



## PROJETO DE PESQUISA – PIBIC

### *“Introdução à Otimização Evolutiva”*

**Aluno:** Huang Shiheng – RA 174718

**Orientador:** Prof. Dr. Romis Attux – DCA/FEEC/UNICAMP

#### 1 – Introdução

Diversos problemas práticos podem ser vistos como sendo tarefas de otimização de uma determinada métrica com respeito a um ou mais parâmetros. Problemas de otimização global requerem, via de regra, a utilização de métodos heurísticos. Dentre esses métodos, podemos destacar as meta-heurísticas bio-inspiradas conhecidas como algoritmos evolutivos (Talbi, 2009). Esses algoritmos utilizam operadores conceitualmente baseados, em alguma medida, no processo de seleção natural e na genética moderna para gerar novas soluções e, gradativamente, cobrir o espaço de busca do problema em questão.

O estudo desses algoritmos é uma interessante porta de entrada para a área de computação evolutiva e para o campo de computação natural num sentido mais amplo. Por isso, este projeto visou o estudo e implementação acerca de otimização evolutiva centrado nessas três técnicas para, num segundo momento, realizar os testes de benchmarks multimodais.

#### 2 – Algoritmos Evolutivos

De modo geral, os algoritmos evolutivos apresentam as seguintes características (Eiben; Smith, 2015):

- Uma população de indivíduos candidatos à solução do problema, cada um desses indivíduos corresponde a uma estrutura de dado que codifica um ponto no espaço de busca;
- Uma função objetivo (fitness) que avalia o desempenho das soluções candidatas. Melhores desempenhos ocasionam maiores probabilidades de sobrevivência ou de reprodução;
- A cada iteração do programa, a população atual gera uma população descendente de forma sexuada (com recombinação/crossover: troca de trechos de dados entre dois indivíduos pais) e/ou assexuada (com mutação: variação de alguns dados do indivíduo);

- Os indivíduos da próxima geração são então escolhidos entre a população atual e a população descendente.

O processo é repetido até atingir um critério de parada, como por exemplo o limite de iterações ou achar uma solução ótima para o problema. Alguns exemplos de algoritmos evolutivos são: os algoritmos genéticos (Mitchell, 1998), as estratégias evolutivas (Schwefel, 1995) e os métodos de evolução diferencial (Price; Storn; Lampinen, 2005). Essas variantes surgem devidos às diferentes formas de representação dos indivíduos, aos diferentes mecanismos de reprodução e de seleção.

### 3 – Materiais e Métodos

Os algoritmos foram implementados em Python e testados utilizando as funções de benchmark descritas na Tabela 1 (Liang et al., 2006; Oyman, 1999; Salomon, 1996; Rahnamayan; Tizhoosh; Salama, 2007). Tratam-se de problemas de minimização e serão testados para dimensões (n) iguais a 10 (número máximo de avaliações de fitness fixado para 20000) e iguais a 30 (número máximo fixado para 50000). Os algoritmos executaram os problemas 100 vezes e o melhor resultado obtido em cada execução será mostrado e analisado (média e desvio padrão). Os parâmetros usados por cada algoritmos foram iguais para todos as funções, portanto não necessariamente obteria um resultado satisfatório.

Tabela 1: Problemas de minimização usando os algoritmos implementados.

$f_1(x) = \sum_{i=1}^n x_i^2$
$f_2(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$
$f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$
$f_4(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$
$f_5(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^n x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$
$f_6(x) = 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i))$

$f_7(x) = x_1 + \sqrt{\sum_{i=2}^n x_i^2}$
$f_8(x) = 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$
$f_9(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4$

#### 4 – Resultados e Discussão

Os resultados de 100 execuções estão nas tabelas abaixo. Quanto menor o resultado (e mais próximo de 0) melhor, o melhor está em negrito.

Tabela 1: Médias e desvios padrões dos resultados com dimensão n=10.

f	AG	EE	ED
f <sub>1</sub>	4.4750e-13 ± 3.3457e-13	9.6389e-11 ± 2.4807e-10	<b>4.7120e-17 ± 3.8634e-17</b>
f <sub>2</sub>	3.7080e+01 ± 6.9949e+01	4.6299e+01 ± 3.8523e+01	<b>4.3372e+00 ± 7.7768e-01</b>
f <sub>3</sub>	4.6861e-06 ± 1.5815e-06	6.2352e-05 ± 9.2754e-05	<b>6.5211e-08 ± 2.5740e-08</b>
f <sub>4</sub>	<b>4.8720e-02 ± 3.3686e-02</b>	1.2074e+00 ± 7.0224e-01	2.6012e-01 ± 7.7272e-02
f <sub>5</sub>	2.4187e-01 ± 6.8088e-02	7.5976e-01 ± 3.7176e-01	<b>1.0136e-01 ± 9.9794e-03</b>
f <sub>6</sub>	5.1041e+00 ± 2.5586e+00	<b>1.8108e+00 ± 1.2217e+00</b>	1.7901e+01 ± 3.1237e+00
f <sub>7</sub>	2.9717e-01 ± 1.4944e-01	1.4309e-04 ± 8.3355e-04	<b>1.6399e-09 ± 8.9414e-10</b>
f <sub>8</sub>	1.1054e+03 ± 2.6510e+02	<b>9.9848e+01 ± 7.5185e+01</b>	3.5577e+02 ± 2.9316e+02
f <sub>9</sub>	3.0418e-01 ± 1.4063e+00	8.7620e-01 ± 1.5723e+00	<b>1.0480e-15 ± 9.0544e-16</b>

Tabela 2: Médias e desvios padrões dos resultados com dimensão n=30.

f	AG	EE	ED
f <sub>1</sub>	5.9751e-12 ± 1.6621e-12	<b>1.7081e-12 ± 5.3576e-13</b>	1.7397e-11 ± 1.0755e-11
f <sub>2</sub>	6.5913e+01 ± 6.6987e+01	6.1846e+01 ± 3.0035e+01	<b>2.4325e+01 ± 6.8766e-01</b>
f <sub>3</sub>	1.0815e-05 ± 1.6123e-06	1.2627e-04 ± 1.2419e-03	<b>2.7126e-05 ± 1.0827e-05</b>
f <sub>4</sub>	6.2811e-03 ± 8.3947e-03	2.3636e+00 ± 7.6881e-01	<b>1.6778e-04 ± 1.0665e-03</b>
f <sub>5</sub>	1.2739e+00 ± 3.3664e-01	1.5907e+00 ± 5.6149e-01	<b>3.2865e-01 ± 4.3163e-02</b>
f <sub>6</sub>	2.3272e+01 ± 5.8405e+00	<b>4.9449e+00 ± 2.1864e+00</b>	1.6744e+02 ± 1.2839e+01
f <sub>7</sub>	2.8141e-01 ± 1.1757e-01	<b>2.1438e-06 ± 5.9357e-06</b>	2.7798e-06 ± 8.1631e-07
f <sub>8</sub>	5.4907e+03 ± 5.2532e+02	<b>3.2681e+02 ± 1.9672e+02</b>	6.3402e+03 ± 3.7688e+02
f <sub>9</sub>	8.2709e+01 ± 1.5625e+02	<b>7.3684e-09 ± 6.9474e-08</b>	7.3774e-09 ± 3.7427e-09

Para testes realizados com dimensão 10, em geral, a ED apresentou um desempenho médio melhor que os outros dois algoritmos, enquanto para os testes com dimensão 30, a EE apresentou um melhor desempenho médio.

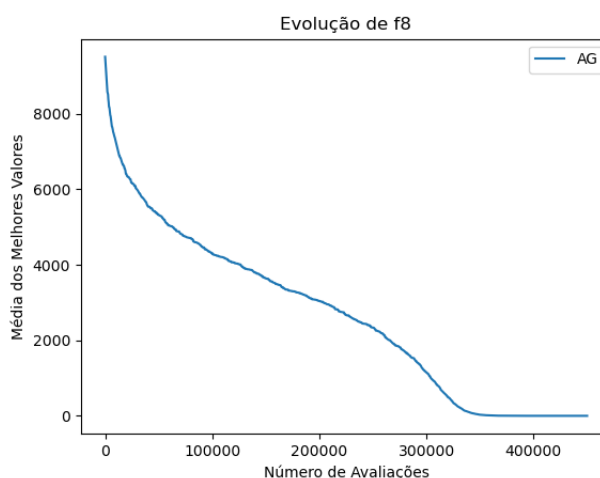
Os operadores implementados para AG ocasionam uma seleção extremamente elitista entre os pais e os filhos para compor a próxima geração, fazendo com que ele consiga atingir rapidamente algum mínimo local.

A EE implementada utiliza um procedimento de crossover que é capaz de criar uma grande diversidade genética entre os filhos, aumentando assim o poder de explorar o espaço de busca. Os pais não participam da próxima geração, o algoritmo consegue escapar de mínimos locais, diferente do algoritmo de AG implementado, que fica preso neles. Isso pode ser visto comparando os resultados de  $f_6$  e de  $f_8$ , esses são problemas com diversos mínimos locais. O AG é o que converge mais rapidamente para algum mínimo local, enquanto a EE converge mais lentamente, porém apresenta melhores resultados.

A ED é um algoritmo de fácil implementação e não apresenta muitos parâmetros a serem definidos pelo usuário. Apresenta uma seleção elitista entre uma solução e um filho criado associado, e ao mesmo tempo mantém a diversidade nos indivíduos da população. A medida em que ocorre a convergência num mínimo local, o raio de busca também fica cada vez menor.

Como para nenhum dos algoritmos as otimizações da  $f_8$  de dimensão 10 e 30 foram satisfatórias, foi executado novamente com o algoritmo da ED ajustando os novos parâmetros para  $n = 30$ : foi aumentado o número de iterações, o tamanho da população e diminuído a probabilidade do crossover para escapar dos mínimos locais. Executando 30 vezes, o resultado obtido foi  $5.4262e-04 \pm 1.8913e-04$ , com 450000 avaliações de fitness por execução. A evolução média das execuções está na figura 1.

Figura 1: Evolução de  $f_8$  com 30 execuções.



O número de avaliações seria muito maior se tentasse calcular o valor da função nos pontos igualmente espaçados dentro do intervalo de busca, para

num segundo momento aplicar métodos de otimização clássica como método de Newton-Raphson nesses pontos. No caso da  $f_g$  de dimensão 30, cada variável pode assumir qualquer valor entre -500 e 500. Se analisasse todos os pontos de uma variável espaçados de 1, cada variável terá 1001 pontos a serem analisados, gerando assim uma combinação total de  $1001^{30}$  pontos iniciais a serem avaliados. Este é um exemplo de um problema complexo em que o método clássico apresenta um custo computacional muito maior que os algoritmos evolutivos, e as vezes chega a ser inviável de implementação.

## 5 – Conclusão

Os algoritmos evolutivos são ferramentas poderosas e úteis para a otimização dos problemas complexas, uma vez que consegue fornecer soluções ótimas muito próximas da melhor solução, utilizando um número muito menor de avaliações e em menor tempo comparado com os algoritmos clássicos.

## 7 – Referências

- 1 - Eiben, A. E.; Smith, J. E.. **Introduction to Evolutionary Computing**. 2. Ed. Berlin: Springer, 2015.
- 2 - Liang et al.. Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. **IEEE Transactions on Evolutionary Computation**, v. 10, n. 3, jun. 2006.
- 3 - Mitchell, M.. **An Introduction to Genetic Algorithms**. Cambridge: MIT Press, 1998.
- 4 - Oyman, A. I.. **Convergence Behavior of Evolution Strategies on Ridge Functions**. University of Dortmund, 1999.
- 5 - Price, K. V.; Storn, R. M.; Lampinen, J. A.. **Differential Evolution: A Practical Approach to Global Optimization**. Berlin: Springer, 2005.
- 6 - Rahnamayan, S.; Tizhoosh, H.; Salama, M. M. A.. A Novel Population Initialization Method for Accelerating Evolutionary Algorithms. **Computers and Mathematics with Applications**, v. 53, n. 10, p. 1605-1614, 2007.
- 7 - Salomon, R.. Reevaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions: A Survey of Some Theoretical and Practical Aspects of Genetic Algorithms. **BioSystems**, v. 39, n. 3, p. 263-278, 1996.
- 8 - Schwefel, H. P.. **Evolution and Optimum Seeking**. New York: Wiley, 1995.
- 9 - Talbi, E. G.. **Metaheuristics: From Design to Implementation**. Hoboken: Wiley, 2009.