XXI ENMC
ENCONTRO NACIONAL DE MODELAGEM COMPUTACIONAL
IX ECTM
ENCONTRO DE CIÊNCIA E TECNOLOGIA DE MATERIAIS

08 a 11 de Outubro de 2018
Instituto Federal Fluminense
Búzios - RJ

ANÁLISE COMPARATIVA SOBRE O DESEMPENHO DO MÉTODO DE OTIMIZAÇÃO ALCATEIA

Ítalo Santos Monteiro[1] - i_ s_ monteiro@id.uff.br
Joseana Veiga de Souza Frango[1] - joseanafrango18@idd.uff.br
Rosilene Abreu Portella Corrêa[1] - rosileneportella@id.uff.br
Cleber de Almeida Corrêa Junior[1] - cleberacj@id.uff.br
[1]Universidade Federal Fluminense, Instituto do Noroeste Fluminense de Educação Superior - Santo Antônio de Pádua, RJ, Brasil

Resumo. *Neste trabalho é apresentada uma comparação entre quatro métodos de otimização estocástica já consagrados na solução de problemas de minimização de funções objetivos não lineares, com o recém publicado método Alcateia. Os métodos utilizados para comparação são: Luus-Jaakola, Simuleated Aneeling, Particle Swarm Optimization, Differential Evolution. Esses métodos são utilizados na minimização de algumas funções não-lineares determinadas, como as funções Ackley, Sum Squares, Booth e Rosenbrock, onde realiza-se uma analise a cerca dos resultados gerados individualmente por cada método e em seguida comparam-se o desempenho dos mesmos, de forma que evidenciam a aplicabilidade e eficiência na resolução dos problemas propostos. Para isto, usou-se o software matemático Matlab, de modo que foram implementadas as referidas funções e posteriormente resolvidas computacionalmente pelos métodos estocásticos escolhidos. Os resultados obtidos são apresentados neste trabalho.*

Palavras-chave: *Metódos Estocásticos, Otimização, Funções Não-linear, Matlab*

1. INTRODUÇÃO

Os métodos de otimização estocástica são bastante utilizados na minimização numérica de funções e problemas diversos que abragem as áreas de ciências exatas e da terra. Tais métodos são frequentes na busca de melhores adequações e/ou processos que envolvem problemas de otimização. A resolução desses problemas, podem usar de métodos convencionais de otimização já existentes. Entretanto, os problemas de otimização globais são, na maioria das vezes, não-lineares, desta forma, ao utilizar de métodos heurísticos e suas variações possibilitam na resolução de problemas mais complexos.

Segundo Costa (2009), a partir da segunda metade do século XX técnicas de otimização com base em processos aleatórios e ponderadas por alguma heurística, conhecidas como técnicas estocásticas, ganharam destaques após seus desenvolvimentos obterem alguns êxitos nas soluções

de problemas que até então eram desconhecidos. Desde então, foram desenvolvidas diversas outras técnicas de otimização baseadas nos mesmos conceitos, porém com características distintas sem deixar as teorias heurísticas de lado, e puderam ser implementadas com maior eficiência no decorrer do avanço da capacidade de processamento dos computadores. Como consequência de tais estudos, destacam-se alguns métodos que tornaram-se consagrados e foram criados, fundamentados, dessas ideias, como: o *Luus-Jaakola* (LJ), que tem como objetivo selecionar soluções aleatórias em uma região de busca que decresce de tamanho com o decorrer das iterações; o *Simuleated Aneeling* (SA), em que se baseia no processo de recozimento, onde um material é fundido a alta temperatura e depois lentamente resfriado; o *Particle Swarm Optimization* (PSO), inspirado no comportamento biológico de enxames e aspectos de adaptação social; o *Differential Evolution* (DE), que é um método heurístico com fundamento em estratégias evolutivas populacional; e por fim um método pouco conhecido, por ser mais recente, o Alcatéia, inspirado no comportamento dos lobos (Luus & Jaakola (1973); Kennedy (1995); Corrêa (2013); Corrêa Jr (2015)).

Posto isto, o objetivo do presente trabalho é apresentar, por meio de testes computacionais com funções não-lineares selecionadas, os resultados que evidenciam a aplicabilidade e eficiência dos métodos supracitados na resolução de problemas de otimização. Para isso utilizou-se do software matemático Matlab.

2. MÉTODO LUUS-JAAKOLA

A idéia básica do algoritmo estocástico de *Luus-Jaakola* (Luus e Jaakola (1973)) consiste em selecionar soluções aleatórias em uma região que decresce de tamanho com o decorrer das iterações. Para iniciar o método, escolhe-se o tamanho de busca inicial $\mathbf{r}^0$; o número de laços internos (n_{int}) e externos (n_{ext}); o coeficiente de contração c e a solução inicial $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}_0$.

Para cada iteração j do laço interno, uma nova candidata a solução do problema será dada por,

$$\boldsymbol{\alpha}^j = \boldsymbol{\alpha}^* + \mathbf{R}^j \mathbf{r}^{(i-1)} \tag{1}$$

onde $\mathbf{R}$ é uma matriz diagonal formada por números aleatórios entre -0,2 e 0,2. Para cada nova solução candidata, o valor do funcional é avaliado e, caso um menor valor seja encontrado (em um problema de minimização) faz-se,

$$\boldsymbol{\alpha}^* = \boldsymbol{\alpha}^j. \tag{2}$$

Ao final de cada laço interno, enquanto o número de iteração i for menor que n_{ext}, o raio de busca deve ser contraído da seguinte forma,

$$\mathbf{r}^i = (1 - c)\mathbf{r}^{(i-1)}. \tag{3}$$

3. MÉTODO PARTICLE SWARM OPTIMIZATION

O PSO, desenvolvido por Kennedy e Eberhart (1995), é inspirado em um modelo de interação social. Cada indivíduo, ou partícula, na posição $\mathbf{p}$, é uma solução candidata, a qual se move

no espaço de busca com velocidade $\mathbf{v}$, dinamicamente ajustada de acordo com sua própria experiência de movimento e pela experiência de movimento do grupo.

Em cada geração t, as partículas Q são manipuladas de acordo com as seguintes relações,

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + r_1c_1(\mathbf{b}_i(t) - \mathbf{p}_i(t)) + r_2c_2(\mathbf{b}_g(t) - \mathbf{p}_i(t)); \tag{4}$$

$$\mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{v}_i(t+1), \tag{5}$$

onde w é denominado termo de inércia, c_1 e c_2 são coeficientes de aceleração; r_1 e r_2 são valores randômicos no intervalo [0,1]; $\mathbf{b}_i$ é a melhor posição ocupada pela partícula i, e $\mathbf{b}_g$ é a melhor posição encontrada por todo o grupo.

Altos valores para w favorecem a *exploration*, definida como a habilidade de explorar regiões do espaço de busca, enquanto pequenos valores de w promovem o que costuma-se chamar de *exploitation*, que é a habilidade de concentrar a busca ao redor de uma área promissora para refinar a solução candidata. Além disso, baixos valores para c_1 e c_2 promovem uma trajetória suave das partículas, enquanto que altos valores promovem maior aceleração e movimentos abruptos. Entretanto, valores pequenos para esses parâmetros de otimização podem render uma convergência lenta, enquanto valores altos podem render uma convergência prematura.

4. MÉTODO SIMULATED ANNEALING

O SA foi proposto por Kirkpatrik e Gellat (1983) e é baseado no processo de recozimento, onde um material é fundido a alta temperatura e depois lentamente resfriado. A teoria desse método consiste no fato de que, em temperaturas elevadas, o material possui uma configuração de alta energia, permitindo suas partículas se arranjarem aleatoriamente. A cada resfriamento de temperatura, o material muda para uma configuração de energia mais baixa. O objetivo é alcançar o nível de energia mínima, o que só acontece se a temperatura inicial T_0 for suficientemente alta e o processo de resfriamento for suficientemente lento. Se o resfriamento for rápido, o sólido não atingirá o nível de mínima energia e conterá inúmeras imperfeições.

A idéia básica do algoritmo de otimização SA é permitir, no início do processo (temperatura alta), que soluções piores que a atual sejam selecionadas, visando facilitar a obtenção do ótimo global. Quando a temperatura é baixa, apenas soluções que minimizem o funcional proposto são aceitas. Os movimentos que pioram o valor do funcional são permitidos de acordo com a distribuição de probabilidade,

$$Pb = e^{\frac{-\triangle\mathcal{E}}{kT}}, \tag{6}$$

onde, $\triangle\mathcal{E}$ é a variação da energia; k é uma contante de Boltzmann, cujo valor utilizado neste trabalho é 5×10^{-6}; e T é a temperatura de controle do recozimento.

O método é inicializado com a escolha da temperatura inicial e final, da solução inicial s e de um número de iterações internas para cada temperatura. Inicialmente, faz-se $\mathrm{s}^* = \mathrm{s}$. Enquanto a temperatura T for menor que a temperatura final T_f, para cada iteração interna, uma solução vizinha de s, s', é obtida. Assim, a variação de energia $\triangle\mathcal{E}$ é calculada como,

$$\triangle\mathcal{E} = \mathcal{F}(\mathbf{s}') - \mathcal{F}(\mathbf{s}). \tag{7}$$

Se $\triangle\mathcal{E} < 0$, faz-se $\mathbf{s} = \mathbf{s}'$. Se o valor da função objetivo para $\mathbf{s}'$ for menor que o valor atual ($\mathcal{F}(\mathbf{s}^*)$), a solução é imediatamente aceita, e assim $\mathbf{s}^* = \mathbf{s}'$. Caso contrário, o valor da probabilidade (Pb) será comparado com um número gerado aleatoriamente entre 0 e 1, obtido na iteração. Se o valor retornado Pb for maior que o número aleatório, o estado vizinho $\mathbf{s}'$, mesmo possuindo custo superior ao da solução atual, será selecionado, $\mathbf{s} = \mathbf{s}'$.

Ao final de cada laço interno a temperatura é reduzida. O resfriamento é executado, multiplicando-se a temperatura por uma variável α_r, tal que $0 < \alpha_r < 1$. Em geral, quanto mais α_r está próximo de 1, melhor é o resultado obtido e maior é o número de iterações necessárias.

5. MÉTODO DIFFERENTIAL EVOLUTION

O DE foi desenvolvido por Storn e Price (1995), e trata-se de um método de busca direta estocástica que surgiu da tentativa de resolver o problema de ajuste polinomial de Chebychev.

O método DE é inicializado com a escolha de uma população composta por $\mathbf{N}_p$ indivíduos espalhados pelo espaço de busca. Cada indivíduo é um vetor, cujo número de componentes é igual ao número de parâmetros do problema a ser resolvido. Essa população passará por três estágios para chegar à próxima geração. São eles: mutação, cruzamento e seleção.

Na mutação, a população é modificada adicionando-se, a um terceiro indivíduo, a diferença vetorial ponderada entre dois indivíduos aleatórios da população. Sendo $\mathbf{x}_\alpha$, $\mathbf{x}_\gamma$ e $\mathbf{x}_\delta$ vetores distintos entre si e escolhidos aleatoriamente de uma população composta por $\mathbf{N}_p$ indivíduos, o vetor $\mathbf{x}_\alpha$ será perturbado, resultando no vetor modificado ou doador ($\mathbf{v}_d$), dado na $(q+1)$-ésima iteração, através da seguinte expressão:

$$\mathbf{v}_d^{q+1} = \mathbf{x}_\alpha^q + \mathbf{F}_p(\mathbf{x}_\gamma^q - \mathbf{x}_\delta^q), \tag{8}$$

onde $\mathbf{F}_p$ é um número real positivo pertencente ao intervalo [0, 2] e denominado *fator de perturbação*.

O cruzamento é introduzido para aumentar a diversidade dos indivíduos que sofrem mutação. As componentes do indivíduo doador, $\mathbf{v}_d(i)^{q+1}$, são misturadas com as componentes $\mathbf{x}_a(i)^q$ de um indivíduo $\mathbf{x}_a^q$, escolhido aleatoriamente (vetor alvo), resultando num vetor tentativa ou experimental $\mathbf{v}_t^{q+1}$. As componentes do vetor experimental, $\mathbf{v}_t^{q+1}$, são escolhidas pela seguinte comparação:

$$\mathbf{v}_t(i)^{q+1} = \begin{cases} \mathbf{v}_d(i)^{q+1}, & \text{se} \quad \text{rand}_i \leq P_c. \\ \mathbf{x}_a(i)^q, & \text{se} \quad \text{rand}_i > P_c, \quad i = 1, \ldots, \mathbf{N}_p. \end{cases} \tag{9}$$

onde n_p refere-se ao número de componentes do vetor $\mathbf{x}_a$, rand_i é um número gerado aleatoriamente e P_c é a probabilidade de cruzamento, ambos no intervalo [0,1]. P_c é escolhido pelo usuario e representa a probabilidade de o vetor experimental herdar os valores das variáveis do vetor doador. Este tipo de cruzamento é denominado *cruzamento binomial*.

O operador seleção realiza um teste sobre a população gerada. Se o valor do funcional (ou função objetivo) do vetor experimental for menor que o valor do funcional para o vetor alvo, então o vetor experimental será escolhido para a próxima geração. Caso contrário, o vetor alvo é mantido na próxima geração.

6. MÉTODO ALCATEIA

Sejam dados um conjunto $\mathbf{D} \subset \mathbb{R}^n$ e uma função $f : \Omega \to \mathbb{R}$, onde $\Omega \subset \mathbf{D}$. O algorítmo Alcateia trata-se de uma metaheurística para o problema de achar um minimizador (ou maximizador) de f no conjunto $\mathbf{D}$. Podendo, tal problema, ser escrito como:

$$min\, f(\mathbf{x}) \quad sujeito\ a\ \mathbf{x} \in \mathbf{D}. \tag{10}$$

Ou seja, o problema consiste em encontrar uma solução $\bar{\mathbf{x}}$, com $\bar{\mathbf{x}} \in \mathbf{D}$, tal que $f(\bar{x}) \leq f(\mathbf{x})$, $\forall \mathbf{x} \in \mathbf{D}$. Para a resolução do problema supracitado, parte-se da característica comportamental do lobo alfa em basear-se em escolhas plausíveis para a captura da presa com o enfoque no sucesso da caça. Durante a execução do método Alcateia, os lobos estão suscetíveis a mudanças hierárquicas, ou seja, a cada momento será nomeado como lobo alfa aquele que escolher a melhor presa, ou seja, a melhor solução durante aquela iteração. Além disso, há o coeficiente de independência, que faz com que os outros lobos tendam a buscar abater a presa que o lobo alfa está indicando, refinando assim, a cada iteração, a solução encontrada até chegar à solução ótima para o problema proposto.

O número de laços internos é contraído por uma proporção conforme são executados os laços externos.

Simulando o comportamento dos lobos, podemos dizer que a cada laço interno, o lobo, está reposicionando-se no cerco da sua presa, e a cada laço externo o lobo está indo em direção à presa.

A seguir, apresenta-se, na Figura 1, o pseudo-código do algoritmo Alcateia, com passo-a-passo de seu funcionamento: Nota-se que não há no algoritmo um critério de parada, mas pode-se, de acordo com a aplicação, definir algum critério, não havendo necessidade de realização de todos os laços estipulados.

O algoritmo foi implementado no software Matlab.

```
(1)  Escolha o passo de busca inicial Δ // intervalo [a b]
(2)  Escolha o número de iterações externas Le
(3)  Escolha o número de iterações internas Li
(4)  Escolha o número de lobos Nlobos
(5)  Escolha o coeficiente de contração c
(6)  Escolha a constante de independência id
(7)  Gerar, aleatoriamente, uma solução inicial x0
(8)  para i de 1 até Le faça
(9)  aux ← x0 // variável auxiliar
(10)         para j de 1 até Li faça
(11)                 para k de 1 até Nlobos faça
(12)                         x[:,k] ← x0[:,k] + λk* Δ[:,k] // λ é um vetor de números randômicos entre -1 e 1. * é o produto termo-a-termo de vetores.
(13)                         se f(x[:,k]) < f(x0[:,k]) então
(14)                                 x0[:,k] ← x[:,k]
(15)                                 g[k] ← f(x[:,k])
(16)                         fim se
(17)                 fim para
(18)         fim para
(19) [fxalfa p] ← min(g) //função min(g) retorna a componente de menor valor de g, fxalfa, e sua posição p
(20) xalfa ← x0[:,p];
(21) Li ← Li x (1-c)
(22)         para t de 1 até Nlobos faça
(23)                 x0[:,t] ← (id x x0[:,t] + xalfa)/(id+1)
(24)                 Δ[:,t] ← |x0[:,t] - aux[:,t]|
(25)         fim para
(26) fim para
```

Figura 1- Pseudo-Código do método Alcateia

No algoritmo Alcateia, $\mathbf{x}$ é uma matriz, onde as colunas $x(:,k)$ representam os lobos. Em outras palavras, $x(:,k)$ são as variáveis que receberão as possíveis soluções ótimas (presas) para os problemas avaliados.

7. RESULTADOS E DISCUSSÕES

Os métodos estocásticos descritos neste trabalho foram implementados no software matemático Matlab, e acoplados as funções determinadas e conhecidas na literatura como, *Ackley*, *Sum Squares*, *Booth* e *Rosenbrock*, com o intuito de evidenciar a aplicabilidade e eficiência dos referidos métodos. Para a realização dos testes, algumas caracteristicas acerca da programação foram fixadas para todos os experimentos, desta forma, adotou-se a tolerância de 10^{-7} e a quantidade de duas variáveis, alterando apenas o intervalo de busca inicial $[a,b]$, como mostra a Tabela 1, o qual varia conforme o domínio de cada função, apresentadas a seguir nas Figuras 2-5.

No método LJ e Alcateia utiliza-se um laço interno e um laço externo. O critério de parada utilizado substitui o laço externo em ambos os métodos. Sendo assim, apenas ao final de todas iterações do laço interno, será verificado o critério de parada. Dessa forma, valores menores do que a tolerância adotada podem ser alcançados para as funções analisadas.

A Seguir é apresentada a função *Ackley*. O gráfico desta função é apresentado na Figura 2.

$$f(x) = -20e^{\left(-0.2\sqrt{\frac{1}{2}\sum_{i=1}^{2} x_i^2}\right)} - e^{\left(\frac{1}{2}\sum_{i=1}^{2} cos(2\pi x_i)\right)} + 20 + e \tag{11}$$

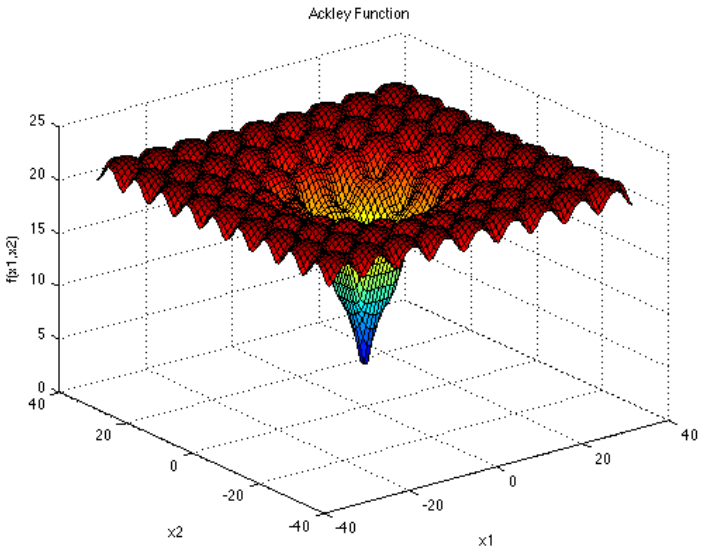

Figura 2- Gráfico da função *Ackley*. Fonte: Surjanovic & Bingham (2013).

Agora será apresentada a função *Sum Square*. O gráfico desta função é apresentado na Figura 3.

$$f(x) = \sum_{i=1}^{2} i x_i^2 \tag{12}$$

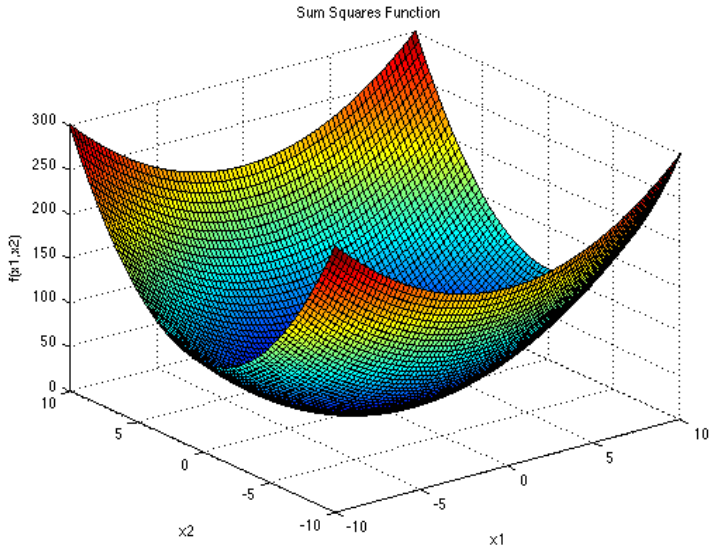

Figura 3- Gráficos da função *Sum Square*. Fonte: Surjanovic & Bingham (2013).

A função *Booth* é apresentada abaixo. Na Figura 4 apresenta-se o gráfico da função.

$$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \tag{13}$$

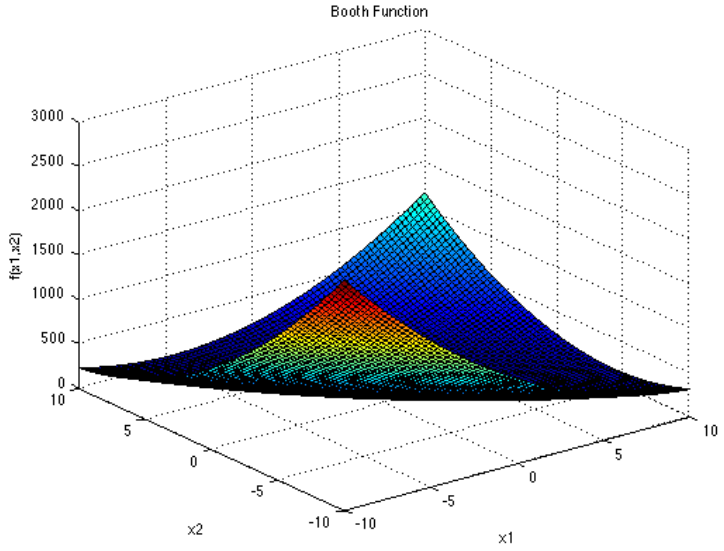

Figura 4- Gráfico da função *Booth*. Fonte: Surjanovic & Bingham (2013).

A função *Rosenbrock* é apresentada abaixo e seu gráfico na Figura 5

$$f(x) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2 \tag{14}$$

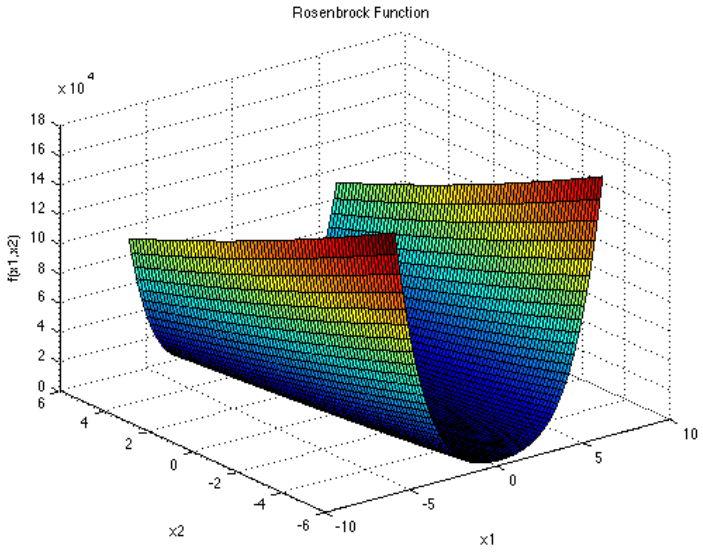

Figura 5- Gráficos da função *Rosenbrock*. Fonte: Surjanovic & Bingham (2013).

Tabela 1- Domínio de cada função

Ackley	$x_i \in [-32.768, 32.768]$	$i = 1, 2$
Sum Square	$x_i \in [-10, 10]$	$i = 1, 2$
Booth	$x_i \in [-10, 10]$	$i = 1, 2$
Rosenbrock	$x_i \in [-50, 50]$	$i = 1, 2$

Os resultados obtidos dessa implementação são explicitados nas Tabela 2 à 5. Desta forma, em cada tabela compara-se a solução analítica minimizadora com a sua aproximação encontrada pelos métodos LJ, PSO, SA, DE e Alcateia, para cada função apresentada anteriormente.

Tabela 2- Resultados Analíticos e Numéricos para a função *Ackley*

Métodos	Mínimo Analítico	Mínimo Numérico
LJ	$f(0,0) = 0$	$f(0.2141 \times 10^{-6}, 0.2521 \times 10^{-6}) = 4.5912 \times 10^{-8}$
PSO	$f(0,0) = 0$	$f(0.2609 \times 10^{-7}, 0.2006 \times 10^{-7}) = 9.3067 \times 10^{-8}$
SA	$f(0,0) = 0$	$f(0.9685, 0.9684) = 3.5745$
DE	$f(0,0) = 0$	$f(-0.1238 \times 10^{-6}, 0.2061 \times 10^{-6}) = 6.8868 \times 10^{-8}$
Alcatéia	$f(0,0) = 0$	$f(0.0680 \times 10^{-12}, 0.3530 \times 10^{-12}) = 1.0170 \times 10^{-12}$

Tabela 3- Resultados Analíticos e Numéricos para a função *Sum Squares*

Métodos	Mínimo Analítico	Mínimo Numérico
LJ	$f(0,0) = 0$	$f(0.0002, -0.0012) = 1.1088 \times 10^{-10}$
PSO	$f(0,0) = 0$	$f(0.0920 \times 10^{-3}, -0.1073 \times 10^{-3}) = 3.1467 \times 10^{-8}$
SA	$f(0,0) = 0$	$f(0.2023 \times 10^{-4}, -0.0003 \times 10^{-4}) = 4.0912 \times 10^{-10}$
DE	$f(0,0) = 0$	$f(0.0009, 0.0024) = 7.3319 \times 10^{-8}$
Alcatéia	$f(0,0) = 0$	$f(-0.0197 \times 10^{-11}, 0.2536 \times 10^{-11}) = 1.2897 \times 10^{-23}$

Tabela 4- Resultados Analíticos e Numéricos para a função *Booth*

Métodos	Mínimo Analítico	Mínimo Numérico
LJ	$f(1,3) = 0$	$f(1.0006, 2.9977) = 9.3941 \times 10^{-8}$
PSO	$f(1,3) = 0$	$f(1.0002, 2.9998) = 9.9195 \times 10^{-8}$
SA	$f(1,3) = 0$	$f(1,3) = 6.2212 \times 10^{-10}$
DE	$f(1,3) = 0$	$f(1.0020, 2.9961) = 5.7446 \times 10^{-8}$
Alcatéia	$f(1,3) = 0$	$f(1,3) = 1.2070 \times 10^{-26}$

Tabela 5- Resultados Analíticos e Numéricos para a função *Rosenbrock*

Métodos	Mínimo Analítico	Mínimo Numérico
LJ	$f(1,1) = 0$	$f(0.9999, 0.9982) = 2.2927 \times 10^{-8}$
PSO	$f(1,1) = 0$	$f(1,1) = 2.7653 \times 10^{-8}$
SA	$f(1,1) = 0$	$f(1,1) = 0$
DE	$f(1,1) = 0$	$f(0.9984, 0.9969) = 9.5563 \times 10^{-8}$
Alcatéia	$f(1,1) = 0$	$f(1,1) = 4.1020 \times 10^{-26}$

Vale ressaltar que todos os métodos foram inicializados randomicamente. Dos resultados expostos, observa-se que todos os métodos conseguiram gerar uma aproximação satisfatória do mínimo de cada função analisada. Em particular, pôde-se notar que o método Alcatea obteve os melhores resultados númericos para cada função escolhida. Por outro lado, métodos consagrados como o SA e LJ, também foram satisfatórios ao encontrar o mínimo para cada uma das funções estabelecidas.

8. CONCLUSÕES

O presente trabalho mostrou uma comparação entre o método Alcateia, recentemente apresentado por Corrêa Jr (2015), e outros métodos já consagrados na solução de problemas de otimização não linear. Como pode ser observado, o desempenho do Alcateia foi excelente para todas as funções objetivo analisadas. Os métodos de otimização estocástica são cada vez mais utilizados para resolver problemas não lineares de diversas áreas do conhecimento por ter vantagens como, por exemplo, a não utilização de derivadas e não ser necessário que a função seja contínua.

REFERÊNCIAS

Correa JR, C. A.; Corrêa, R. A. P.; Rangel, I. C. S. S.; Rangel, L. S.; Stutz, L. T. (2015), "Utilização da nova metaheurística Alcateia na identificação de danos estruturais", *XXXII Congresso de Matemática Aplicada e Computacinal da Região Sudeste*, Vitória, vol 3., N.2.

Corrêa, R. A. P. (2013), "*Identificação de danos em estruturas bi-dimensionais via matriz de flexibilidade baseada em um modelo de dano contínuo*", Tese de Doutorado, IPRJ/UERJ, Nova Friburgo, (2013).

Costa, J. S; Lima JR, C. A. S; Sacco, W. F. (2009), "Comparação entre os algoritmos Luus-Jaakola e Particle Swarm Optimization (PSO) em dois problemas de otimização global", *XXXII Congresso Nacional de Matemática Aplicada e Computacional*, Cuiabá, vol 2, 873-879.

Johnson, D.; Aragon, C.; Mcgeoch, L.; Schevon, C. (1991), Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Coloring and Number Partitioning.

Johnson, D.; Aragon, C.; Mcgeoch, L.; Schevon, C. (1991), Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning.

Kennedy, J. e Eberhat, T. H. (1995), Particle Swarm Opitimization, Proceedings of the 1995 IEEE Internacional Conference on Neural Networks, Perth, Australia, 1942-1948.

Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. (1983), Optimization by Simulated Annealing, Science, 220, 4598, 671-680.

Luus, R. e Jaakola, T. H. (1973), Optimization by Direct Search and Systematic Reduction of the Size of Search Region, AICDhE Journal, 19, 760-766.

Surjanovic, S. & Bingham, D. (2013), Virtual Library of Simulation Experiments: Test Functions and Datasets, Disponível em: <http://www.sfu.ca/∼ssurjano >. Acesso em: 13 jul. 2018.

COMPARATIVE ANALYSIS ON THE PERFORMANCE OF THE ALCATEIA OPTIMIZATION METHOD

Abstract. *In this paper we present a comparison of five existing and established stochastic optimization methods, more precisely the methods in the context of metaheuristic algorithms, Luus-Jaakola, Simulateated Aneeling, Particle Swarm Optimization, (Ackley, Sum Squares, Booth, and Rosenbrock), where an analysis of the theory and fundamentals of the results generated individually by each method and then their performance is compared, in a way that evidences the applicability and efficiency in the resolution of the proposed problems. For this, Matlab mathematical software was used, so that these functions were implemented and later solved computationally by the chosen stochastic methods. The results obtained were satisfactory in the search for minimums for each function.*

Keywords: *Stochastic Methods, Optimization, Nonlinear Functions, Matlab*