

Aplicação da metaheurística DEEPSO a problemas de otimização global em larga escala

Carolina Gil Marcelino

Centro Federal de Educação Tecnológica de Minas Gerais
Laboratório de Sistemas Inteligentes
Belo Horizonte, MG, 7675, Brazil
carolina@lsi.cefetmg.br

Paulo Eduardo Maciel de Almeida, Elizabeth Fialho Wanner

Centro Federal de Educação Tecnológica de Minas Gerais
Departamento de Computação
Belo Horizonte, MG, 7675, Brazil
pema@lsi.cefetmg.br, efwanner@decom.cefetmg.br

Leonel Magalhães de Carvalho, Vladimiro H. B. P. de Miranda

Institute for Systems and Computer Engineering of Porto
Centre for Power and Energy Systems, Porto, Portugal
{lcarvalho, vmiranda}@inesctec.pt

RESUMO

Neste trabalho, uma metaheurística denominada DEEPSO, é proposta para resolver problemas de otimização global em larga escala. DEEPSO pode ser visto como um algoritmo evolutivo composto das regras de recombinação emprestadas do PSO, ou um método de otimização por enxame de partículas com propriedades de seleção e autoadaptabilidade. Para avaliar seu desempenho, o algoritmo DEEPSO foi submetido a 15 problemas de *benchmark* de característica contínua, propostos na competição LSGO do CEC'2015. Os experimentos realizados mostram que DEEPSO possui melhores resultados, quando comparado com o algoritmo de referência DECC-G da referida competição.

PALAVRAS CHAVE. Methaeurísticas. DEEPSO. Inteligência Computacional.

Área Principal: (Metaheurísticas, Outras aplicações em PO)

ABSTRACT

In this paper, a metaheuristic called DEEPSO, is proposed to solve large-scale optimization problems. DEEPSO can be viewed as an evolutionary algorithm with recombination rules borrowed from PSO, or a swarm optimization method with selection and self-adaptiveness properties. To assess its the algorithm performance, the algorithm is run over continuous 15 benchmark problems presented in LSGO competition of CEC'2015. The experimental results show that the approach reaches better results when compared to the reference algorithm DECC-G from that competition.

KEYWORDS. Methaeuristics. DEEPSO. Computational Intelligence.

Main Area: Methaeuristics and Other applications in PO

1. Introdução

Problemas de otimização em larga escala podem ser facilmente encontrados em inúmeras aplicações práticas, tais como controle industrial, biomedicina, sistemas aeroespaciais e de logística. Tipicamente, estes problemas são difíceis de resolver devido à dificuldade inerente de se encontrar o ótimo global em espaços de elevada dimensão. Assim novos métodos de otimização, que são em sua maioria metaheurísticas, estão sendo propostos para superar os problemas de dimensionalidade.

Dada a vasta gama de metaheurísticas disponíveis atualmente na literatura, é comum organizar competições, não só para encontrar os algoritmos mais promissores, mas também incentivar a investigação original e ganhar novos conhecimentos sobre como lidar com esses problemas difíceis. Por exemplo, a pesquisa publicada em [1] propõe uma co-evolução cooperativa multinível (MLCC) para o tratamento de problemas em larga escala. No MLCC, a otimização é decomposta e a solução é construída usando grupos aleatórios de diferentes tamanhos. A eficiência desta abordagem foi avaliada em seções especiais do *IEEE Congress on Evolutionary Computation - CEC'2008* e *CEC'2010* [2].

Outros algoritmos baseados na cooperação co-evolutiva também foram propostos. Li e Yao [3] propuseram um algoritmo baseado em cooperação a partir da metaheurística de otimização por Enxame de Partículas (PSO). Os autores denominaram o novo algoritmo CCPSO, e sua melhoria usando distribuição Gaussiana CCPSO2. Concluiu-se que CCPSO2 é um algoritmo competitivo para resolver problemas com características multimodais em larga escala.

Um algoritmo baseado em cooperação e co-evolução, chamado *Smoothing and Auxiliary functions based Cooperative Co-evolution* (SACC), foi proposto em [4]. Neste algoritmo, co-evolução e cooperação são usadas para paralelizar a busca em várias regiões do espaço ao mesmo tempo. Os experimentos realizados para as funções de referência da competição *CEC'2013* mostraram que SACC fornece resultados eficientes e competitivos.

Em [5] é proposta nova abordagem de busca chamado *Offspring Multiple Sampling* (MOS). Esta abordagem se baseia na construção de uma hiper heurística usando Algoritmos Genéticos (GA) e de Evolução Diferencial (DE). Os autores demonstraram que com uso de um bom planejamento experimental, é possível combinar vários algoritmos e superar qualquer de seus antecessores, ao selecionar automaticamente o algoritmo mais adequado para cada função e fase da busca. Esta abordagem foi comparada ao algoritmo de referência da competição, DECC-G. Outros trabalhos importantes que envolvem lidar com os problemas de larga escala usando cooperação, co-evolução, heurísticas, hiper-heurísticas, podem ser encontrados em [6, 7, 8, 9].

Este trabalho relata a aplicação de uma metaheurística, chamada DEEPSO, na solução dos problemas de otimização global em larga escala propostos na competição *Large Scale Global Optimization* (LSGO) do *CEC'2015*. Este artigo está organizado como se segue: a Seção 2 apresenta e descreve a nova metaheurística DEEPSO. A Seção 3 discute as funções de *benchmark* testadas. A Seção 4 apresenta os resultados dos experimentos realizados e, finalmente, a Seção 5 inclui uma discussão sobre os resultados obtidos e apresenta uma breve conclusão do trabalho de pesquisa relatado neste artigo.

2. O algoritmo DEEPSO

O algoritmo de otimização Evolucionário Diferencial por Enxame de Partículas, do termo em inglês *Differential Evolutionary Particle Swarm Optimization* (DEEPSO) [10], é uma metaheurística que incorpora características distintas dos algoritmos evolucionários EPSO [11] e DE. A metaheurística DEEPSO é uma extensão do algoritmo EPSO. Este algoritmo que pode ser visto como um método de otimização por enxame de partículas com propriedade de seleção e autoadaptação,

ou um Algoritmo Evolucionário (AE) que implementa as regras de recombinação emprestadas do PSO.

Como qualquer metaheurística populacional, tanto EPSO quanto DEEPSO empregam o uso dos operadores de mutação, recombinação e seleção para criar novas soluções (indivíduos). Isto é feito de modo que a população é melhorada gradualmente, por meio da aptidão dos melhores indivíduos a cada geração, até que um determinado critério de parada seja atingido.

A geração de novos indivíduos no DEEPSO se dá por meio da recombinação de soluções correntes e pela seleção das melhores soluções obtidas na população anterior. A recombinação de soluções é obtida por meio da regra de movimento apresentada em (1) e (2):

$$V_t = w_i^* V_{t-1} + w_m^* (X_b - X_{t-1}) + w_c^* C (X_{gb}^* - X_{t-1}), \quad (1)$$

$$X_t = X_{t-1} + V_t, \quad (2)$$

nas quais,

- t é a geração corrente;
- X é a solução corrente;
- X_b é a melhor solução encontrada pelo enxame;
- X_{gb} é a melhor solução encontrada na geração;
- V é a velocidade do indivíduo;
- C é uma matriz diagonal de variáveis aleatórias amostrada a cada iteração com probabilidade P de comunicação, e
- “*” indica que o parâmetro é submetido ao processo de mutação.

Tipicamente, a mutação do peso w de um determinado indivíduo genérico segue a regra:

$$w^* = w + \tau \times N(0, 1), \quad (3)$$

sendo que τ representa a taxa de mutação com probabilidade P e deve ser definida pelo usuário. $N(0, 1)$ é um número amostrado a partir da distribuição gaussiana. Note que o peso mutado não deve ser negativo ou maior que 1. Na Equação (1) a melhor solução da geração, X_{gb} , também é mutada.

Esta posição é ligeiramente atraída pelo movimento no espaço de busca usando uma distribuição Gaussiana. Isto é feito para evitar que uma população fique presa em uma determinada região, onde é constatado que determinado indivíduo possui valor dominante frente aos demais [10]. A mutação de X_{gb} é obtida por meio da Equação (4),

$$X_{gb}^* = X_{gb} [1 + \tau \times N(0, 1)]. \quad (4)$$

A característica que distingue DEEPSO consiste na utilização de uma melhor percepção da paisagem de otimização. Similarmente a DE e PSO, esta percepção pode ser grosseiramente obtida comparando soluções diferentes. Para aproveitar as informações coletadas no espaço de busca, DEEPSO faz uso de um conjunto de memória. Para este propósito, o termo de memória na Equação 1 é substituído por um novo termo, que recebeu a nomenclatura de “termo de percepção”, conforme:

$$V_t = w_i^* V_{t-1} + w_p^* (X_r - X_{t-1}) + w_c^* C (X_{gb}^* - X_{t-1}), \quad (5)$$

na qual X_r é um indivíduo diferente de X_{t-1} e pode ser obtido de acordo com as opções:

1. extraído de todos os indivíduos da geração atual $\rightarrow S_g$;
2. extraído a partir de uma memória “B” dos melhores indivíduos das gerações anteriores $\rightarrow P_B$;
3. extraído como uma recombinação uniforme dos indivíduos da geração atual $\rightarrow S_g\text{-}rnd$;
4. extraído como uma recombinação uniforme da memória “B” $\rightarrow P_B\text{-}rnd$, e;
5. uma **nova proposta** deste trabalho que une a ideia contida em 3 e 4 $\rightarrow S_g P_B\text{-}rnd$.

A estrutura de memória “B” guarda a posição e a aptidão do indivíduo. A informação da aptidão é usada para verificar se X_{t-1} vai ser atraído por X_r ou o contrário. Se a aptidão de X_r é melhor que a aptidão de X_{t-1} , a Equação (5) é usada sem modificações. Caso contrário, a Equação (6) é utilizada:

$$V_t = w_i^* V_{t-1} + w_p^* (X_{t-1} - X_r) + w_c^* C(X_{gb}^* - X_{t-1}). \quad (6)$$

Na Equação (6) os termos X_{t-1} e X_r estão em posição invertida em relação à Equação (5) para excluir um indivíduo de pior aptidão. Neste caso, nas opções 3 e 4, onde uma recombinação uniforme de diferentes soluções é usada para obter X_r , a inversão da posição de X_{t-1} e X_r em (6) é feita em cada dimensão do espaço de busca. Isto faz com que uma seleção aleatória da posição de memória forneça a dimensão i de X_r . A aptidão da posição selecionada é comparada com a aptidão de X_{t-1} para decidir se o indivíduo vai ser atraído ou não nesta dimensão particular no espaço de busca. Este procedimento é repetido em todas as dimensões de X_r .

3. Funções de Benchmark

A modalidade de uma função está relacionada com o número de pontos (máximos e mínimos) presentes em sua paisagem, o que induz uma tendência para que o algoritmo fique preso em um deles durante o processo de busca. Isto ocasiona um impacto negativo sobre a busca, fazendo com que ela caminhe para regiões afastadas das verdadeiras soluções ótimas. Uma função que possui mais de um valor ótimo é chamada de multimodal.

A separabilidade é também uma propriedade que pode implicar em dificuldade para um problema de otimização. Uma função é dita separável se as variáveis são independentes umas das outras e, por conseguinte, pode ser escrita como uma soma de outras funções subobjetivo com apenas uma variável de decisão. Quando estas variáveis de decisão são inter-relacionadas, a função é chamada não-separável [12].

As funções de *benchmark* propostas pela competição LSGO CEC’2015 são apresentadas em [13]. Este trabalho define quatro categorias principais para funções de larga escala: deslocadas, não-separáveis, com subcomponentes sobrepostos e rotacionadas. As funções de base, que são utilizadas para formar as 15 funções propostas, são listadas a seguir:

Função Esférica:

$$f_{sphere}(x) = \sum_{i=1}^D x_i^2. \quad (7)$$

Função Elíptica:

$$f_{elliptic}(x) = \sum_{i=1}^D 10^{6 \frac{i-1}{D-1}} x_i^2. \quad (8)$$

Função Rastrigin:

$$f_{rastrigin}(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]. \quad (9)$$

Função Ackley:

$$f_{ackley}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e. \quad (10)$$

Função Schwefel:

$$f_{schwefel}(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2. \quad (11)$$

Função Rosenbrock,

$$f_{rosenbrock}(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]. \quad (12)$$

A partir de tais funções, 15 problemas¹ *benchmark* foram propostos e estão apresentados na Tabela 1.

Tabela 1: Problemas de *Benchmark* prpostos no LSGO'CEC 2015

Nº	Tipo	Função
<i>f</i> 1	Unimodal	Elíptica Deslocada
<i>f</i> 2	Multimodal	Rastrigin Deslocada
<i>f</i> 3	Multimodal	Ackley Deslocada
<i>f</i> 4	Unimodal	Elíptica 7-Não-separável, 1-Separável Deslocada/Rotacionada
<i>f</i> 5	Multimodal	Rastrigin 7-Não-separável, 1-Separável Deslocada/Rotacionada
<i>f</i> 6	Multimodal	Ackley 7-Não-separável, 1-Separável Deslocada/Rotacionada
<i>f</i> 7	Multimodal	Schwefel 7-Não-separável, 1-Separável Deslocada
<i>f</i> 8	Unimodal	Elíptica 20-Não-separável, 1-Separável Deslocada/Rotacionada
<i>f</i> 9	Multimodal	Rastrigin 20-Não-separável, 1-Separável Deslocada/Rotacionada
<i>f</i> 10	Multimodal	Ackley 20-Não-separável, 1-Separável Deslocada/Rotacionada
<i>f</i> 11	Unimodal	Schwefel 20-Não-separável, 1-Separável Deslocada/Rotacionada
<i>f</i> 12	Multimodal	Rosenbrock Deslocada
<i>f</i> 13	Unimodal	Schwefel Deslocada (sub-componentes sobrepostos em conformidade)
<i>f</i> 14	Unimodal	Schwefel Deslocada (sub-componentes sobrepostos em conflito)
<i>f</i> 15	Unimodal	Schwefel Deslocada

¹O valor ótimo global das 15 funções de *benchmark* é zero

4. Experimentos realizados

Nesta seção os resultados do algoritmo DEEPSO, submetido às 15 funções de teste, são apresentados e discutidos. DEEPSO foi executado 25 vezes para cada problema *benchmark* em um computador Intel Xeon E5530 - 2.40GHz com 12GB de memória RAM, com sistema operacional Windows 7 (64 bits) e em ambiente Matlab R2011a. A Tabela 2 apresenta os parâmetros de inicialização do algoritmo DEEPSO para solução destes problemas. O critério de parada adotado em DEEPSO foi o número máximo de avaliações de função (FEs), conforme o utilizado por DECC-G.

Tabela 2: Parâmetros de inicialização

DEEPSO $S_g P_b$ - <i>rnd</i>	$f1 - f11$	$f12 - f15$
Avaliações	3.0×10^6	3.0×10^6
Dimensão	1000	905
População	500	500
Memória	50	50
Mutação	0.8	0.8
Comunicação	0.9	0.9

4.1. Resultados dos Experimentos

Os resultados dos testes indicam alguns valores estatísticos imediatos, tais como média, mediana e desvio padrão das respostas do algoritmo DEEPSO, nas 25 execuções realizadas para cada problema, além de mostrar os melhores e piores resultados quando o máximo para avaliações de função (FE) atingir os critérios de 1.2×10^5 , 6.0×10^5 e 3.0×10^6 . Os resultados deste experimento são apresentados na Tabela 3, na qual a primeira coluna lista o número de FE.

As Figuras 1, 2 e 3 apresentam as curvas de convergência média do DEEPSO para os seis problemas selecionados pela competição, os quais são: f_2 , f_7 , f_{11} , f_{12} , f_{13} e f_{14} , nos quais o eixo “ x ” informa o número máximo de avaliações de função (FEs) e o eixo “ y ” indica o valor de função (*fitness*). É possível notar que, em todas as figuras, o decaimento da curva de convergência média é suave. Todas as curvas indicam um desempenho progressivo e estável em direção à solução final.

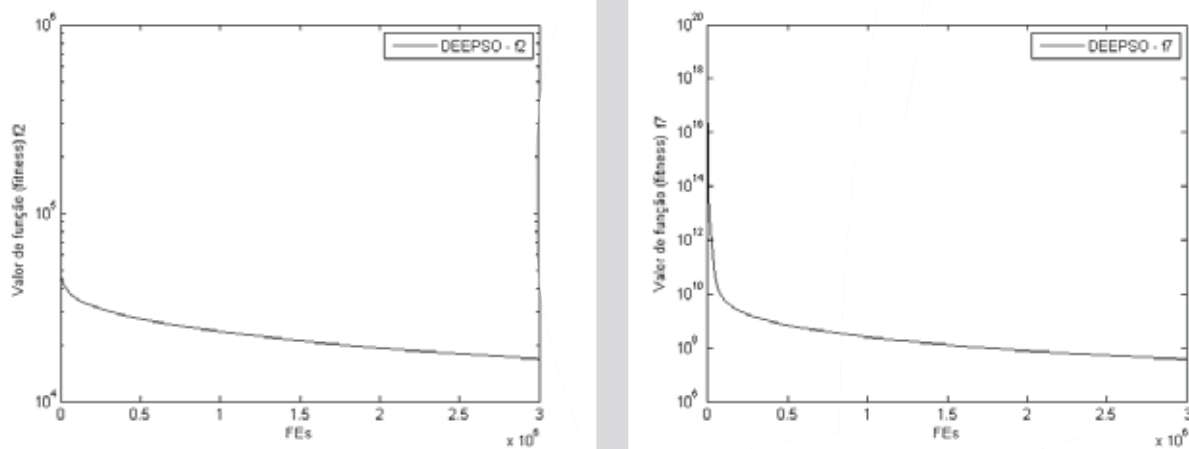
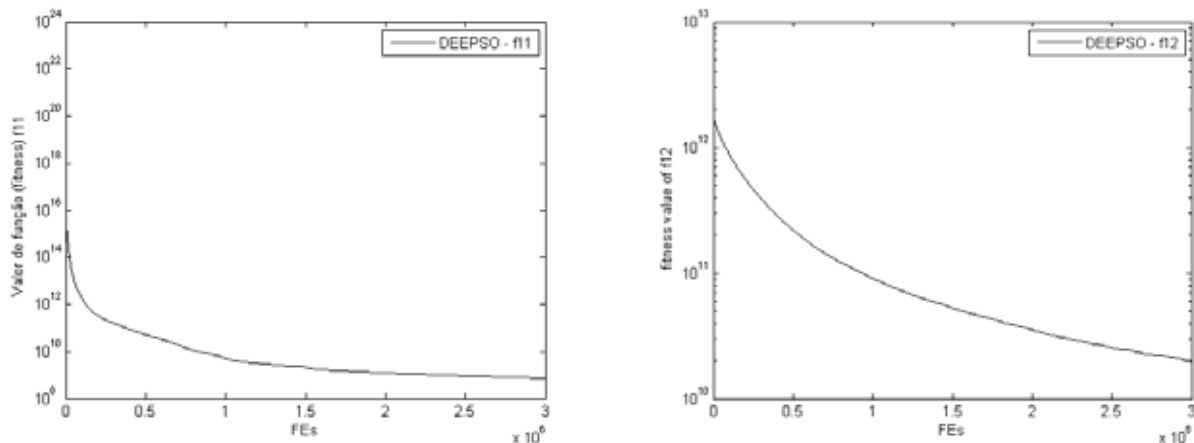


Figura 1: Curvas de convergência média do algoritmo DEEPSO nas funções f_2 e f_7 .

Funções		$f1$	$f2$	$f3$	$f4$	$f5$
1.2×10^5	Melhor	2,99E+10	3,45E+04	2,12E+01	4,27E+11	1,56E+07
	Mediana	5,58E+10	3,75E+04	2,14E+01	9,06E+11	1,76E+07
	Pior	4,83E+11	1,48E+05	2,17E+01	2,47E+14	9,74E+07
	Média	7,34E+10	3,89E+04	2,14E+01	1,17E+12	1,93E+07
	DPadrão	4,85E+10	6,13E+03	9,50E-02	2,18E+12	5,09E+06
6.0×10^5	Melhor	4,61E+09	2,61E+04	2,09E+01	4,95E+10	1,46E+07
	Mediana	1,18E+10	2,99E+04	2,11E+01	1,49E+11	1,46E+07
	Pior	4,83E+11	1,48E+05	2,17E+01	2,47E+14	9,70E+07
	Média	2,37E+10	3,12E+04	2,11E+01	4,74E+11	1,56E+07
	DPadrão	3,34E+10	5,15E+03	1,56E-01	1,40E+12	2,95E+07
3.0×10^6	Melhor	1,44E+08	1,49E+04	2,04E+01	4,77E+09	1,45E+07
	Mediana	2,53E+08	1,68E+04	2,08E+01	7,52E+09	1,84E+07
	Pior	7,43E+08	1,88E+04	2,06E+01	1,07E+10	2,20E+07
	Média	3,18E+08	1,68E+04	2,06E+01	7,42E+09	1,86E+07
	DPadrão	1,70E+08	8,63E+02	9,34E-02	1,88E+09	2,16E+06
Funções		$f6$	$f7$	$f8$	$f9$	$f10$
1.2×10^5	Melhor	1,05E+06	3,50E+09	8,33E+14	1,03E+09	9,41E+07
	Mediana	1,05E+06	2,30E+10	3,09E+15	1,20E+09	9,43E+07
	Pior	1,08E+06	3,20E+18	1,73E+19	8,95E+09	9,92E+07
	Média	1,06E+06	1,50E+14	3,35E+16	1,29E+09	9,45E+07
	DPadrão	2,90E+03	1,36E+16	1,45E+17	4,39E+09	5,04E+05
6.0×10^5	Melhor	1,04E+06	2,40E+08	1,68E+14	9,29E+09	9,22E+07
	Mediana	1,04E+06	6,80E+08	3,00E+14	9,60E+09	9,35E+07
	Pior	1,08E+06	3,25E+18	1,72E+14	8,95E+09	9,92E+07
	Média	1,05E+06	3,01E+13	6,97E+15	1,02E+09	9,35E+07
	DPadrão	6,13E+06	6,11E+15	6,66E+16	2,39E+09	7,34E+07
3.0×10^6	Melhor	1,02E+06	1,54E+07	5,42E+12	9,17E+08	9,07E+07
	Mediana	1,04E+06	3,20E+07	1,48E+13	1,29E+09	9,18E+07
	Pior	1,05E+06	7,23E+07	6,74E+13	1,88E+09	9,38E+07
	Média	1,04E+06	3,70E+07	1,79E+13	1,34E+09	9,20E+07
	DPadrão	7,54E+03	1,48E+07	1,29E+13	2,38E+08	7,91E+07
Funções		$f11$	$f12$	$f13$	$f14$	$f15$
1.2×10^5	Melhor	7,52E+11	7,80E+11	1,82E+11	2,18E+12	6,60E+12
	Mediana	7,76E+12	1,10E+11	1,96E+12	1,19E+13	5,22E+13
	Pior	3,15E+19	9,61E+11	3,13E+20	2,35E+19	4,36E+18
	Média	7,84E+15	1,18E+11	9,28E+15	7,02E+15	3,21E+15
	DPadrão	2,92E+17	5,42E+11	1,28E+18	1,77E+17	5,93E+16
6.0×10^5	Melhor	8,03E+09	1,55E+11	1,29E+10	1,63E+11	3,00E+07
	Mediana	7,82E+10	3,80E+11	2,54E+10	3,94E+11	9,60E+07
	pior	3,25E+19	9,61E+12	3,13E+20	2,35E+19	4,36E+18
	Média	1,56E+15	5,22E+11	1,85E+20	1,44E+15	6,44E+14
	DPadrão	1,31E+17	4,36E+11	5,73E+17	7,92E+16	2,65E+16
3.0×10^6	Melhor	5,60E+08	1,54E+10	8,75E+08	4,33E+08	7,04E+06
	Mediana	7,59E+08	3,43E+10	1,47E+09	1,65E+09	9,57E+06
	Pior	9,04E+08	1,78E+10	3,34E+09	3,17E+10	1,75E+07
	Média	7,43E+08	1,98E+10	1,54E+09	5,19E+09	1,02E+07
	DPadrão	1,13E+08	5,83E+09	4,78E+08	8,19E+09	2,43E+06

 Tabela 3: Resultados do algoritmo DEEPSO nas funções $f1 - f15$


 Figura 2: Curvas de convergência média do algoritmo DEEPSO nas funções f_{11} e f_{12} .

A partir da Tabela 4, se pode verificar que, em 8 problemas de *benchmark*, o algoritmo DEEPSO apresenta melhores resultados em relação ao algoritmo DECC-G, de acordo com as medidas estatísticas da competição. Isso indica que o algoritmo DEEPSO se mostra competitivo para solução de problemas em larga escala.

Medidas	f	DEEPSO	DECC-G	f	DEEPSO	DECC-G	f	DEEPSO	DECC-G
Melhor	f1	1,44E+08	1,75E-13	f2	1,49E+04	9,90E+02	f3	1,45E+07	2,63E-10
Mediana		2,53E+08	2,00E-13		1,68E+04	1,03E+03		7,52E+09	2,85E-10
Pior		7,43E+08	2,45E-13		1,88E+04	1,07E+03		1,07E+10	3,16E-10
Média		3,18E+08	2,03E-13		1,68E+04	1,07E+03		7,42E+09	2,87E-10
DPadrão		1,70E+08	1,78E-14		8,63E+02	2,26E+01		1,88E+09	1,38E-11
Best	f4	4,77E+09	7,58E+09	f5	1,45E+07	7,28E+14	f6	1,02E+06	6,96E-08
Mediana		7,52E+09	2,12E+10		1,84E+07	7,28E+14		1,04E+06	6,08E+04
Pior		1,07E+10	6,99E+10		2,20E+07	7,28E+14		1,05E+06	1,10E+05
Média		7,42E+09	2,60E+10		1,86E+07	7,28E+14		1,04E+06	4,85E+04
DPadrão		1,88E+09	1,47E+10		2,16E+06	1,51E+05		7,54E+03	3,98E+04
Melhor	f7	1,54E+07	1,96E+08	f8	5,42E+12	1,43E+14	f9	9,17E+08	2,20E+08
Mediana		3,20E+07	4,27E+08		1,48E+13	3,88E+14		1,29E+09	4,17E+08
Pior		7,23E+07	1,78E+09		6,74E+13	7,75E+14		1,88E+09	6,55E+08
Média		3,70E+07	6,07E+08		1,79E+13	4,26E+14		1,34E+09	4,27E+08
DPadrão		1,48E+07	4,09E+08		1,29E+13	1,53E+14		2,38E+08	9,89E+08
Melhor	f10	9,07E+07	9,29E+04	f11	5,60E+08	4,68E+10	f12	1,54E+10	9,80E+02
Mediana		9,18E+07	1,19E+07		7,59E+08	1,60E+11		3,43E+10	1,03E+03
Pior		9,38E+07	1,73E+07		9,04E+08	7,16E+11		1,78E+10	1,30E+03
Média		9,20E+07	1,10E+07		7,43E+08	2,46E+11		1,98E+10	1,04E+03
DPadrão		7,91E+07	4,00E+07		1,13E+08	2,03E+11		5,83E+09	5,76E+01
Melhor	f13	8,75E+08	2,09E+10	f14	4,33E+08	1,91E+11	f15	7,04E+06	4,63E+07
Mediana		1,47E+09	3,35E+10		1,65E+09	6,27E+11		9,57E+06	5,61E+07
Pior		3,34E+09	4,64E+10		3,17E+10	1,04E+12		1,75E+07	7,15E+07
Média		1,54E+09	3,42E+10		5,19E+09	6,08E+11		1,02E+07	6,05E+07
DPadrão		4,78E+08	6,41E+10		8,19E+09	2,06E+11		2,43E+06	6,45E+07

 Tabela 4: Comparação entre os algoritmos DEEPSO e DECC-G ($f_1 - f_{15}$)

Além disso, DEEPSO parece indicar melhor desempenho em 6 dos 7 problemas de característica unimodal, mostrando ser eficiente para busca em funções deste tipo. Porém, não há como neste momento, realizar um planejamento de experimentos estatístico para corroborar a hipótese de que DEEPSO possui melhor desempenho, uma vez que os resultados do algoritmo da literatura não estão disponíveis para tal abordagem.

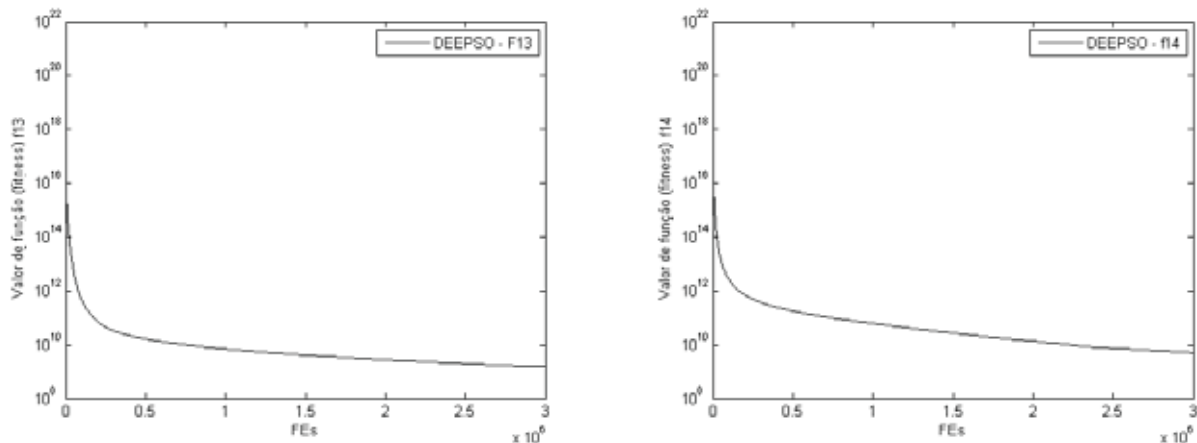


Figura 3: Curvas de convergência média do algoritmo DEEPSO nas funções f_{13} e f_{14} .

5. Conclusão

Este trabalho investigou o algoritmo DEEPSO com uso de uma nova estratégia de mutação denominada $S_g P_B\text{-}rnd$, para solução de problemas em larga escala. DEEPSO é uma metaheurística mono-objetivo que incorpora algumas características de Programação Evolutiva, Otimização por enxame de Partículas e Algoritmos de Evolução Diferencial. Com o intuito de testar o desempenho do algoritmo DEEPSO, o mesmo foi submetido a 15 problemas *benchmark* em larga escala.

Os resultados indicam que DEEPSO é um algoritmo eficiente e competitivo para solucionar problemas em larga escala, especialmente os que possuem característica unimodal. Os resultados experimentais mostram também que esta metaheurística apresenta melhores soluções quando comparada com o algoritmo de referência, o DECC-G da competição LSGO CEC'2013.

Como trabalho futuro propõe-se o refinamento dos parâmetros de inicialização do algoritmo DEEPSO visando obter melhores resultados do que os aqui apresentados. Assim, espera-se que este algoritmo possa apresentar resultados ainda melhores para tratamento de problemas em grande dimensionalidade.

6. Agradecimentos

Os autores agradecem ao CEFET-MG e INESC TEC pela infraestrutura utilizada para realização desta pesquisa, a CAPES, CNPq e FAPEMIG pelo apoio financeiro. Também agradecem o apoio de todos os investigadores do INESC TEC que contribuíram no decorrer deste trabalho.

Referências

- [1] Z. Yang, K. Tang, and X. Yao. Multilevel cooperative coevolution for large scale optimization. In *Proc. IEEE Congress on Evolutionary Computation*. pp 1663-1670, 2008.
- [2] M. N. Omidvar, X. Li, Z. Yang, and Yao. Cooperative coevolution for large scale optimization through more frequent random grouping. In *Proc. IEEE Congress on evolutionary Computation*. pp. 1754-1761, 2010.

- [3] X. Li and X. Yao. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Transactions on Evolutionary Computation*, 16:210–224, 2012.
- [4] F. Wei, Y. Wang, and Y. Huo. Smoothing and auxiliary functions based cooperative coevolution for global optimization. In *Proc: IEEE Congress on Evolutionary Computation*. pp 2736-2741, 2013.
- [5] A. LaTorre and J-M. Muelas, S. Pena. Large scale global optimization: Experimental results with mos-based hybrid algorithms. In *Proc: IEEE Congress on Evolutionary Computation*. pp:2742-2749, 2013.
- [6] A. LaTorre, Muelas S., and Pena J-M. Multiple offspring sampling in large scale global optimization problems. In *Proc: IEEE Congress on evolutionary Computation*. pp: 964-971, 2012.
- [7] D. Molina, M. Lozano, and F. Herrera. Ma-sw-chains: Memetic algorithm based on local search chains for large scale continuous global optimization. In *Proc: IEEE Congress on Evolutionary Computation*. pp: 1-8, 2010.
- [8] K Yang, K. Tang, and X. Yao. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178:2986–2999, 2008.
- [9] L. Tseng and C. Chen. Multiple trajectory search for large scale global optimization. In *Proc: IEEE Congress on Evolutionary Computation*. pp: 3052-3059, 2008.
- [10] V. Miranda and R. Alves. Differential evolutionary particle swarm optimization (deepso): a successful hybrid. In *Proc: 11th Brazilian Congress on Computational Intelligence (BRICS-CCI)*. pp 368-374, 2013.
- [11] V. Miranda and N. Fonseca. Epsos best-of-two-worlds meta-heuristic applied to power system problems. In *Proc: of the 2002 Congress on Evolutionary Computation (CEC)*, vol. 2, pp. 10801085, 2002.
- [12] M. Locatelli. A note on the griewank test function. *Journal of global optimization*, 25(2):169-174, 2003.
- [13] X. Li, K. Tang, M. Omidivar, Z. Yang, and K. Qin. Benchmark functions for the cec'2013 special session and competition on large-scale global optimization. Technical report, IEEE CEC, 2013.