# Schaffer Evolucionário: Um problema de optimização.

Lia Sucupira Furtado, Gabriel Alves das Neves, Gabriel da Rocha Silva e André Luis Dantas Gadelha

Universidade Federal do Ceará, Dept. de Teleinformática Campus do Pici, Av. Mister Hull, Brasil

**Resumo**. A computação evolucionária é uma área de pesquisa dentro da ciência da computação. E como o nome sugere se inspira no processo de evolução natural. Os algoritmos genéticos, também conhecidos como evolucionários, fazem parte dessa computação, e assim são inspirados no processo biológico de seleção natural. Para Charles Darwin, a seleção natural consistia : "Quanto melhor um indivíduo se adaptar ao seu ambiente, maior a probabilidade de sua sobrevivência e geração de descendentes".

No presente trabalho estudaremos a otimização da função Schaffer N.2 comparando a utilização de algorítimos genéticos e do método do Gradiente. As análises que foram realizadas na função optimizada pelo algorítimo genético demonstram que o algorítimo, apesar de seus custos, chega a resultados mais próximos da realidade se comparado com o método do Gradiente.

## 1 Introdução

Devido ao caráter evolucionário, os algorítimos genéticos são usados principalmente para resolução de problemas de optimização.

Outra classe de métodos usado para problemas de optimização é conhecida como heurística, por exemplo o algoritmo 'simulated annealing'. Infelizmente, de acordo com A.E. Eiben and J.E. Smith [1] estes algoritmos são bastante fracos, no sentido de que eles não irão identificar a otimização global. O termo otimização global refere-se ao processo de tentativa de encontrar a solução com o valor ideal para alguma função de fitness.

Tem também os algoritmos de buscas locais, freqüentemente chamados de 'hill climbers' que tem a vantagem de serem rápidos em identificar uma boa solução para o problema. No entanto, a desvantagem é que os problemas frequentemente exibem numerosos ótimos locais,que é uma solução que é ótima para um conjunto de soluções vizinhas mas que podem ser significativamente piores que o ótimo global.

Existem vários recursos dos Algorítimos Genéticos que os diferenciam desses algoritmos mencionados acima,relacionados principalmente ao uso de uma população. A capacidade dos EAs de manter um conjunto diversificado de população, proporcionado pelos procedimentos de cruzamento e mutação fornece um meio de fugir de ótimos locais.

Assim, é preferido o uso desses algoritmos devido a essas vantagens mencionadas.

## 2 Metodologia

A função Schaffer N.2 faz parte de uma família de funções que avaliam o desempenho e ilustram muito bem a eficácia e a aplicação de algorítimos de otimização. a função é descrita como:

$$f(x,y) = 0.5 + \frac{sin^2(x^2-y^2)-0.5}{(1+0.001(x^2+y^2))^2}$$

A dificuldade da optimização vem da quantidade e da proximidade de máximos e mínimos locais. Na literatura temos que o mínimo global da função é 0,onde os valores de x1 e x2 são ambos 0[3].

Dessa forma, serão utilizados dois métodos para encontrar o mínimo global. O primeiro método é um algorítimo genético, já o segundo é o Operador Gradiente. Para a implementação de ambos foi utilizada a linguagem R.

Um exemplo de pseudo-código que demonstra os passos principais de um algorítimo genético é ilustrado na figura 1.

**Algorithm 1** Generic Evolutionary Algorithm

Let $t = 0$ be the generation counter;
Create/initialize the $n_x$-dimensional population, $C(0)$, to consist of $n_s$ individuals

1: **while** *stopping condition*$(s)$ *not true* **do**
2: Evaluate the fitness, $f(x_i(t))$, of each individual, $x_i(t)$ (select the parents);
3: Perform reproduction to create offsprings (reproduction and mutation);
4: Select a new population, $C(t+1)$ (survivor selection);
5: Advance to the new generation, i.e., $t = t+1$;
6: **end while**

Figura 1: Pseudocódigo algoritmos genéticos

Os parâmetros taxa de mutação e a taxa de cross-over exercem grande influência nos resultados tendo em vista que eles determinam os descendentes a partir dos quais as populações seguintes evoluirão. Já o Elitismo, faz com que os indivíduos que estão melhor adaptados sejam selecionados e possam transmitir os melhores cromossomos para as gerações futuras. Dessa forma cada nova população será mais evoluída que a de seu pais.

### 2.1 Implementação

O método Gradiente faz diversas derivadas para encontrar o mínimo, que, indesejávelmente, pode ser um mínimo local e não global. Para implementá-la em R, a linguagem nos permite utilisar a função optim() que, por padrão, aplica o método do Gradiente.

A linguagem R dispõe da biblioteca GA que nos permite implementar um algorítimo genético facilmente. Nela a população é gerada aleatoriamente, o numero de iterações pode ser definido a gosto do usuário, bem como a taxa de mutação e de cross-over.

## 3 Resultados

Após implementar a função Schaffer N.2, foi definido o domínio da função no intervalo de -100 até 100 para ambas as variáveis (x1 e x2). em seguida plotamos o gráfico da função Na figura 2, podemos visualizar a abundância e concentração de máximos e mínimos locais.

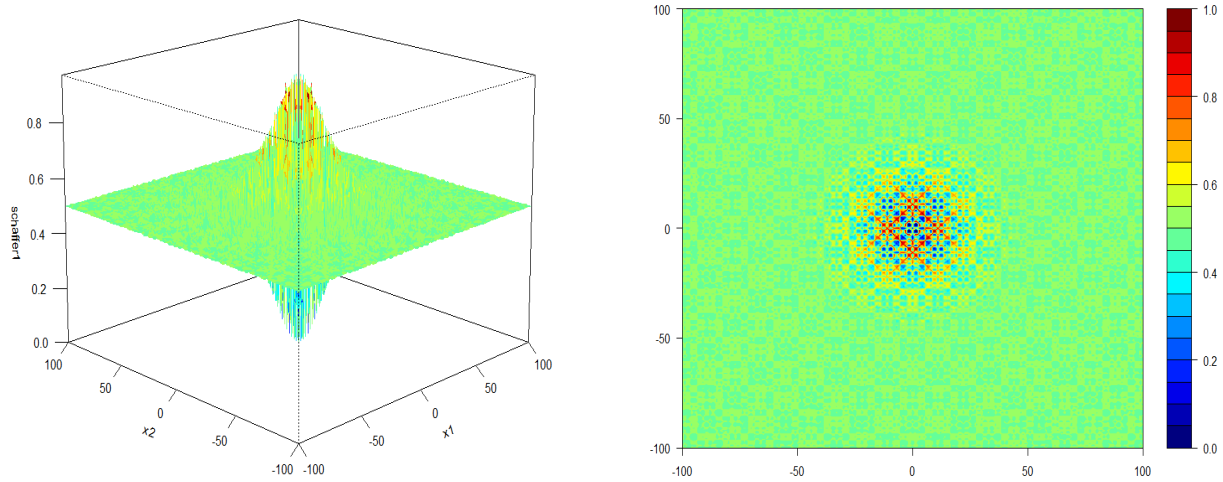


Figura 2: Plot da função Schaffer N.2 (esquerda) e vista de cima da função (direita)

Como parâmetros do algoritmo, aplicamos uma taxa de cross-over de 0,7, uma taxa de mutação de 0,1, uma taxa de elitismo de 5 e um numero máximo de iterações de 100. Esses valores foram escolhidos para garantir que há uma taxa de cross-over alta e que o número de mutações não atrapalhem na convergência do algoritmo.

Quando executamos o código, observamos que os valores dos métodos foram distintos, onde os valores do algorítimo genético foram mais próximas da realidade que os valores do Gradiente (tabela 1).

Os algoritmos genéticos se destacam em produzir resultados mais precisos que o método do Gradiente, apesar dos problemas apresentados nesse método. A melhor performance se deve à capacidade de escapar de ótimos locais, que estão presentes em alguns modelos de otimização de processos com múltiplas respostas. As desvantagens ficam por conta de um maior tempo computacional gasto, e na dificuldade de implementação quando comparado ao Gradiente.

| | GA | Gradiente |
|---|---|---|
| f(x1,x2) | -9.474421e-12 | 2.698604e-08 |
| x1 | 5.65052e-05 | 0.004203868 |
| x2 | 7.925928e-05 | 0.003040143 |

Tabela 1: Comparação entre os métodos utilizados

Traçamos o Gráfico do Algorítimo Genético (Fig. 3.) para comparar o valor da função fitness para cada geração. Neste caso, o Algorítimo se comporta melhor que o gradiente, devido ao seu caráter aleatório na geração dos descendente a serem avaliados, enquanto o gradiente segue até encontrar um vale, seja ele um mínimo global ou não.

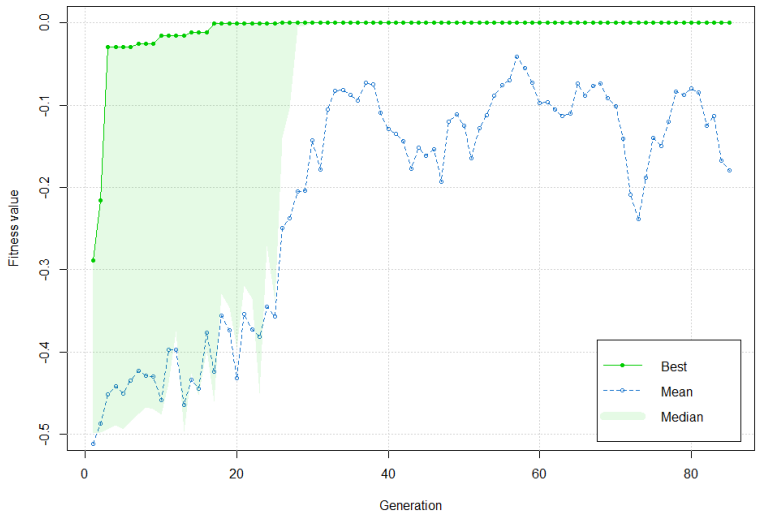


Figura 3: Plot da função fitness

## Referências

[1] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*, Springer, 2007

[2] G. James, H. Trevor, W. Daniela, T. Robert. *An Introduction to Statistical Learning with Applications in R*, 2013