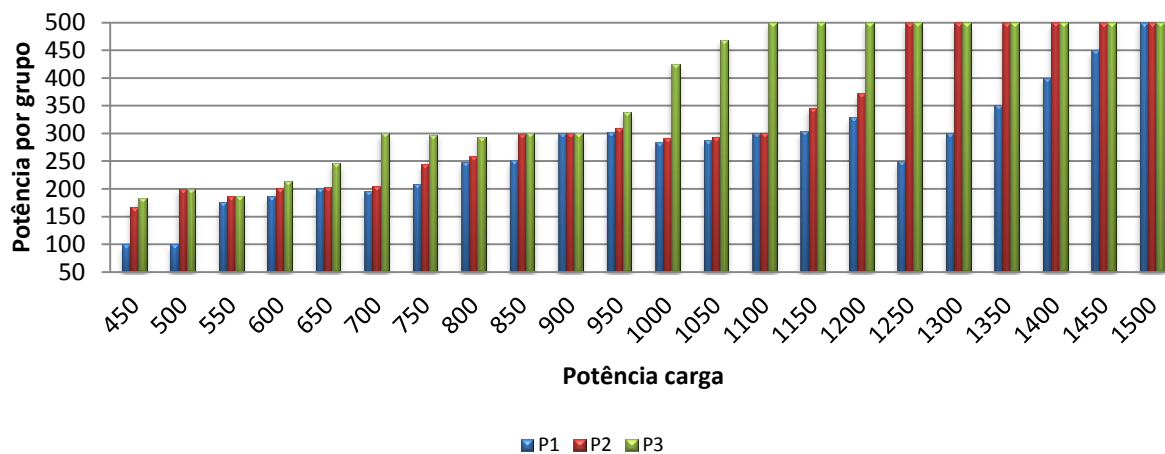


Figura 7-20 – Caso estudado 1 com PSO.

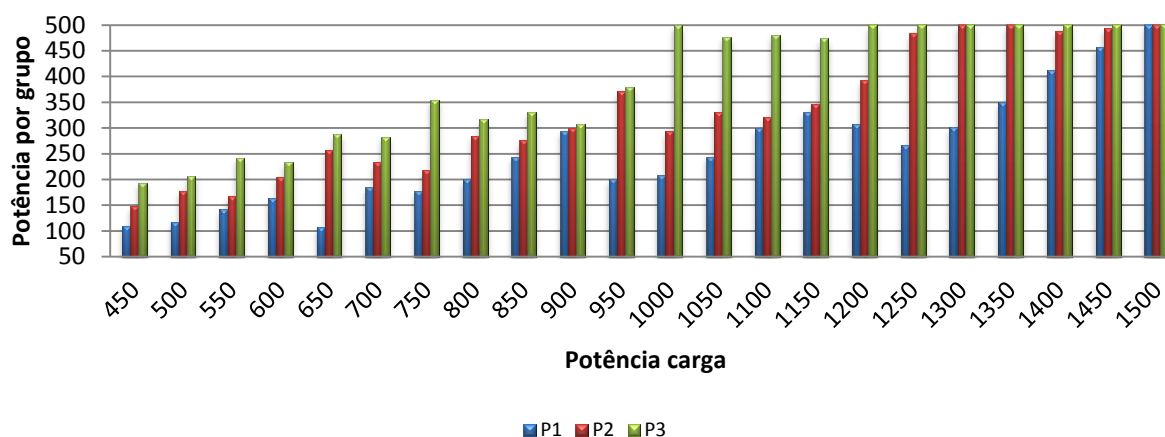


Figura 7-21 – Caso estudado 1 com BCO.

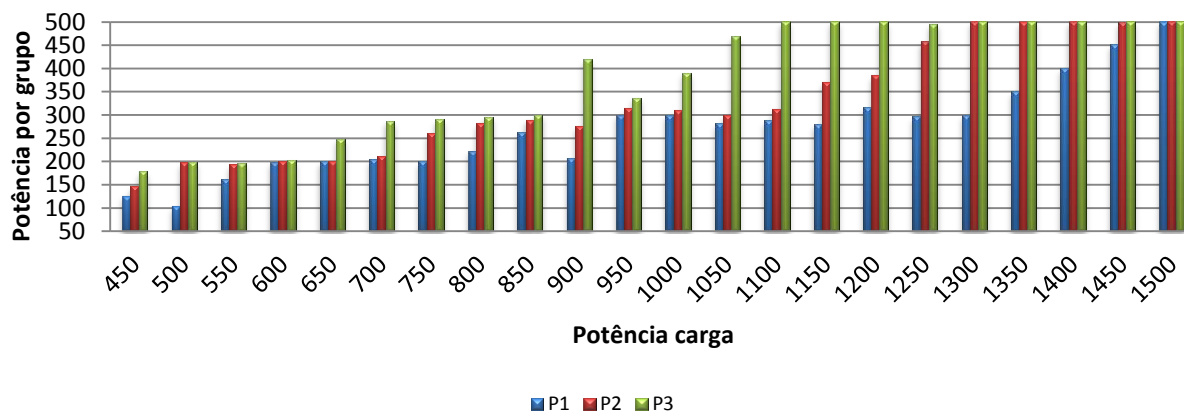


Figura 7-22 – Caso estudado 1 com CSO.

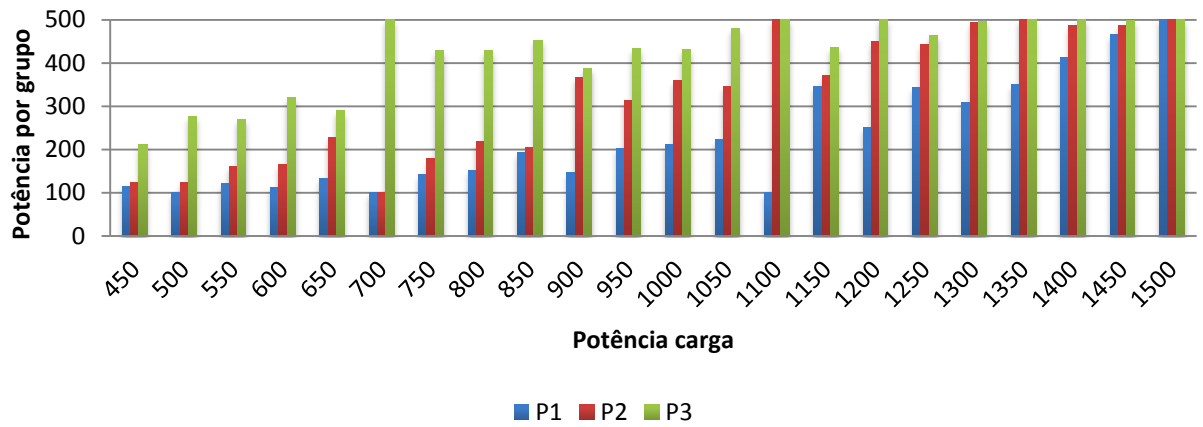


Figura 7-23 – Caso estudado 1 com SCO.

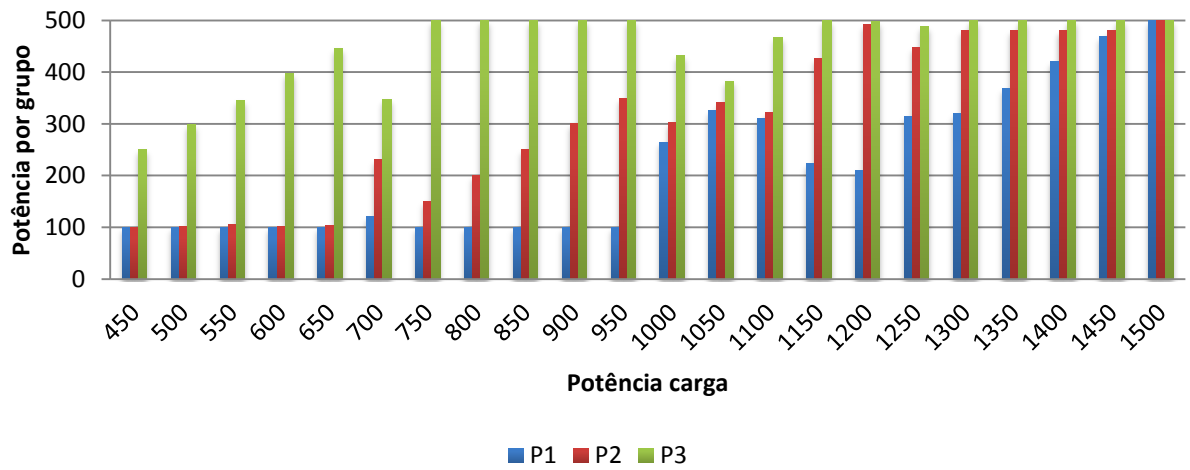


Figura 7-24 – Caso estudado 1 com PCO.

Anexo 16 – Evolução da potência por grupo para o caso estudado 1 com todos os algoritmos

