

Combinação de Otimizadores por Exame de Partículas com Memória

Barreto, R. V. S., Lizandro, N. S.

Escola Politécnica de Pernambuco – Universidade de Pernambuco

rvsb@ecomppoli.br, lizandro.nunes@poli.br

Resumo

Um problema de otimização é aquele no qual é possível encontrar vários resultados, mas é desejado encontrar a solução onde o valor da função é máximo ou mínimo. A fim de resolver estes problemas, foram desenvolvidos diversos algoritmos com inteligência artificial baseado na Natureza. Porém, dado a variedade de problemas de otimização é difícil concluir qual o melhor algoritmo a ser utilizado. Com o propósito de resolver o máximo destes problemas é analisada a combinação desses otimizadores. Este trabalho introduz um novo algoritmo de Combinação de Otimizadores chamado EPSOM (*Ensembled Particle Swarm Optimizer with Memory*) visando resolver uma grande diversidade destes problemas.

1. Introdução

Em problemas de otimização, o interesse é encontrar a solução em que a função objetivo é mínima ou máxima. Este tipo de problema aparece em diversas áreas da ciência, contendo variados tipos de natureza, nos quais podem ser citados problemas unimodais ou multimodais.

Uma função objetivo unimodal é aquela onde apresenta apenas um ponto de máximo ou de mínimo, ou seja, possui apenas uma solução ótima [1]. Já uma função multimodal pode conter vários pontos onde a solução é ótima. Essas soluções em funções multimodais são descritas como ótimos locais. A quantidade soluções e dimensões da função objetivo podem contribuir no grau de dificuldade no encontro da solução de um problema de otimização e na quantidade de recursos computacionais utilizados.

Em razão de resolver estas dificuldades foram desenvolvidos nas últimas décadas diversos algoritmos com base no que se observa na natureza, como

Otimização por Enxame de Partículas (*PSO – Particle Swarm Optimizer*) [2], Colônia Artificial de Abelha (*ABC – Artificial Bee Colony*) [3], Otimização por Colônia de Formigas (*ACO – Ant Colony Optimization*) [4], Algoritmos Genéticos (*GA – Genetic Algorithm*) [5], etc.

Porém, é difícil concluir qual o melhor algoritmo a ser aplicado na resolução de um problema de otimização, pois cada um é especializado numa série de funções específicas, as quais podem não se encaixar em novos cenários de otimização [6]. Esta dificuldade foi exibida no teorema da inexistência do almoço grátis [7], em que se estabelece que nenhum algoritmo de otimização pode obter as melhores soluções com um baixo custo computacional e com rapidez em todos os tipos de problemas de otimização.

Utilizando como base o algoritmo EPSO, este artigo tem como objetivo introduzir uma modificação nesse algoritmo com o intuito de deixá-lo mais rápido e mais preciso.

1.1. Descrição do Problema

Com objetivo de resolver estes problemas de otimização, foi proposto um algoritmo chamado Otimizador por Combinação de Otimizadores por Enxame de Partículas (*EPSO – Ensemble Particle Swarm Optimizer*) [8]. Ele é composto por 5 (cinco) variantes do PSO, cada um especialista em resolver um tipo de problema específico, enquanto que o EPSO escolhe de maneira auto-adaptativa o melhor algoritmo pela técnica de seleção por torneio.

Durante um período de aprendizagem (*Learning Period – LP*), a variante do PSO escolhida da geração é selecionada de acordo com uma probabilidade igualitária e neste período é armazenado quantas partículas conseguiram melhorar seu *fitness* (N_s) e quantas pioraram (N_f). Após essa fase, o algoritmo

escolhido é baseado em uma probabilidade que varia de acordo com o N_s e N_f das gerações anteriores.

Contudo, a aprendizagem do EPSO é baseada em apenas na quantidade de sucesso e falhas das gerações anteriores, não fazendo uma distinção entre gerações antigas e as gerações mais novas.

2. Método Proposto

Neste trabalho é proposto o algoritmo Combinação de Otimizadores por Exame de Partículas com Memória (EPSOM - Ensemble Particle Swarm Optimizer with Memory), que atribui um peso para cada geração como uma forma de memória destas gerações e destaca as gerações mais evoluídas.

Dado que o EPSOM é uma derivação do EPSO, também são utilizados os mesmos cinco métodos: WPSO (Weighted Particle Swarm Optimizer) [9], o CLPSO (Comprehensive Learning Particle Swarm Optimizer for Global Optimizer) [10], o FDRPSO (Fitness-Distance-Ratio Based Particle Swarm Optimizer) [11], o LIPS (Distance-based locally informed PSO) [12] e o HPSOTVAC (Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients) [13].

2.1 Método PSO

O método PSO simula o comportamento de um enxame de partículas onde cada partícula representa uma potencial solução para o problema e possui d dimensões espaciais. Cada i -ésima partícula possui uma posição e velocidade com d dimensões, representadas pelas variáveis V_i^d e X_i^d . A posição representa os pontos utilizados no cálculo da função objetivo do problema (*fitness*) e a velocidade dita a variação desses pontos.

Na execução do algoritmo, cada partícula é iniciada aleatoriamente e comunica-se com o resto do enxame para obter as melhores posições e ajustar sua velocidade e posição. Assumindo que r_1 e r_2 são variáveis geradas aleatoriamente e uniformemente entre 0 e 1, p_i é a melhor posição já encontrada para a i -ésima partícula, g é a melhor posição já encontrada pelo enxame e c_1 e c_2 são coeficientes de individualidade e sociabilidade, respectivamente; as equações do cálculo da velocidade e posição são dadas por:

$$V_i^d = V_i^d + r_1^d c_1 (p_i^d - X_i^d) + r_2^d c_2 (g^d - X_i^d) \quad (1)$$

$$X_i^d = X_i^d + V_i^d \quad (2)$$

A velocidade da partícula em cada dimensão é limitada a uma magnitude máxima V_{max} , definida pelo usuário. Se V_i^d exceder a magnitude máxima, seu valor será $V_i^d = sign(V_i^d) \times V_{max}$, onde *sign* é a função sinal [14]. O mesmo pode ocorrer para a posição, que está limitada entre X_{min} e X_{max} dependendo da função.

Considerando que o processo será repetido por G gerações, que a quantidade de indivíduos na população é P , que o $Fit(X_i)$ é o resultado do cálculo da função objetivo utilizando a posição da i -ésima partícula e que k , i e d são contadores de geração, partícula e dimensão, respectivamente; segue abaixo um fluxograma deste algoritmo:



Fig. 1. Fluxograma do algoritmo PSO

2.1.1 WPSO

Este algoritmo propõe a inclusão de um novo coeficiente w que possa aumentar o espaço de busca nas primeiras gerações, procurando um ótimo global, e nas últimas gerações procurar pela melhor posição nos ótimos locais [9]. Com esse ajuste, o cálculo da velocidade do PSO torna-se

$$V_i^d = wV_i^d + r_1^d c_1 (p_i^d - X_i^d) + r_2^d c_2 (g^d - X_i^d) \quad (3)$$

e o w entre $w_k = 0.9 - 0.2 * \left(\frac{k}{G}\right)$. Onde o k é o valor da geração atual. Com essas características, este algoritmo é adequado quando o espaço de busca é grande e variado.

2.1.2 CLPSO

O algoritmo CLPSO utiliza uma variação do WPSO para o cálculo da sua velocidade, porém sem o fator da melhor partícula global e fazendo um ajuste no cálculo da melhor posição da partícula. Esse ajuste faz com que a busca não seja feita que a melhor posição pela partícula, e sim na melhor posição de outras partículas da população aleatoriamente escolhidas de acordo com uma variável $fi(d)$.

Com isso, o cálculo da velocidade torna-se

$$V_i^d = wV_i^d + r_1^d c_1 (p_{fi(d)}^d - X_i^d) \quad (4)$$

e para cada partícula, um valor aleatório Pc é gerado na primeira geração:

$$Pc_i = a + b \times \frac{\exp\left(\frac{10 * i}{P}\right) - 1}{\exp(10) - 1} \quad (5)$$

Em que o valor padrão é $a = 0,05$ e $b = 0,45$, e i é a i -ésima partícula. Para cada dimensão d , se um valor aleatório gerado $r_{2,i}^d$ for menor que o Pc_i , então é escolhido duas partículas de forma aleatória e a que possuir o melhor p_i será selecionada como $fi(d)$; se não, o cálculo da velocidade será feito com seu próprio

p_i . No caso de todos os $r_{2,i}^d$ serem maior que o Pc , será escolhida uma dimensão e partícula aleatória para calcular a velocidade de acordo com a equação (4).

Este algoritmo é adaptado para resolver funções multimodais, porém não é a melhor escolha quando se trata de funções unimodais.

2.1.3 HPSO-TVAC

O HPSO-TVAC é resultado da junção de dois algoritmos; o *Time-Varying Acceleration Coefficients* (TVAC) [13], onde os coeficientes c_1 e c_2 variam de acordo com as gerações, e o *Particle Swarm Optimizer with "Mutation" and Time-Varying Acceleration Coefficients* (MPSO-TVAC) [13], no qual a velocidade pode ser alterada conforme uma mutação probabilística.

Neste algoritmo, os coeficientes c_1 e c_2 da equação (1) se comportam como na seguinte equação

$$c_1 = (c_{1f} - c_{1i}) \frac{i}{\text{iterações}} + c_{1i} \quad (6)$$

$$c_2 = (c_{2f} - c_{2i}) \frac{i}{\text{iterações}} + c_{2i} \quad (7)$$

E o cálculo da velocidade neste algoritmo segue a próxima equação (8), removendo o termo V_i^d da equação (1).

$$V_i^d = r_1^d c_1 (p_i^d - X_i^d) + r_2^d c_2 (g^d - X_i^d) \quad (8)$$

Se a velocidade V_i^d calculada for igual a 0, ela assumirá um novo valor de acordo com uma velocidade v de reinício e um valor aleatório r_3 entre 0 e 1. Então esse cálculo é feito da seguinte forma:

$$V_i^d = r_3 \times v \quad (9)$$

Este algoritmo tem êxito ao trabalhar com funções unimodais porém não realiza bem as funções multimodais por convergir em mínimos locais prematuramente.

2.1.5 FDRPSO

O FDRPSO utiliza as partículas próximas para o cálculo da velocidade. Porém a escolha dessas partículas leva em consideração a distância e o melhor fitness visitado e não apenas a distância euclidiana. Esse cálculo de distância é feito conforme a equação (12).

$$D_{i,j} = \frac{Fitness(p_j) - Fitness(X_i)}{|p_j^d - X_i^d|} \quad (12)$$

Após a escolha da partícula mais próxima $n_{i,j}$, o cálculo da velocidade é realizado com três variáveis: a melhor posição da partícula (p_i), a melhor posição do enxame (g) e a melhor posição da melhor partícula próxima ($n_{i,j}$).

$$V_i^d = wV_i^d + c_1r_1^d \times (p_i^d - X_i^d) + c_2r_2^d \times (g_i^d - X_i^d) + c_3 \times (n_{i,j}^d - X_i^d) \quad (13)$$

Em que c_1 , c_2 e c_3 são coeficientes de individualidade, sociabilidade e de vizinhança, respectivamente.

O algoritmo FDRPSO é capaz de evitar convergência prematura e é menos provável que fique preso em um mínimo local.

2.1.4 Método LIPS

Ao contrário dos métodos por enxame citados anteriormente, que utilizam a melhor posição já encontrada pela partícula (p_i) e pela população (g), o algoritmo LIPS utiliza as partículas mais próximas no cálculo da velocidade (n_i). Para descobrir as partículas mais próximas, é realizado o cálculo da distância de acordo com a equação de distância euclidiana

$$D_i = \sqrt{(\sum_{d=0}^n X_i^d - X^d)^2}$$

Considerando que φ_j são valores gerados uniformemente distribuídos aleatoriamente entre 0 e $4.1/n_{size}$ e φ é a somatória de todos φ_j , e n_{size} é a

quantidade de partículas vizinhas definidos pelo usuário.

A posição média das partículas vizinhas é dada por:

$$P_i = \frac{\sum_{j=1}^{n_{size}(\varphi_j \times n_j)} / n_{size}}{\varphi} \quad (10)$$

Seguindo essas definições, o cálculo da velocidade é determinado conforme a equação (11).

$$V_i^d = w \times (V_i^d + \varphi(P_i^d - X_i^d)) \quad (11)$$

Com isto, o LIPS usa a informação dos vizinhos da mesma região e isso melhora a capacidade de busca local do algoritmo, especialmente nos últimos estágios da pesquisa dos pontos. Essa capacidade permite que as partículas possam convergir rapidamente para o melhor global (g) com alta precisão.

2.2 Método EPSO

O método EPSO (*Ensembled Particle Swarm Optimizer*) utiliza esses cinco métodos variantes do PSO; WPSO, CLPSO HPSO-TVAC, FDRPSO e LIPS, de forma auto adaptativa. É utilizado uma probabilidade p_k para cada método e é aplicado a técnica de seleção definido em [15] na escolha do método da geração. Inicialmente, o p_k é inicializado com $1/k$. Em que k é o número total de métodos PSO existentes no *ensemble*.

Na geração g , após avaliar toda a população, o número de sucesso e de falhas do k -ésimo método será armazenado em $ns_{k,g}$ e $nf_{k,g}$. Em que o número de sucesso é dado pela quantidade de indivíduos que conseguiram melhorar seu *fitness* e o número de falhas é a quantidade de indivíduos que não melhoraram seu *fitness*.

Essas memórias de sucesso ($ns_{k,g}$) e de falha ($nf_{k,g}$) serão armazenadas e atualizadas de acordo com o número fixo de gerações, definido pelo período de aprendizagem (LP – *Learning Period*). Nas gerações após o LP a probabilidade de escolha de um algoritmo é dada por:

$$p_{k,g} = \frac{S_{k,g}}{\sum_{k=1}^K S_{k,g}} \quad (14)$$

em que $S_{k,g}$, a taxa de sucesso, é calculado de acordo com

$$S_{k,g} = \frac{\sum_{g=G-LP}^{G-1} n s_{k,g}}{\sum_{g=G-LP}^{G-1} n s_{k,g} + \sum_{g=G-LP}^{G-1} n f_{k,g}} + \varepsilon; G > LP \quad (15)$$

em que $\varepsilon = 0,01$ é utilizado para evitar possíveis taxas de sucessos nulas.

O método com maior taxa de sucesso terá maior probabilidade de ser selecionado para guiar a população da geração atual, ou seja, a variante de PSO mais adequada para um problema específico é gradualmente a mais selecionada pela estratégia de seleção auto-adaptativa.

2.3 Método EPSOM

De acordo com a equação (15), todas as gerações entre $G - 1$ e $G - LP$ possuem o mesmo grau de significância entre elas. Face ao exposto, a técnica EPSOM (*Ensemble Particle Swarm Optimizer with Memory*) sugere um termo de ajuste para todas as gerações formadas pelos algoritmos, ou seja, cada geração de partículas receberá um peso, denotado de w_g , tal que

$$w_g = \begin{cases} 1, & \text{se } M < \Delta G \\ 1 - \frac{\Delta G}{M}, & \text{caso contrário} \end{cases} \quad (16)$$

em que $\Delta G = (G - 1 - g)$, G é a geração máxima, g é geração atual e M é o número de gerações para os quais $w_g > 0$. Em outras palavras, M representa a memória do sistema proposto, isto é, a quantidade de gerações que serão consideradas para a escolha do algoritmo de otimização.

Dado o parâmetro w_g , ao aplicarmos em (15), a taxa de sucesso $S_{k,g}$ será

$$S_{k,g} = \frac{\sum_{g=G-LP}^{G-1} w_g n s_{k,g}}{\sum_{g=G-LP}^{G-1} w_g n s_{k,g} + \sum_{g=G-LP}^{G-1} w_g n f_{k,g}} + \varepsilon; \quad G > LP \quad (17)$$

No decorrer das gerações, ao cruzar um limite M_0 , definido pelo usuário, as gerações são armazenadas e passam a receber os pesos w_g até atingirem sua evolução máxima em G . Conforme a equação (16), ΔG indica o quão forte é o peso da geração atual g . Quando a geração estiver no ponto M_0 , a memória M começará a armazenar as gerações com os seus devidos pesos. Em seguida, aumentará à medida que se aproxima de G , dando mais importância para as gerações mais próximas a melhor solução do problema.



Fig. 2. Fluxograma do EPSOM

Na figura acima, a geração inicial é onde o $g = 1$. Até a geração LP , a probabilidade de escolha de um algoritmo variante do PSO, p_k , é $1/k$. Entre as gerações LP e M_0 , o cálculo da probabilidade p_k é realizado conforme a equação (14), a taxa de sucesso $S_{k,g}$ é calculado de acordo com a equação (15). Após a geração M_0 , o cálculo da taxa de sucesso $S_{k,g}$ é feito com o peso w_g , segundo a equação (17); onde ΔG informa a diferença entre a geração máxima ($G - 1$) e a geração atual (g).

3. Resultados

Para a realização dos testes todos os 7 métodos, EPSOM, EPSO, WPSO, CLPSO, FDRPSO, HPSO-TVAC e LIPS foram implementados em Java.

Foram realizados testes utilizando como base as funções definidas na competição CEC de 2005 [16]. Nesta competição são definidas 25 funções do tipo unimodal, multimodal, expandida e híbrida.

Os testes foram realizados com 10 dimensões e 10 partículas; cada função foi simulada 30 vezes com 10000 gerações.

Segue abaixo os parâmetros dos algoritmos utilizados:

Algoritmo	Coefficientes	Outros
WPSO	$c_1 = c_2 = 2$	$w_f = 0.9;$ $w_i = 0.1$
LIPS	$c = 2$	$X = 0.729;$ $nsize = 3$
CLPSO	$c_1 = 2.25$	$w_f = 0.9;$ $w_i = 0.2$
FDRPSO	$c_1 = c_2 = 2;$ $c_3 = 2$	$w_f = 0.9;$ $w_i = 0.2$
HPSO-TVAC	$c_{1f} = c_{2i} = 2.5;$ $c_{1i} = c_{2f} = 0.5$	$v = 0.5$
EPSO		LP = 300
EPSOM		LP = 300; $M_0 = 400$

A performance do algoritmo proposto EPSOM é testada usando diferentes classes de problemas como unimodal, multimodal, expandida e problemas híbridos. Os resultados obtidos na figura 3 mostram que o algoritmo EPSOM, comparado com outros PSO, foi melhor que as estratégias individuais PSO e que o algoritmo EPSO, porém ficou equiparado com o CLPSO.

Tabela 1. Parâmetros dos algoritmos

Os resultados dos testes exibem o ranking dos algoritmos de acordo com a média das melhores soluções encontrada em cada simulação, sendo 1 o melhor rank e 7 o pior rank. O gráfico abaixo exhibe quantas vezes cada algoritmo obteve o primeiro ou segundo rank nos testes. Os detalhes dos resultados são exibidos na tabela em anexo.

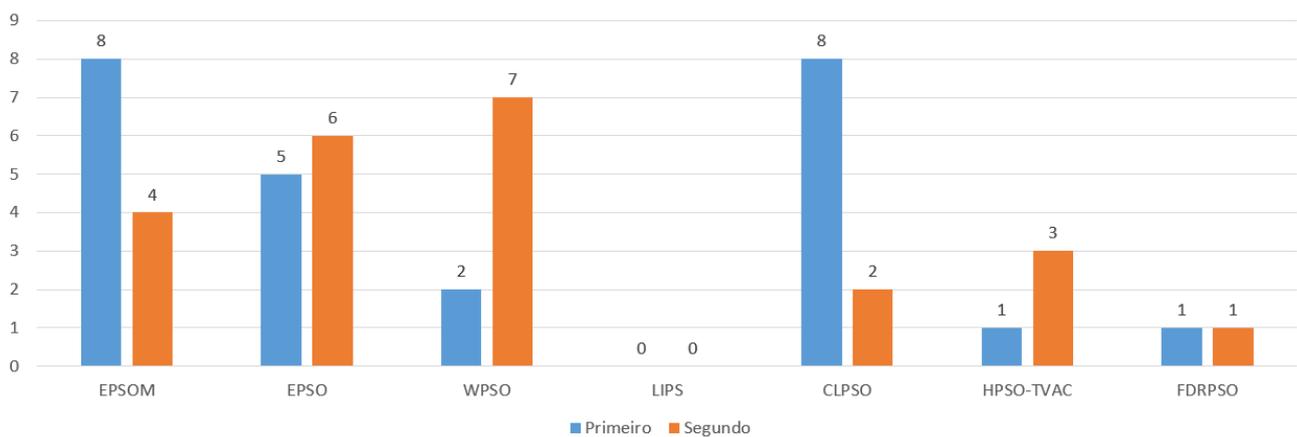


Fig. 3. Ranking de testes – CEC 2005

4. Conclusão

Neste artigo foi apresentado um novo algoritmo que utiliza variados tipos de PSO empregando um componente de memória para cada geração, dando mais relevância para as gerações mais avançadas, ou seja, mais próximas da solução. Aplicando estas variações do PSO, é possível usufruir da diversidade dos mesmos para obter melhores resultados do que estes algoritmos individuais.

Foram realizados 25 testes, contendo 4 tipos diferentes de funções com base no CEC2005, para experimentar a performance do EPSOM em relação com o EPSO e os outros algoritmos. Os resultados obtidos mostram que, na média das simulações, o EPSOM obteve resultados melhores que o EPSO, porém ficou igualado com o CLPSO.

Um passo para expandir este trabalho seria realizar testes com outros algoritmos PSO, como PSOBBO [17] e FST-PSO [18], com o propósito de diversificar ainda mais a população e abranger uma quantidade maior de problemas. Um outro passo seria empregar este algoritmo com dados do mundo real com intenção de validar a aplicação em variados tipos de problemas.

5. Referências

- [1] KHINCHIN, A.; On unimodal distributions. In University of Tomsk. 2(2):1-7, 1938.
- [2] EBERHART, R.; JAMES, K. S. A new optimizer using particle swarm theory. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, pages 39-43. IEEE, 1995.
- [3] DERVIS, K.; BAHRIYE, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*. 39(3): 459-471, 2007.
- [4] DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4): 28-39, 2006.
- [5] HOLLAND, J. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2): 88-105, 1973.
- [6] LINDEN, R. Algoritmos Genéticos. 3a.ed. [S.l.]: Editora Ciência Moderna Ltda., 2014.
- [7] WOLPERT, D.; MACREARY, W. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1): 67-82, 1997.
- [8] LYNN, N.; SUGANTHAN, P. Ensemble particle swarm optimizer. *Applied Soft Computing*. 55:533-548, 2017.
- [9] SHI, Y.; EBERHART, R.; A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation Proceedings*. 69-73, 1998.
- [10] LIANG, J.; QIN, Q.; SUGANTHAN, P. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. 10(3): 281-295, 2006.

- [11] PERAM, T; VEERAMACANENI, K.; MOHAN, C. Fitness-distance-ratio based particle swarm optimization. Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE (2003). 174-181, 2003.
- [12] QU, B.; SUGANTHAN, P.; DAS, S. A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization. 17(3): 387-402, 2012.
- [13] RATNAWEERA, A.; HALGAMUGE, S.; WATSON, H. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. 8(3): 240-255, 2004.
- [14] SHIROKOV, Y.; Algebra of one-dimensional generalized functions. TMF. 39(3): 417-477, 1979.
- [15] BAIK, S. et al. Genetic evolution approach for target movement prediction. Computational Science-ICCS 2004, [S.l.], p.678–681, 2004.
- [16] SUGANTHAN, P.; HANSEN, N.; LIANG, J. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL report. 2005005: 2005, 2005.
- [17] YOGESH, C. K. et al. A new hybrid PSO assisted biogeography-based optimization for emotion and stress recognition from speech signal. Expert Systems with Applications, v. 69, p. 149-158, 2017.
- [18] NOBILE, Marco S. et al. Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization. Swarm and Evolutionary Computation, 2017.

ANEXO A – Tabela de Resultados

Função	EPSOM	EPSO	WPSO	LIPS	CLPSO	HPSOTVAC	FDRPSO
Sphere	Rank: 1 Média: 0,0000E00 Desvio Padrão: 0,0000E00	Rank: 2 Média: 0,0000E00 Desvio Padrão: 0,0000E00	Rank: 4 Média: 1,3263E-14 Desvio Padrão: 2,4453E-14	Rank: 7 Média: 1,8475E04 Desvio Padrão: 3,4946E03	Rank: 3 Média: 0,0000E00 Desvio Padrão: 0,0000E00	Rank: 5 Média: 6,0633E-14 Desvio Padrão: 3,3156E-14	Rank: 6 Média: 1,2016E-01 Desvio Padrão: 3,8820E-02
Schwefel	Rank: 3 Média: 3,9790E-14 Desvio Padrão: 2,6494E-14	Rank: 2 Média: 3,0316E-14 Desvio Padrão: 2,8843E-14	Rank: 5 Média: 8,5645E-06 Desvio Padrão: 1,4465E-05	Rank: 7 Média: 1,8881E04 Desvio Padrão: 4,5828E03	Rank: 1 Média: 1,5158E-14 Desvio Padrão: 2,5567E-14	Rank: 4 Média: 1,8341E-12 Desvio Padrão: 1,7407E-12	Rank: 6 Média: 9,0138E-02 Desvio Padrão: 2,3659E-02
Elliptic	Rank: 1 Média: 1,5024E04 Desvio Padrão: 7,6052E03	Rank: 4 Média: 2,9278E04 Desvio Padrão: 1,7934E04	Rank: 3 Média: 2,8415E04 Desvio Padrão: 2,3942E04	Rank: 7 Média: 4,9412E08 Desvio Padrão: 2,2937E08	Rank: 6 Média: 4,3782E04 Desvio Padrão: 1,2128E04	Rank: 2 Média: 2,6066E04 Desvio Padrão: 1,2392E04	Rank: 5 Média: 3,5385E04 Desvio Padrão: 1,3944E04
Schwefel Noise	Rank: 2 Média: 5,8738E-14 Desvio Padrão: 3,4956E-14	Rank: 1 Média: 3,0316E-14 Desvio Padrão: 3,2477E-14	Rank: 3 Média: 1,5728E-04 Desvio Padrão: 1,8612E-04	Rank: 7 Média: 1,9619E04 Desvio Padrão: 4,3613E03	Rank: 5 Média: 3,4797E03 Desvio Padrão: 9,5043E02	Rank: 6 Média: 1,3915E04 Desvio Padrão: 2,6075E03	Rank: 4 Média: 1,0143E-01 Desvio Padrão: 3,8922E-02
Schwefel Global	Rank: 1 Média: 1,0253E-09 Desvio Padrão: 3,7323E-10	Rank: 3 Média: 2,1715E-09 Desvio Padrão: 1,4757E-09	Rank: 4 Média: 2,4652E-09 Desvio Padrão: 8,6274E-10	Rank: 7 Média: 2,2421E04 Desvio Padrão: 1,7580E03	Rank: 2 Média: 1,3389E-09 Desvio Padrão: 3,3812E-09	Rank: 5 Média: 1,8759E02 Desvio Padrão: 3,7652E02	Rank: 6 Média: 4,5395E00 Desvio Padrão: 2,1080E00
Rosenbrock	Rank: 1 Média: 2,6706E-01 Desvio Padrão: 1,0112E00	Rank: 2 Média: 3,0529E-01 Desvio Padrão: 1,0113E00	Rank: 5 Média: 6,4178E00 Desvio Padrão: 5,3767E00	Rank: 7 Média: 6,1428E09 Desvio Padrão: 2,1415E09	Rank: 4 Média: 4,2959E00 Desvio Padrão: 3,2686E00	Rank: 3 Média: 2,9690E00 Desvio Padrão: 2,2356E00	Rank: 6 Média: 2,0633E01 Desvio Padrão: 6,7969E00
Griewank	Rank: 1 Média: 1,2670E03 Desvio Padrão: 6,3832E-10	Rank: 2 Média: 1,2670E03 Desvio Padrão: 1,4119E-08	Rank: 5 Média: 1,2673E03 Desvio Padrão: 1,9466E-01	Rank: 6 Média: 1,2675E03 Desvio Padrão: 2,4303E-01	Rank: 4 Média: 1,2672E03 Desvio Padrão: 6,9811E-02	Rank: 7 Média: 1,2693E03 Desvio Padrão: 5,8184E00	Rank: 3 Média: 1,2670E03 Desvio Padrão: 1,2795E-03
Ackley	Rank: 4 Média: 2,0300E01 Desvio Padrão: 8,5264E-02	Rank: 5 Média: 2,0302E01 Desvio Padrão: 6,5548E-02	Rank: 2 Média: 2,0248E01 Desvio Padrão: 6,7811E-02	Rank: 6 Média: 2,0323E01 Desvio Padrão: 5,9610E-02	Rank: 3 Média: 2,0278E01 Desvio Padrão: 8,1644E-02	Rank: 1 Média: 2,0143E01 Desvio Padrão: 6,6446E-02	Rank: 7 Média: 2,0325E01 Desvio Padrão: 6,7310E-02
Rastrigin	Rank: 3 Média: 3,9733E01 - Desvio Padrão: 8,3561E00	Rank: 6 Média: 4,3612E01 Desvio Padrão: 7,8303E00	Rank: 2 Média: 3,9566E01 Desvio Padrão: 9,0764E00	Rank: 7 Média: 8,9686E01 Desvio Padrão: 8,2068E00	Rank: 1 Média: 3,2849E01 Desvio Padrão: 7,3134E00	Rank: 5 Média: 4,2883E01 Desvio Padrão: 1,3730E01	Rank: 4 Média: 4,2808E01 Desvio Padrão: 7,0832E00
Rastrigin Rotated	Rank: 1 Média: 7,1971E01 Desvio Padrão: 1,0729E01	Rank: 6 Média: 7,6192E01 Desvio Padrão: 1,3217E01	Rank: 5 Média: 7,5882E01 - Desvio Padrão: 1,8114E01	Rank: 7 Média: 1,4027E02 Desvio Padrão: 1,5528E01	Rank: 4 Média: 7,5592E01 Desvio Padrão: 1,8349E01	Rank: 2 Média: 7,2797E01 Desvio Padrão: 2,2032E01	Rank: 3 Média: 7,4929E01 Desvio Padrão: 1,4767E01
Weierstrass Rotated	Rank: 4 Média: 9,2274E00 Desvio Padrão: 1,3467E00	Rank: 2 Média: 8,8197E00 Desvio Padrão: 1,5432E00	Rank: 1 Média: 7,3307E00 Desvio Padrão: 1,6905E00	Rank: 7 Média: 1,4072E01 Desvio Padrão: 7,5115E-01	Rank: 5 Média: 9,5401E00 Desvio Padrão: 1,2355E00	Rank: 6 Média: 9,7156E00 Desvio Padrão: 1,1994E00	Rank: 3 Média: 8,9744E00 Desvio Padrão: 1,7096E00
Schwefel Multimodal	Rank: 5 Média: 1,6529E03 Desvio Padrão: 3,4101E03	Rank: 6 Média: 1,7281E03 Desvio Padrão: 3,4844E03	Rank: 3 Média: 1,2512E03 Desvio Padrão: 2,7247E03	Rank: 7 Média: 2,6845E05 Desvio Padrão: 8,2016E04	Rank: 1 Média: 1,4500E01 Desvio Padrão: 9,0507E00	Rank: 4 Média: 1,3321E03 Desvio Padrão: 3,5578E03	Rank: 2 Média: 1,8940E02 Desvio Padrão: 4,6823E02
Griewank + Rosenbrock	Rank: 2 Média: 7,6942E-01 Desvio Padrão: 4,3180E-01	Rank: 3 Média: 8,2874E-01 Desvio Padrão: 5,6496E-01	Rank: 4 Média: 8,4107E-01 Desvio Padrão: 4,3123E-01	Rank: 7 Média: 3,4859E01 Desvio Padrão: 1,7051E01	Rank: 1 Média: 4,1100E-01 Desvio Padrão: 2,6865E-01	Rank: 6 Média: 1,6373E00 Desvio Padrão: 1,3084E00	Rank: 5 Média: 1,4195E00 Desvio Padrão: 5,9456E-01
Scaffer	Rank: 3 Média: 3,7588E00 Desvio Padrão: 3,7596E-01	Rank: 5 Média: 3,9133E00 Desvio Padrão: 4,4022E-01	Rank: 2 Média: 3,6922E00 Desvio Padrão: 4,4200E-01	Rank: 7 Média: 4,5186E00 Desvio Padrão: 1,1010E-01	Rank: 4 Média: 3,8264E00 Desvio Padrão: 2,0001E-01	Rank: 6 Média: 4,0193E00 Desvio Padrão: 2,9113E-01	Rank: 1 Média: 3,6222E00 Desvio Padrão: 3,2662E-01
Hybrid Composition Function	Rank: 2 Média: 7,0753E02 Desvio Padrão: 1,5501E02	Rank: 3 Média: 7,1303E02 Desvio Padrão: 1,3559E02	Rank: 4 Média: 7,5550E02 Desvio Padrão: 1,4386E02	Rank: 7 Média: 1,0883E03 Desvio Padrão: 1,2142E02	Rank: 1 Média: 6,7030E02 Desvio Padrão: 1,2077E02	Rank: 5 Média: 7,8514E02 Desvio Padrão: 1,9445E02	Rank: 6 Média: 8,1620E02 Desvio Padrão: 1,1947E02
Rotated Hybrid Composition Function	Rank: 4 Média: 5,7548E02 Desvio Padrão: 1,3320E02	Rank: 5 Média: 5,7678E02 Desvio Padrão: 1,2166E02	Rank: 2 Média: 4,6187E02 Desvio Padrão: 7,6757E01	Rank: 7 Média: 1,0293E03 Desvio Padrão: 2,1468E02	Rank: 1 Média: 4,2235E02 Desvio Padrão: 4,7875E01	Rank: 6 Média: 6,5603E02 Desvio Padrão: 1,1743E02	Rank: 3 Média: 5,1716E02 Desvio Padrão: 6,2522E01
Rotated Hybrid Composition Function Noise	Rank: 5 Média: 6,3282E02 Desvio Padrão: 1,1292E02	Rank: 3 Média: 5,8234E02 Desvio Padrão: 1,1555E02	Rank: 2 Média: 5,1525E02 Desvio Padrão: 1,2310E02	Rank: 7 Média: 1,1416E03 Desvio Padrão: 1,5716E02	Rank: 1 Média: 5,0276E02 Desvio Padrão: 6,2341E01	Rank: 6 Média: 7,5373E02 Desvio Padrão: 1,8130E02	Rank: 4 Média: 5,9044E02 Desvio Padrão: 1,1866E02
Rotated Hybrid Composition Function 2	Rank: 2 Média: 9,0000E02 Desvio Padrão: 1,0252E-02	Rank: 5 Média: 9,1327E02 Desvio Padrão: 7,2704E01	Rank: 6 Média: 1,0011E03 Desvio Padrão: 1,3202E02	Rank: 3 Média: 9,0006E02 Desvio Padrão: 1,5194E-02	Rank: 1 Média: 9,0000E02 Desvio Padrão: 2,1136E-03	Rank: 7 Média: 1,0253E03 Desvio Padrão: 2,3854E02	Rank: 4 Média: 9,0768E02 Desvio Padrão: 4,1269E01
Rotated Hybrid Composition Function Global	Rank: 2 Média: 9,0000E02 Desvio Padrão: 2,7077E-04	Rank: 1 Média: 9,0000E02 Desvio Padrão: 0,0000E00	Rank: 6 Média: 1,0140E03 Desvio Padrão: 1,6604E02	Rank: 4 Média: 9,0006E02 Desvio Padrão: 1,5230E-02	Rank: 3 Média: 9,0000E02 Desvio Padrão: 2,7704E-03	Rank: 7 Média: 1,1271E03 Desvio Padrão: 2,5678E02	Rank: 5 Média: 9,1670E02 Desvio Padrão: 9,0548E01
Rotated Hybrid Composition Function with Global Bounds	Rank: 1 Média: 9,0000E02 Desvio Padrão: 0,0000E00	Rank: 5 Média: 9,1891E02 Desvio Padrão: 7,2128E01	Rank: 6 Média: 1,0436E03 Desvio Padrão: 1,3989E02	Rank: 3 Média: 9,0006E02 Desvio Padrão: 1,1271E-02	Rank: 2 Média: 9,0000E02 Desvio Padrão: 1,3492E-03	Rank: 7 Média: 1,0979E03 Desvio Padrão: 2,5253E02	Rank: 4 Média: 9,0016E02 Desvio Padrão: 3,8424E-02
Rotated Hybrid Composition Function 3	Rank: 5 Média: 1,4173E03 Desvio Padrão: 5,1112E01	Rank: 1 Média: 1,3716E03 Desvio Padrão: 1,9241E02	Rank: 2 Média: 1,3748E03 Desvio Padrão: 6,5972E01	Rank: 7 Média: 1,5363E03 Desvio Padrão: 4,7907E01	Rank: 3 Média: 1,3883E03 Desvio Padrão: 5,4288E01	Rank: 6 Média: 1,4201E03 Desvio Padrão: 5,6407E01	Rank: 4 Média: 1,4043E03 Desvio Padrão: 5,8530E01
Rotated Hybrid Composition Function High	Rank: 3 Média: 1,3715E03 Desvio Padrão: 9,3513E01	Rank: 2 Média: 1,2948E03 Desvio Padrão: 2,5039E02	Rank: 1 Média: 1,1892E03 Desvio Padrão: 2,0301E02	Rank: 7 Média: 1,6843E03 Desvio Padrão: 1,1408E02	Rank: 6 Média: 1,4214E03 Desvio Padrão: 7,9436E01	Rank: 4 Média: 1,3725E03 Desvio Padrão: 1,2296E02	Rank: 5 Média: 1,4128E03 Desvio Padrão: 2,2766E02
Non Continuous Rotated Hybrid Composition Function	Rank: 1 Média: 1,5235E03 Desvio Padrão: 4,2811E01	Rank: 4 Média: 1,5380E03 Desvio Padrão: 5,1130E01	Rank: 5 Média: 1,5412E03 Desvio Padrão: 5,1642E01	Rank: 6 Média: 1,5480E03 Desvio Padrão: 4,3480E01	Rank: 3 Média: 1,5339E03 Desvio Padrão: 4,8027E01	Rank: 2 Média: 1,5315E03 Desvio Padrão: 6,9692E01	Rank: 7 Média: 1,5522E03 Desvio Padrão: 5,7565E01
Rotated Hybrid Composition Function 4	Rank: 3 Média: 1,4167E03 Desvio Padrão: 3,5777E01	Rank: 1 Média: 1,3929E03 Desvio Padrão: 1,9035E02	Rank: 2 Média: 1,5346E03 Desvio Padrão: 3,0246E01	Rank: 7 Média: 1,5346E03 Desvio Padrão: 3,3096E01	Rank: 6 Média: 1,4539E03 Desvio Padrão: 3,1831E01	Rank: 5 Média: 1,4340E03 Desvio Padrão: 2,5726E01	Rank: 4 Média: 1,4228E03 Desvio Padrão: 2,3050E01
Rotated Hybrid Composition Function Bounds	Rank: 5 Média: 1,7716E03 Desvio Padrão: 2,2326E01	Rank: 1 Média: 1,7628E03 Desvio Padrão: 2,0715E01	Rank: 6 Média: 1,7786E03 Desvio Padrão: 1,8269E01	Rank: 3 Média: 1,7684E03 Desvio Padrão: 3,2010E01	Rank: 2 Média: 1,7683E03 Desvio Padrão: 2,0331E01	Rank: 4 Média: 1,7713E03 Desvio Padrão: 1,8460E01	Rank: 7 Média: 1,7807E03 Desvio Padrão: 2,3794E01
Melhor / Segundo	8 / 4	5 / 6	2 / 7	0 / 0	8 / 2	1 / 3	1 / 1

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 14 de dezembro de 2017, às 11:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **RAUL VITOR DA SILVA BARRETO**, orientado pelo professor **Lizandro Nunes**, sob título **Combinação de Otimizadores por enxame de partículas com memória**, a banca composta pelos professores:

Lizandro Nunes

Paulo Hugo Espírito Santo Lima

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 9,25 (NOVE VÍRGULA VINTE E CINCO)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá _____ dias para entrega da versão final da monografia a contar da data deste documento.

Lizandro Nunes da Silva

LIZANDRO NUNES

Paulo Hugo Espírito Santo Lima

PAULO HUGO ESPÍRITO SANTO LIMA

* Este documento deverá ser encadernado juntamente com a monografia em versão final.