



**Primeira Lista de Exercícios**  
**Otimização por enxame de partículas (PSO) e Colonia artificial de abelhas (ABC)**  
**Data de entrega (22 de Setembro de 2017)**

**Nota:** A lista é individual. Usar template LaTeX da IEEE conference. Enviar via moodle os arquivos Matlab e o relatório em PDF em uma pasta zipada chamada “nome\_sobrenome”

**Questão única:** Implementar no Matlab o PSO canônico e o algoritmo ABC para minimizar as funções *benchmark Esfera*, *Quadric (ou rotated hyper ellipsoid)*, *Rosenbrock*, *Rastrigin*, *Ackley*, *Schwefel* e *Michalewicz*. No seguinte link pode-se conferir a posição do mínimo global de cada função e o respectivo valor da função custo:

[http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar\\_files/TestGO\\_files/Page364.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm)

$$\text{Griewank: } f_1(\vec{x}) = 1 + \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right); x_i \in [-512, 512]$$

$$\text{Rastrigin: } f_2(\vec{x}) = \sum_{i=1}^N \left( x_i^2 - 10 \cos(2\pi x_i) + 10 \right); x_i \in [-8.0, 8.0]$$

$$\text{Rosenbrock: } f_3(\vec{x}) = \sum_{i=1}^{N/2} 100 \cdot (x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2; x_i \in [-8.0, 8.0]$$

$$\text{Ackley: } f_4(\vec{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^N x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^N \cos(2\pi x_i)\right) + 20 + e; x_i \in [-32.768, 32.768]$$

$$\text{Schwefel: } f_5(\vec{x}) = 418.9829 N - \sum_{i=1}^N x_i \sin\left(\sqrt{|x_i|}\right); x_i \in [-500, 500]$$

$$\text{Michalewicz: } f_6(\vec{x}) = - \sum_{i=1}^N \sin(x_i) \sin\left(\frac{ix_i^2}{\pi}\right)^{2m}; x_i \in [0, \pi]; m = 10$$

Configure os algoritmos segundo as seguintes instruções:

- 1) Usar três tamanhos diferentes de enxame com valores entre  $S=10, 15, 20$ .
- 2) Usar cinco valores diferentes de dimensionalidade dos problemas. Escolha valores no intervalo  $N=[6, 22]$ . Use um número par de dimensões.
- 3) Repetir cada experimento 32 vezes, inicializando o enxame em diferentes posições aleatórias.
- 4) Configure os parâmetros dos algoritmos da seguinte maneira:
  - Usar como critério de parada o número máximo de iterações.
  - Número máximo de iterações = 1000
  - $c_1=c_2=2.05$  (PSO)
  - $w$ : decresce linearmente na faixa de valores  $[0.9 \text{ a } 0.1]$  (PSO)
  - $v_{inicial} = v_{max}/3$  (PSO)
  - Número de iterações para enviar abelhas escoteiras  $limit = 20$  (ABC)
  - $threshold = 0.01$  para todas as funções *benchmark*, exceto para a função *Michalewicz* cujo  $threshold$  deve ser achado mediante uma regra de três simples. Neste primeiro exercício não vamos usar o  $threshold$  como critério de parada. Apenas será usado para verificar se o algoritmo alcançou um valor mínimo desejado para a função custo (*goal*).
- 5) Para cada experimento vamos usar o melhor valor da função custo (valor da função custo após as 1000 iterações) para calcular a média, mediana, desvio padrão, valor mínimo e número de acertos (*goals*) entre os 32 experimentos. Apresentar a solução do problema de otimização (posição do ponto mínimo encontrado),



entre os 32 experimentos. Observe-se que dos 32 experimentos, aquele que apresente o valor mínimo da função custo representa a melhor solução do problema. Para cada função *benchmark* deve-se apresentar a seguinte tabela contendo os resultados estatísticos.

Tabela 1. Algoritmo \_\_\_\_\_. Resultados de convergência para a função \_\_\_\_\_ (32 runs).

		Média	Mediana	Mínimo	Desvio Padrão	goals/32
S=10	N=					
	N=					
	N=					
	N=					
	N=					
S=15	N=					
	N=					
	N=					
	N=					
	N=					
S=20	N=					
	N=					
	N=					
	N=					
	N=					

6) Dica1: implemente cada *benchmark* como uma função  $f(\mathbf{x}, N)$  que receba como parâmetros a posição da partícula ( $\mathbf{x}$ ) e o número de dimensões do problema ( $N$ ).

7) Dica2: implemente os *scripts* no Matlab de forma a automatizar o processo de coleta de dados.