



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO**  
**CURSO DE ENGENHARIA MECATRÔNICA**

**RELATÓRIO DA 3ª UNIDADE DE**  
**CONTROLE FUZZY DE SISTEMAS DINÂMICOS**  
**OTIMIZAÇÃO NUMÉRICA**

FERNANDO LEANDRO FERNANDES (20150146106)

ROSENILDO PEREIRA AGUIAR FURTADO (20150146554)

TIAGO BATISTA SILVA SOUSA (20150146198)

Natal-RN  
2016

FERNANDO LEANDRO FERNANDES (20150146106)

ROSENILDO PEREIRA AGUIAR FURTADO (20150146554)

TIAGO BATISTA SILVA SOUSA (20150146198)

**RELATÓRIO DA 3ª UNIDADE DE  
CONTROLE FUZZY DE SISTEMAS DINÂMICOS  
OTIMIZAÇÃO NUMÉRICA**

Terceiro Relatório apresentado à disciplina de Controle Fuzzy de Sistemas dinâmicos, correspondente à avaliação da 3ª unidade do semestre 2016.2 do 9º período do curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte, sob orientação do mestrando **Eng. Alcemy Gabriel Vitor Severino.**

Professor **Dr. Fábio Meneghetti Ugulino de Araújo.**

Natal-RN  
2016

## RESUMO

Esta atividade é componente do conjunto de avaliações da 3a. unidade da disciplina Controle *Fuzzy* de Sistemas Dinâmicos (DCA0126), do período 2016.2, do curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte.

O presente relatório descreve métodos de otimização numérica(maximização ou minimização) de funções bidimensionais utilizando os métodos "algoritmo genético"e "otimização por enxame de partículas". Os algoritmos de otimização foram implementados no MATLAB R2014a e aplicados a algumas das funções disponíveis no site *Virtual Test Functions and Datasets* disponível no site <http://www.sfu.ca/~ssurjano/optimization.html> (acesso em 29 de novembro de 2016).

**Palavras-chave:** *Otimização Numérica, Algoritmo Genético, GA, Enxame de partículas, PSO.*

## Lista de Figuras

1	Fluxograma do algoritmo genético. . . . .	9
2	Fluxograma do algoritmo do enxame de partículas. . . . .	10
3	Gráfico da função Bukin no. 6. . . . .	12
4	Gráfico da função Ackley. . . . .	12
5	Gráfico da função Bohachevsky no. 1. . . . .	13

# Sumário

<b>1</b>	<b>Introdução</b>	<b>6</b>
<b>2</b>	<b>Referencial Teórico</b>	<b>7</b>
2.1	Algoritmo Genético . . . . .	7
2.1.1	Representação dos indivíduos . . . . .	7
2.1.2	Métodos de seleção . . . . .	7
2.1.3	Cruzamento . . . . .	8
2.1.4	Mutação . . . . .	8
2.1.5	Fluxograma . . . . .	8
2.2	Enxame de partículas . . . . .	9
2.2.1	O algoritmo . . . . .	9
2.2.2	Fluxograma . . . . .	10
2.3	Comparação entre GA e PSO . . . . .	10
<b>3</b>	<b>Metodologia</b>	<b>11</b>
3.1	Funções para otimização . . . . .	11
3.1.1	Função <i>Bukin</i> no. 6 . . . . .	11
3.1.2	Função <i>Ackley</i> . . . . .	11
3.1.3	Função <i>Bohachevsky</i> no. 1 . . . . .	13
<b>4</b>	<b>Resultados</b>	<b>13</b>
4.1	Algoritmo Genético . . . . .	13
4.2	Enxame de Partículas . . . . .	14
<b>5</b>	<b>Conclusões</b>	<b>14</b>

# 1 Introdução

Os métodos de otimização numérica consistem algoritmos de busca iterativa por pontos de máximo ou mínimo de funções nas quais a aplicação de métodos analíticos é ineficiente, custosa ou mesmo impraticável.

Os algoritmos numéricos para solução de problemas de otimização são essencialmente classificados em métodos de programação matemática e métodos probabilísticos. A diferença essencial dos métodos de programação matemática para os métodos probabilísticos é que os últimos procuram encontrar o mínimo global do problema de otimização evitando os mínimos locais. Já os métodos de programação matemática fornecem um mínimo local.

Os métodos probabilísticos, como o próprio nome sugere, se utilizam de um processo de busca randômica guiados por decisões probabilísticas para obter o mínimo global. Além disso, os métodos probabilísticos são também ferramentas poderosas para problemas com variáveis discretas.

Dentre os métodos probabilísticos existentes, destacam-se o método do algoritmo genético (*genetic algorithm – GA*), que simula a evolução biológica natural das espécies para procurar o indivíduo (possível solução) com a melhor adaptação e o enxame de partículas (*particle swarm optimization – PSO*).

Para casos de minimização, a adaptação de cada indivíduo é calculada como o inverso do valor da função a ser otimizada em cada indivíduo, que é representado por um par ordenado bidimensional.

Nesta atividade exploraremos especificamente o algoritmo genético e o enxame de partículas na determinação de mínimos e máximos de funções exóticas.

## 2 Referencial Teórico

Apresentamos a seguir os conceitos envolvidos na teoria dos algoritmos de otimização numérica implementados nesta atividade, especialmente o algoritmo genético e algoritmo de otimização por enxame de partículas.

### 2.1 Algoritmo Genético

A classe de algoritmos denominados genéticos (ou GA, do inglês *genetic algorithm*) consistem em uma mimetização do processo evolucionário das espécies [Linden, 2008], basicamente, pela reprodução do processo de evolução biológica das espécies para otimizar problemas diversos, incluindo problemas matemáticos.

O algoritmo teve seu início na primeira metade da década de 1950, resultado do trabalho do matemático norueguês-italiano Nils Aall Barricelli, em Princeton, New Jersey [Kennedy et al, 1995]. Apesar de sua publicação ser de 1954, os fundamentos do algoritmo genético não entraram em evidência até os anos 1960, quando o seu estudo foi vastamente difundido. Também nos anos 1950, o inglês Alex Fraser publicou uma série de artigos sobre o assunto, por exemplo [Fraser, 1957] e [Fraser, 1970]. Genericamente, o algoritmo gera uma população aleatória e com o passar das gerações, a população converge para uma solução de mínimo/máximo global.

Os algoritmos genéticos são técnicas probabilísticas que manipulam um conjunto numeroso de indivíduos (população), sendo uma heurística de busca no espaço de estados. Assim, mesmo com uma população inicial e parâmetros de configuração idênticos, é provável que sejam encontrados resultados diferentes ainda que convergentes a um valor comum.

#### 2.1.1 Representação dos indivíduos

Cada indivíduo é representado por  $n$  valores, onde  $n$  é a dimensão de entrada da função-problema. Esses valores podem ser utilizados nas bases binária ou decimal. Os indivíduos têm um "valor de adaptação ao meio", que é chamado de adaptação que representa o quão próximo da solução aquele indivíduo está. Para casos de maximização, a adaptação é igual ao valor da função naquele ponto (indivíduo), para minimização, o inverso do valor da função avaliada no ponto. O algoritmo selecionará quais indivíduos passarão seus genes para as próximas gerações através dos métodos de seleção descritos na subseção 2.1.2.

#### 2.1.2 Métodos de seleção

As GAs, diferentemente das buscas aleatórias (*random walk*), são buscas orientadas por informações pertinentes ao problema que direcionam sua busca através do mecanismo de seleção, novamente, assemelhado ao processo natural. Alguns métodos de seleção propostos para cumprir o papel da escolha dos melhores indivíduos segundo algum critério de avaliação e que, em tese, estariam mais aptos a otimizar o problema proposto.

A seguir são apresentados alguns desses métodos:

- **ROLETA:** O método da roleta consiste em selecionar o indivíduo baseado na sua adaptação em relação à população total. Dessa forma, o indivíduo(solução) mais adaptado terá uma maior probabilidade de passar no teste e gerar filhos.
- **TORNEIO:** O método do torneio seleciona apenas os melhores indivíduos, ou seja, aqueles pontos que estão mais próximos da solução desejada(mínimo ou máximo). Ele é particularmente eficiente em funções que possuem poucos ou nenhum mínimo local.
- **SORTEIO:** O método do sorteio seleciona aleatoriamente indivíduos na população sem levar em conta a sua adaptação. Por não ser consistente, esse método não é muito utilizado.

### 2.1.3 Cruzamento

Após a subpopulação de pais ser definida pela aplicação de um dos métodos descritos, o cruzamento (troca de material genético) entre os indivíduos selecionados como pais, será realizado nos pares de cromossomos(no caso  $x_1$  e  $x_2$ ). Para que isso ocorra, será selecionado um número aleatório que representa a parte de material genético (coordenadas  $x_1$  e  $x_2$ ) que cada filho herdará de seus pais.

Técnicas de cruzamento:

- Cruzamento na base binária: Neste cruzamento (cruzamento binário), cada indivíduo é representado por  $n$  palavras de  $m$  bits e a troca de material genético é feita sorteando um número inteiro, chamado de *flag*, que representa quantos bits de cada parâmetro do indivíduo pai serão mesclados com o restante de bits do indivíduo mãe. Isto é, se  $m = 8$  e  $flag = 3$ , os 3 primeiros bits do pai e os 5 últimos bits da mãe serão usados para gerar um filho, e os bits restantes são utilizados para gerar o segundo filho.
- Cruzamento na base decimal: De forma semelhante, o cruzamento na base decimal (cruzamento decimal) é feito atribuindo um valor de *flag* percentual e aplicando o mesmo raciocínio do cruzamento binário ao valor decimal dos pais para gerar os filhos.

### 2.1.4 Mutação

O técnica de mutação é utilizada para gerar um indivíduo com características diversa em relação ao resto da população. Isso é feito com intuito de adicionar algum nível de aleatoriedade ao processo e, assim, evitar que o algoritmo fique confinado em um mínimo local.

### 2.1.5 Fluxograma

O fluxograma do algoritmo genético é o apresentado na figura 1.



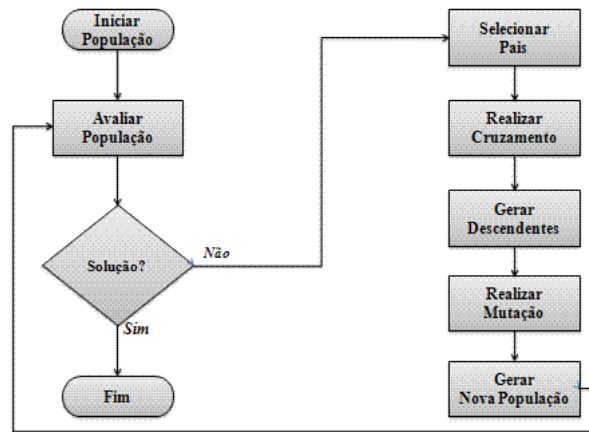


Figura 1: Fluxograma do algoritmo genético.

## 2.2 Enxame de partículas

A técnica de otimização por enxame de partículas, do inglês *particle swarm optimization (PSO)*, foi concebida pelos sócio-psicólogo James Kennedy e pelo engenheiro eletricitista Russel Eberhart, ambos americanos, inspirada no comportamento de uma revoada de pássaros, e seu primeiro intuito era o estudo em experimentos sociais [Kennedy et al, 1995]. A otimização é feita através da aplicação de uma população de partículas e que são movidas no espaço de entrada da função à procura de uma solução de máximo ou mínimo.

A categoria de algoritmos computacionais de otimização por enxame de partículas são caracterizados pela forma iterativa que usam para melhorar uma partícula-solução tendo como métrica a própria função a ser otimizada [Yang et al, 2013]. Esses algoritmos fazem parte da família de algoritmos que aplicam inteligência de enxame *swarm intelligence*.

### 2.2.1 O algoritmo

Como mencionado anteriormente, o enxame de partículas simula o comportamento de uma revoada de pássaros. Ele usa o que aprende do cenário e o usa para resolver os problemas de otimização. No PSO cada solução, ou partícula, é um "pássaro" no espaço de solução. Todas as partículas tem valores de *fitness* que são calculadas da função a ser otimizada e tem velocidades que direcionam o "voo" das partículas. As partículas se movem pelo espaço do problema seguindo as partículas ótimas encontradas.

O algoritmo é inicializado com um grupo de partículas (soluções) aleatórias e, a partir daí, busca pela solução ótima atualizando as gerações. Em cada iteração, cada partícula é atualizada pela melhor solução (*fitness*) que ela alcançou, denominada *pbest*, assim como o melhor solução entre todas apresentadas pelo enxame, o *gbest*, conforme a equação 1.

$$v_p[\cdot] = w * v_p[\cdot] + c_1 * rand() * (pbest[\cdot] - p[\cdot]) + c_2 * rand() * (gbest - p[\cdot]) \quad (1)$$

$$p[\cdot] = p[\cdot] + v_p[\cdot] \quad (2)$$

onde  $w$  é a inércia da partícula,  $v_p$  é a velocidade da partícula,  $c_1$  e  $c_2$  são constantes de aprendizado e  $p$  é a própria partícula.

Desta forma, são considerados tanto a contribuição individual de cada elemento em separado, como o resultado coletivo do enxame.

### 2.2.2 Fluxograma

O fluxograma do algoritmo do enxame de partículas é o apresentado na figura 2.

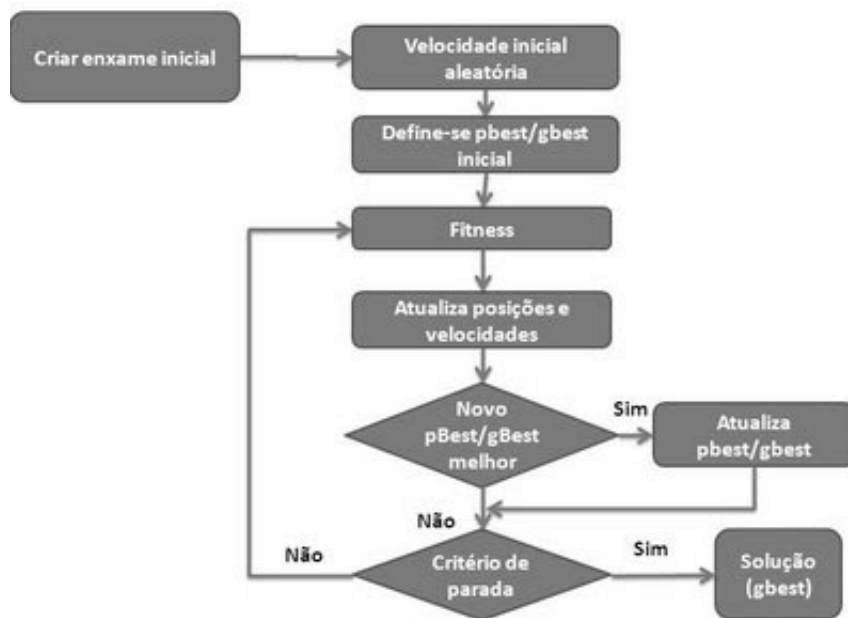


Figura 2: Fluxograma do algoritmo do enxame de partículas.

## 2.3 Comparação entre GA e PSO

Ambas as técnicas, GA e PSO, seguem o procedimento abaixo:

1. Geração aleatória de uma população inicial
2. Cômputo do *fitness* para cada indivíduo. Isso dependerá da distância para o objetivo
3. Reprodução da população com base nos valores de *fitness* de cada indivíduo
4. Se os objetivos foram alcançados, então pare. Se não, volte ao passo 2.

Do procedimento acima podemos afirmar que o PSO compartilha várias características com o GA. Ambos iniciam com uma população gerada aleatoriamente, ambos tem valores de aptidão (*fitness*) para avaliar a população. Ambos atualizam a população e buscam pelo valor ótimo com técnicas ramdômicas. Nenhum dos algoritmos garante sucesso.

Entretanto, o PSO não tem operações "genéticas" como cruzamento e mutação. Nele, as partículas atualizam a si mesmas com a velocidade interna. Elas também tem memória, característica importante do algoritmo.

### 3 Metodologia

A implementação dos algoritmos de otimização apresentados neste relatório foi desenvolvida em *scripts* da ferramenta de simulação MATLAB [Moore, 2012], versão R2014a.

Os testes de execução foram realizados 100 vezes para cada função e algoritmo. Para cada caso foram calculados a mediana, menor e maior valor da posição do indivíduo/agente e o desvio-padrão dos valores da função (solução) obtidos para que ser averiguasse a convergência dos resultados apresentados.

#### 3.1 Funções para otimização

As funções escolhidas para realizar otimização foram retiradas do site *Virtual Test Functions and Datasets*<sup>1</sup>. As funções escolhidas são apresentadas a seguir.

##### 3.1.1 Função *Bukin* no. 6

A sexta função *Bukin* possui diversos mínimos locais, todos recaem em uma fenda. A expressão da função é a apresentada pela equação 3.

$$f(x) = 100\sqrt{|x_2 - 0.01x_1^2|} + 0.01|x_1 + 10| \quad (3)$$

cujo domínio de entrada é dado pelo retângulo  $x_1 \in [-15, 5]$ ,  $x_2 \in [-3, 3]$  e tem mínimo global em  $f(x^*) = 0$ , para  $x^* = (-10, 1)$ .

O gráfico da função é o apresentado na figura 3.

##### 3.1.2 Função *Ackley*

A função *Ackley* é amplamente usada teste de algoritmos de otimização. Em sua forma bidimensional, como mostrada no gráfico abaixo, é caracterizada por uma região externa aparentemente plana e um grande cone no centro. A função possui diversos mínimos locais que podem capturar a atenção dos algoritmos de otimização, especialmente os de escalada.

A expressão da função é a apresentada pela equação 4.

---

<sup>1</sup><http://www.sfu.ca/~ssurjano/optimization.html>, acesso em 29 de novembro de 2016

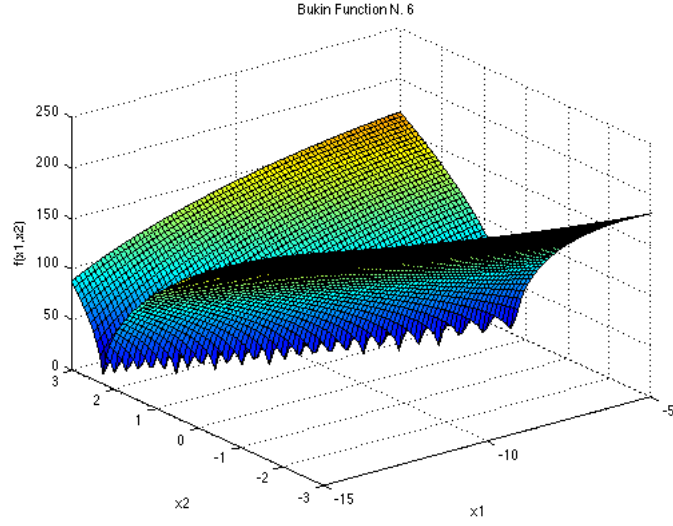


Figura 3: Gráfico da função Bukin no. 6.

$$f(x) = -a \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1) \quad (4)$$

cuja dimensão é  $d$ . O domínio de entrada é dado pelo hipercubo  $x_i \in [-32.768, 32.768]$  e tem mínimo global em  $f(x^*) = 0$ , para  $x^* = (0, \dots, 0)$ . Foram usadas as constantes recomendadas  $a = 20$ ,  $b = 0.2$  e  $c = 2\pi$ .

O gráfico da função é o apresentado na figura 4.

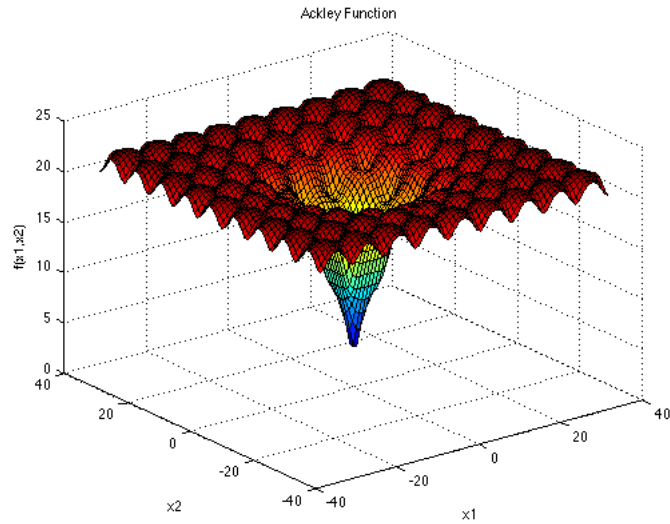


Figura 4: Gráfico da função Ackley.

### 3.1.3 Função *Bohachevsky* no. 1

As funções *Bohachevsky* tem, todas, o mesmo formato de cuba. A expressão da função escolhida é a apresentada pela equação 5.

$$f_1(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7 \quad (5)$$

cujo domínio de entrada é dado pelo quadrado  $x_i \in [-100, 100]$ , para todo  $i = 1, 2$  e tem mínimo global em  $f_j(x^*) = 0$ , para  $x^* = (0, 0)$ , para todo  $j = 1, 2, 3$ .

O gráfico da função é o apresentado na figura 5.

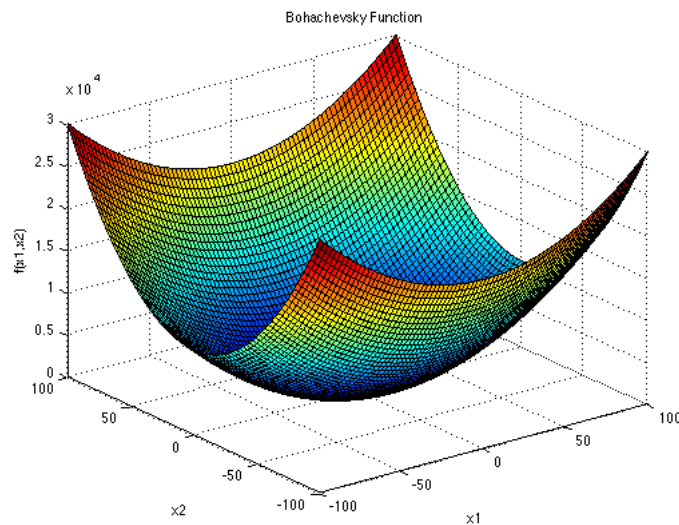


Figura 5: Gráfico da função *Bohachevsky* no. 1.

## 4 Resultados

### 4.1 Algoritmo Genético

Na tabela 1 são apresentados os resultados obtidos das médias e desvios-padrão dos melhores indivíduos obtidos após a evolução de 100 gerações para as funções de otimização escolhidas.

Tabela 1: Resultados da otimização com algoritmo genético.

Função	$x_1$		$x_2$		$f(x)$
	Intervalo	$\bar{x}_1 \pm \sigma_{x_1}$	Intervalo	$\bar{x}_2 \pm \sigma_{x_2}$	
Bukin #6	[-15, 5]	-9,6789 $\pm$ 1,7845	[-3, 3]	0,9683 $\pm$ 0,3439	0,0434 $\pm$ 0,2430
Ackley	[-40, 40]	-0,0216 $\pm$ 0,2605	[-40, 40]	0,0408 $\pm$ 0,2846	0,0434 $\pm$ 0,2430
Bohachevsky #1	[-100, 100]	-0,0329 $\pm$ 0,3797	[-100, 100]	0,0065 $\pm$ 0,2842	0,3348 $\pm$ 1,0575

Os parâmetros utilizados no algoritmo genético, para todas as funções de otimização, foram: população de 100 indivíduos e 100 gerações.

## 4.2 Enxame de Partículas

Na tabela 2 são apresentados os resultados obtidos das médias e desvios-padrão das melhores partículas obtidas após a evolução de 50 enxames para as funções de otimização escolhidas.

Tabela 2: Resultados da otimização com enxame de partículas.

Função	$x_1$		$x_2$		$f(x)$
	Intervalo	$\bar{x}_1 \pm \sigma_{x_1}$	Intervalo	$\bar{x}_2 \pm \sigma_{x_2}$	
Bukin #6	[-15, 5]	-10,3213 $\pm$ 3,1560	[-3, 3]	1,0653 $\pm$ 0,6262	0 $\pm$ 0,0163
Ackley	[-40, 40]	7,7230E-11 $\pm$ 0,2234	[-40, 40]	-1,3569E-10 $\pm$ 0,0816	0 $\pm$ 0,1563
Bohachevsky #1	[-100, 100]	4,1769E-10 $\pm$ 0,2241	[-100, 100]	-1,0877E-10 $\pm$ 0,1322	0 $\pm$ 0,1784

Os parâmetros utilizados no enxame de partículas, para todas as funções de otimização, foram:  $w = 0.05$ ,  $c_1 = 1.4$ ,  $c_2 = 1.4$ ,  $particulas = 30$  e  $enxames = 50$ .

## 5 Conclusões

Os resultados obtidos foram consistentes com as informações de solução para as funções escolhidas tendo ambos algoritmos alcançado a convergência necessária para que se pudessem constatar a precisão e a exatidão dos resultados obtidos, corroborando para a conclusão de corretude e prova satisfatória de funcionamento das técnicas.

Observou-se que o algoritmo genético gerou resultados consistentes, mas menos precisos e com velocidade de convergência menor que as obtidas com o enxame de partículas. Pode-se citar como exemplo das observação feita a ordem de grandeza dos resultados de média das soluções obtidas,  $x_1$  e  $x_2$ , que ficaram na casa de  $10^{-2}$  para o GA, enquanto que para o PSO ficaram em ordem de grandeza significativamente menor, aproximadamente  $10^{-10}$  e  $10^{-11}$  nas funções *Ackley* e *Bohachevsky*.

A experiência de implementar e sintonizar as configuração de ambos algoritmos se mostrou mais desafiador para o algoritmo genético dada a característica da seleção e do cruzamento, principalmente. As alterações súbtas das informações genéticas das novas populações geravam flutuações inesperadas que levou a equipe a reformular a estratégia de aborgagem do problema de condificação de código genético binário para o modelo de pesos. Já o algoritmo enxame de partículas (PSO) se apresentou mais simples de controlar através dos parâmetros de configuração. A convergência para o valor ótimo da solução ocorreu em todas as execuções do algoritmo.

Comparado com os algoritmos genéricos (GA), o mecanismo de compartilhamento no PSO é substancialmente diferente. No GA, os cromossomos compartilham informações uns com os outros. Portanto, toda a população se move como um grupo em direção a uma área ótima. No PSO, apenas o gbest presta informações aos demais, assim todas as partículas tendem a convergir para a melhor solução rapidamente, na maioria dos casos.

## Referências

- [Linden, 2008] LINDEN, Ricardo. (2008). *Algoritmos genéticos: uma importante ferramenta da Inteligência Computacional*. 2a. edição. Rio de Janeiro. Brasport.
- [Fraser, 1957] FRASER, Alex. (1957). *Simulation of genetic systems by automatic digital computers. I. Introduction*. Aust. J. Biol. Sci. 10: 484–491.
- [Fraser, 1970] FRASER, Alex. BURNELL, Donald. (1970). *Computer Models in Genetics*. New York: McGraw-Hill. ISBN 0-07-021904-4.
- [Kennedy et al, 1995] Kennedy, James, Eberhart, Russell. (1995). *Particle swarm optimization*. IEEE int'l conf. on neural networks Vol. IV, pp. 1942-1948. IEEE service center, Piscataway, NJ.
- [Yang et al, 2013] YANG, Xin-She. (2013). *Swarm Intelligence and Bio-Inspired Computation. Theory and Applications*. 1st. ed. Elsevier.
- [Moore, 2012] MOORE, Holly. (2012). *MATLAB® for engineers*. 3rd. edition. Pearson.
- [Santos, 2013] SANTOS, Tássio Naia dos. (2013). LaTeXação. Mais uma apostila de L<sup>A</sup>T<sub>E</sub>X. disponível em <https://www.ime.usp.br/tassio/arquivo/latex/apostila.pdf>. acesso em 28/11/2016.