

Buscando Valor Máximo de Uma Função Utilizando Algoritmos Genéticos

Diego Rubin¹

¹Departamento de Matemática Aplicada e Computação
- Universidade Estadual Paulista "Julio de Mesquita Filho" (UNESP)
Rio Claro – SP – Brazil

rubin.diego@gmail.com

Abstract. *This article aims to show the results for a maximization problem of a function using a technique programming called Genetic Algorithms.*

Resumo. *Este artigo tem como objetivo mostrar os resultados obtidos para um problema de maximização de uma função, utilizando uma técnica de programação chamada de Algoritmos Genéticos.*

1. Introdução

O principal propósito de um programa escrito utilizando Algoritmo Genético é encontrar um valor, máximo ou mínimo, para uma função, que damos o nome de função objetivo. Será mostrado neste artigo um problema de maximização e como a técnica do Algoritmo Genético foi empregado para encontrar os valores das variáveis que resultam a um bom valor para a função objetivo.

Para resolução do problema foi utilizado a linguagem de programação Ruby. O código fonte pode ser encontrado em <http://github.com/diegorubin/Evolutionary-Computation/tree/master/problem1/ruby>

2. O Problema

O Problema que iremos abordar é o seguinte:

A Capitão Caverna S.A., localizada em Pedra Lascada, aluga 3 tipos de barcos para passeios marítimos: jangadas, supercanoas e arcas com cabine. A companhia fornece juntamente com o barco um capitão para navegá-lo e uma tripulação que varia de acordo com a embarcação: uma para jangadas, duas para supercanoas e três para arcas. A companhia tem 4 jangadas, 8 supercanoas e 3 arcas e em seu corpo de funcionários: 10 capitães e 18 tripulantes. O aluguel é por diárias e a Capitão Caverna lucra R\$ 50 por jangada, R\$ 70 por supercanoa e R\$ 100 por arca. Faça um modelo de algoritmo genético que determine o esquema de aluguel que maximiza o lucro.

3. Função Objetivo

O principal problema quando modelamos um algoritmo genético é encontrar a função que descreverá o quão um individuo é bom em uma determinada população. Esta será a função que avaliará cada individuo em busca do melhor.

A função que será utilizada neste problema é $f(x, y, z) = 50x + 70y + 100z$, onde x é o número de jangadas utilizadas, y o número de supercanoa e z o número de arcas.

4. Restrições

Não podemos adotar qualquer valor no plano dos números reais. Qualquer problema que queiramos adotar um Algoritmo Genético deve conter algumas restrições de valores possíveis para a função objetivo.

Para este problema adotaremos as seguintes restrições: $x \leq 4$; $y \leq 8$; $z \leq 3$: Por causa dos números disponíveis de cada embarcação; $x + y + z \leq 10$: Existem apenas 10 capitães e é necessário um para cada embarcação; $x + 2y + 3z \leq 18$: Existem 18 tripulantes disponíveis, sendo que um é necessário para a jangada, dois para a supercanoa e três para arcas.

5. Representação do Cromossomo

Os valores das variáveis estão relacionados com a quantidade de embarcações utilizadas, portanto os valores que serão utilizados são inteiros. Será utilizado apenas um cromossomo composto por três genes.

Uma representação binária será adotada neste experimento, sendo os três primeiros bits representando o valor x , os próximos quatro bits para representar o valor y e os últimos dois bits para representar o valor de z , totalizando nove bits.

6. Operações

Serão adotados dois tipos de operações entre os indivíduos da população, a operação de Crossover, onde dois indivíduos trocam cromossomos gerando um filho e a operação de Mutação, onde um indivíduo gera um filho.

O Crossover utilizado será do tipo uniforme, onde uma máscara é criada aleatoriamente, funcionando da seguinte forma: se a máscara em uma determinada posição conter o valor '1', o filho receberá o valor do pai 1, senão o filho receberá o valor do pai 2.

A Mutação funcionará da seguinte forma, uma posição do cromossomo é escolhida aleatoriamente e seu valor é invertido.

Estas operações podem gerar filhos impossíveis, ou seja, que não obedecem as restrições impostas. Quando isso acontece os genes são arrumados, uma posição aleatória do gene tem seu valor trocado até ter um valor possível.

A seleção dos indivíduos para o cruzamento, ou para mutação, é feita através do método da roleta viciada. São somados todas as aptidões da população, após isso uma série de intervalos é criado de acordo com a porcentagem da aptidão deste indivíduo entre a somatória das aptidões, sendo esta a chance do mesmo ser escolhido para procriar.

7. Resultados

Após varias execuções do programa foi possível perceber que os indivíduos sempre caminham para um mesmo valor. Nas primeiras execuções foram utilizados 1000 indivíduos em 1000 gerações geradas, porém foi possível concluir que por volta da 148ª geração as populações já se mantinham constantes. A 148ª população pode ser vista na Figure 2.

O tempo que a aplicação demorou para rodar um caso de teste com 1000 gerações de 1000 indivíduos foi de aproximadamente 5 horas 40 minutos.

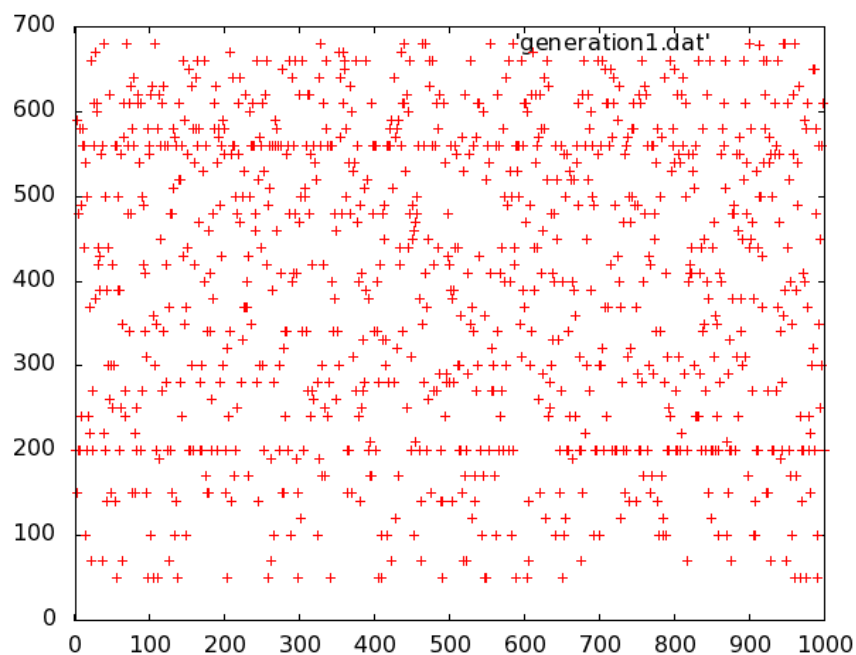


Figure 1. Gráfico da 1ª geração.

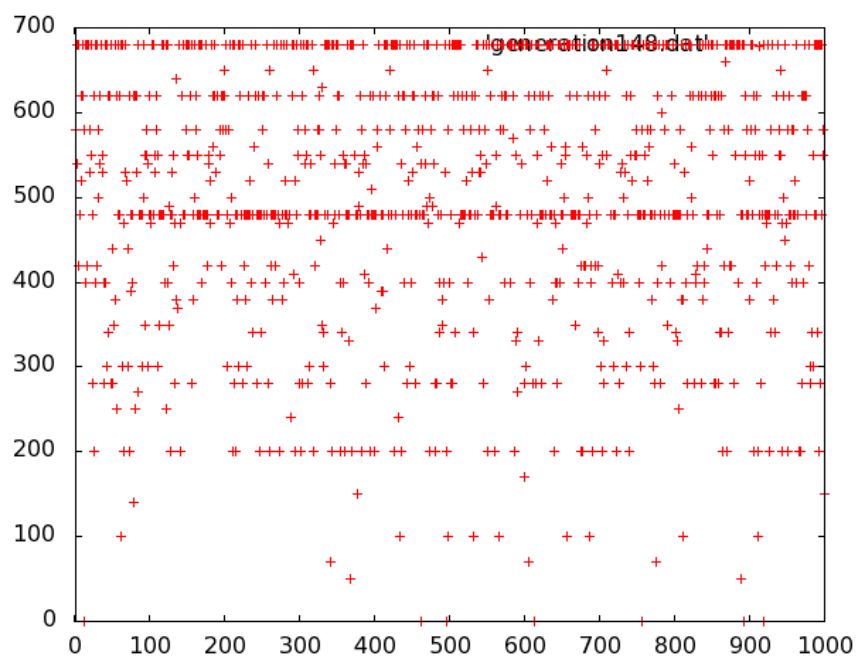


Figure 2. Gráfico da 148ª geração.

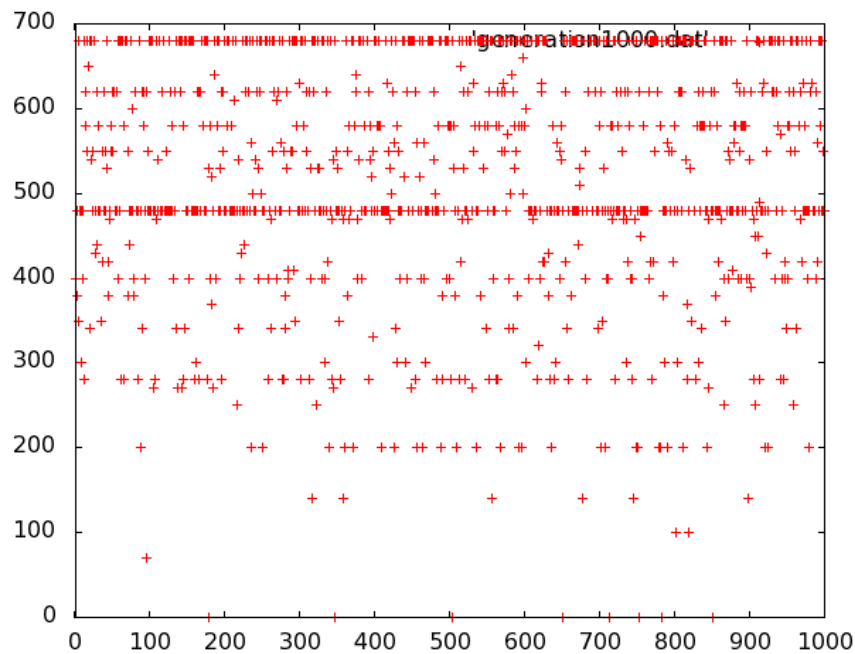


Figure 3. Gráfico da 1000ª geração.

Após executar o programa, o resultado obtido foram os valores, $x = 4, y = 4, z = 2, f(x, y, z) = 680$. Estes são valores muito bons para a solução do problema proposto.

8. Conclusão

Conseguimos alcançar um ótimo valor para o problema, maximizando a função, porém, como o problema é de certa forma simples, a abordagem utilizando Algoritmo Genético talvez não seja a melhor. Foi possível visualizar muitos indivíduos que maximizam a função já na primeira geração. Em um problema com maior complexidade, isto poderia não ocorrer e o algoritmo conseguiria chegar no valor correto. Foi possível notar que escrever uma boa função objetivo e a representação correta dos cromossomos são a chave para a resposta, mas cada detalhe faz a diferença. Como por exemplo, a escolha de como os indivíduos impossíveis são ajustados, neste trabalho, da forma em que se encontra na final fez toda diferença.