

Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Algoritmos Genéticos

Aluno: Fabricio Aparecido Breve

Prof.: Dr. André Ponce de Leon F. de Carvalho

São Carlos – São Paulo

Maio / 2007

Resumo

Este trabalho mostra duas aplicações distintas de algoritmos genéticos, a primeira para encontrar o valor mínimo para um polinômio de quarto grau e a segunda para resolver o tradicional problema do caixeiro viajante (TSP).

Introdução

Um algoritmo genético (AG) é uma técnica de procura utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e busca. Os algoritmos genéticos compõem uma classe particular de algoritmos evolutivos que usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação.

Algoritmos genéticos são implementados como uma simulação de computador em que uma população de representações abstratas de possíveis soluções é selecionada e combinada em busca de soluções melhores. A evolução geralmente se inicia a partir de um conjunto de soluções criado aleatoriamente e é realizada através de gerações. A cada geração, a adaptação de cada solução na população é avaliada, alguns indivíduos são selecionados para a próxima geração, e recombinados ou mutados para formar uma nova população. A nova população então é utilizada como entrada para a próxima iteração do algoritmo.

Na primeira etapa deste trabalho um algoritmo genético é utilizado para minimizar um polinômio de quarto grau. Um conjunto de possíveis soluções é criado e, após sucessivas iterações, espera-se encontrar o valor mínimo. Na segunda etapa utilizamos um algoritmo genético para encontrar uma solução para o problema do caixeiro viajante (TSP).

Minimização do Polinômio de Quarto Grau

Especificação do problema

O problema consiste em encontrar o valor mínimo para o seguinte polinômio:

$$f(x) = 2x^4 - 3x^3 + 7x - 5$$

Deve se assumir $x \in [-31, +32]$ e codificar X como um vetor binário, usando 6 bits, sendo que o primeiro bit deve ser usado para representar o sinal (1 para positivo e 0 para negativo).

A população inicial deve ser de 30 indivíduos. Deve-se utilizar mutação em apenas um gene da população e crossover de um ponto. O número máximo de gerações é 1000.

Codificação do vetor binário

O vetor binário é dado por $x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]$, onde x_1 representa o sinal. Para transformar o vetor binário em um número decimal atendendo as especificações utiliza-se o seguinte algoritmo:

$$x_{dec} = 16x_2 + 8x_3 + 4x_4 + 2x_5 + x_6$$

Se $x_1 = 0$ então $x_{dec} = -x_{dec}$

$$\text{Senão } x_{dec} = x_{dec} + 1$$

Experimento e Análise

Para realização do experimento foi determinada aleatoriamente uma população de 30 indivíduos, cujos vetores binários correspondem a decimais no intervalo $[-31, +32]$ conforme a especificação.

A escala do índice de aptidão foi feita através de *ranking*, onde os indivíduos são ordenados pelo valor de sua função de aptidão antes da escala, e sua pontuação será relativa ao seu posicionamento nesse ranking.

A seleção foi feita pela amostragem estocástica universal, onde uma linha é traçada e cada possível pai corresponde a uma seção dessa linha proporcional a sua probabilidade. O algoritmo se move ao longo da linha em passos uniformes, um passo para cada pai a ser escolhido. A cada passo, o algoritmo escolhe um pai, que corresponde à seção onde ele parou. O primeiro passo é um valor aleatório de tamanho menor que o tamanho dos passos seguintes.

Para cada nova geração os dois indivíduos de maior desempenho da geração anterior foram automaticamente selecionados (elitismo). Do restante, 22 foram obtidos através de cross-over de um ponto e os outros seis foram obtidos através de mutação de um único gene.

Após 50 gerações o valor mínimo encontrado para o polinômio foi -7, obtido com $x = [000001]$, que equivale ao decimal -1, segundo o nosso algoritmo.

A Figura 1 mostra a variação da maior aptidão e da aptidão média dos elementos a cada geração. Podemos observar que a aptidão média começa com um valor bastante elevado e vai diminuindo logo nas primeiras gerações, mostrando a predominância de elementos da população mais aptos ao longo do tempo, ou seja, elementos que correspondem a valores de x que encontram valores menores para o polinômio. A média permanece variável ao longo das gerações, mostrando que a busca por uma maior aptidão continua, e a melhor aptidão permanece constante devido ao elitismo.

Também pode ser observado na Figura 1 que o valor mínimo já é obtido desde a primeira geração, o que provavelmente se deve ao grande número de elementos da população (30) em relação ao universo de possíveis respostas (64), o que implica em uma probabilidade de 47% de que um dos elementos da população inicial já seja o de maior aptidão possível, considerando que a escolha dos elementos iniciais é aleatória e de que apenas uma das respostas possíveis é a de melhor aptidão.

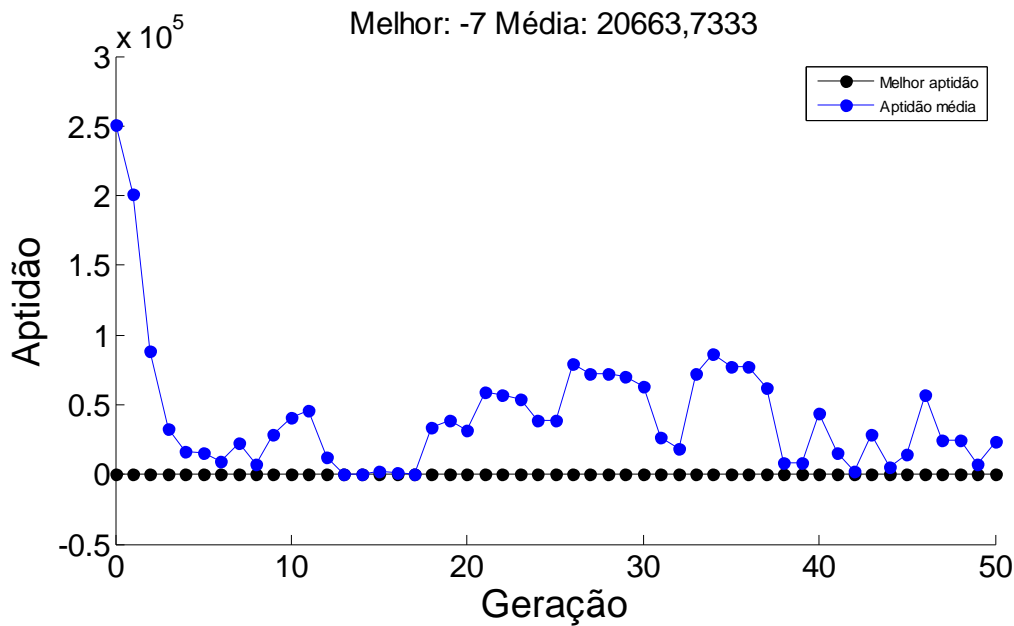


Figura 1 – Variação da maior aptidão e aptidão média dos elementos da população em cada geração para o problema do polinômio

Caixeiro Viajante

Especificação do Problema

Um caixeiro viajante deve visitar N cidades em sua área de vendas. Ele começa de uma base, visita cada cidade uma única vez e retorna à sua cidade no final. A cada viagem está associado um custo proporcional à distância percorrida. O caixeiro deve percorrer a rota mais curta.

Modificações nas funções de crossover e mutação

O algoritmo genético tradicional precisa de algumas modificações para resolver corretamente com o problema do caixeiro viajante. Isto é necessário, pois se codificarmos uma string de cidades (ou caminhos), considerando n como o número de cidades teríamos n^n soluções possíveis, mas apenas $n!$ soluções válidas, pois o problema determina que cada cidade só poderá ser visitada uma única vez e que todas as cidades devem ser visitadas. Pode-se optar por apenas penalizar soluções inválidas com um valor de aptidão bastante alto, mas essa alternativa não resolve completamente o problema, pois a grande maioria dos filhos gerados a cada geração serão inválidos e dificilmente será encontrada uma boa solução.

Para contornar esse problema algumas soluções podem ser adotadas, uma delas consiste em modificar as funções que geram a população inicial e as que fazem o crossover e a mutação. Podemos substituir o crossover tradicional pela troca e/ou inversão de fragmentos de caminhos e a mutação pela troca um para um da posição de cidades do itinerário. Essa foi a solução adotada neste trabalho e que apresentou bons resultados.

Experimento e Análise

Na tabela 1 temos uma lista de oito cidades e a respectiva distância entre elas. O objeto é ordenar essas cidades (na ordem em que serão visitadas) de forma que a distância total percorrida seja a menor possível.

Tabela 1. Distância entre cidades

	A	B	C	D	E	F	G	H
A	0	42	61	30	17	82	31	11
B	42	0	14	87	28	70	19	33
C	61	14	0	20	81	21	8	29
D	30	87	20	0	34	33	91	10
E	17	28	81	34	0	41	34	82
F	82	70	21	33	41	0	19	32
G	31	19	8	91	34	19	0	59
H	11	33	29	10	82	32	59	0

Para realização do experimento foi determinada aleatoriamente uma população de 20 indivíduos, cujos vetores correspondem a uma lista de cidades na ordem em que serão visitadas.

Da mesma forma que foi feito o experimento do polinômio, foram utilizadas: a escala do índice de aptidão através de *ranking*, a seleção por amostragem estocástica universal, e dois indivíduos selecionados por elitismo. Do restante, quatro foram obtidos por mutação e 14 por crossover.

Após 200 gerações o menor caminho encontrado tem um custo de 140. A Figura 2 mostra a variação da maior aptidão e da aptidão média dos elementos a cada geração. Podemos observar que após cerca de 20 gerações já é encontrado o menor caminho do experimento.

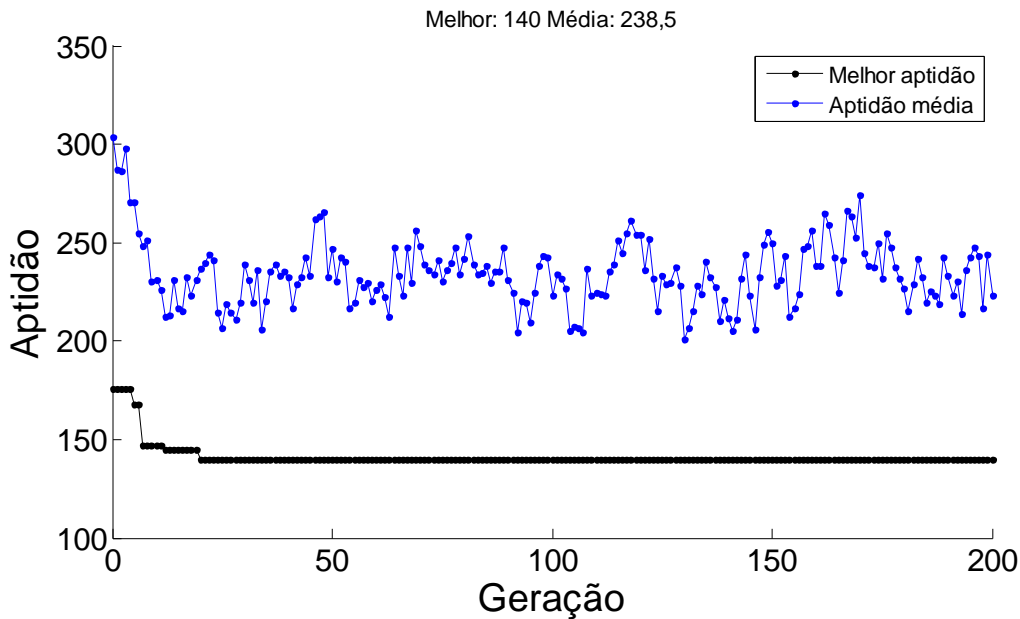


Figura 2 – Variação da maior aptidão e aptidão média dos elementos da população em cada geração para o problema do caixeiro viajante com 8 cidades.

Para testar a eficiência do algoritmo genético e ilustrar graficamente o problema do caixeiro viajante foi realizado um segundo experimento, dessa vez utilizando 60 cidades, distribuídas aleatoriamente em um espaço bidimensional conforme mostrado pelos pontos azuis na Figura 3. Uma tabela de distâncias foi construída a partir da distância euclidiana medida entre cada par de cidades. Dessa vez utilizamos uma população de 100 elementos, sendo que a cada geração dois foram escolhidos por elitismo, 78 gerados por crossover e 20 por mutação. Após 2000 gerações o melhor caminho obtido tem tamanho 8,0978, esse caminho é ilustrado através do traçado vermelho na Figura 3. Podemos observar pela imagem que o caminho obtido talvez não seja o melhor caminho, mas certamente é um bom caminho dentre as mais de 8^{81} soluções possíveis, e foi obtido em uma fração muito pequena do tempo que levaríamos para testar cada uma dessas possibilidades por força bruta.

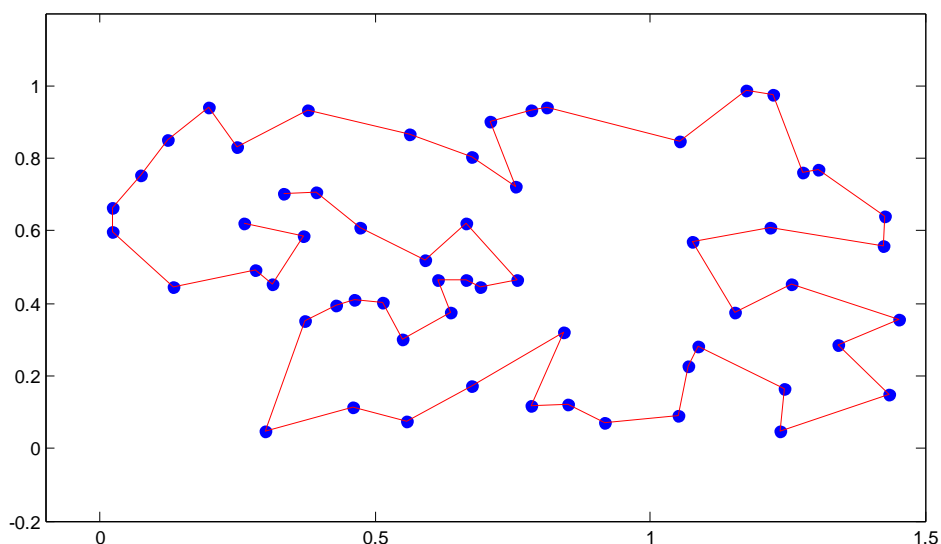


Figura 3 – 60 cidades geradas para o problema do caixeiro viajante (representadas pelos pontos azuis) e o menor caminho obtido pelo algoritmo genético (traçado vermelho).

A Figura 4 mostra a variação da melhor aptidão (menor caminho) e aptidão média dentre os 100 indivíduos de cada geração. Podemos observar que no início há uma forte queda em ambas as linhas, mostrando que rapidamente os caminhos muito ruins vão sendo eliminados e a cada nova geração são obtidos melhores caminhos. A tendência de queda vai diminuindo ao longo das gerações, parecendo estar quase estabilizado em torno de 500 gerações, mas mesmo após 1500 gerações ainda podemos notar ainda uma pequena diminuição no elemento de melhor aptidão.

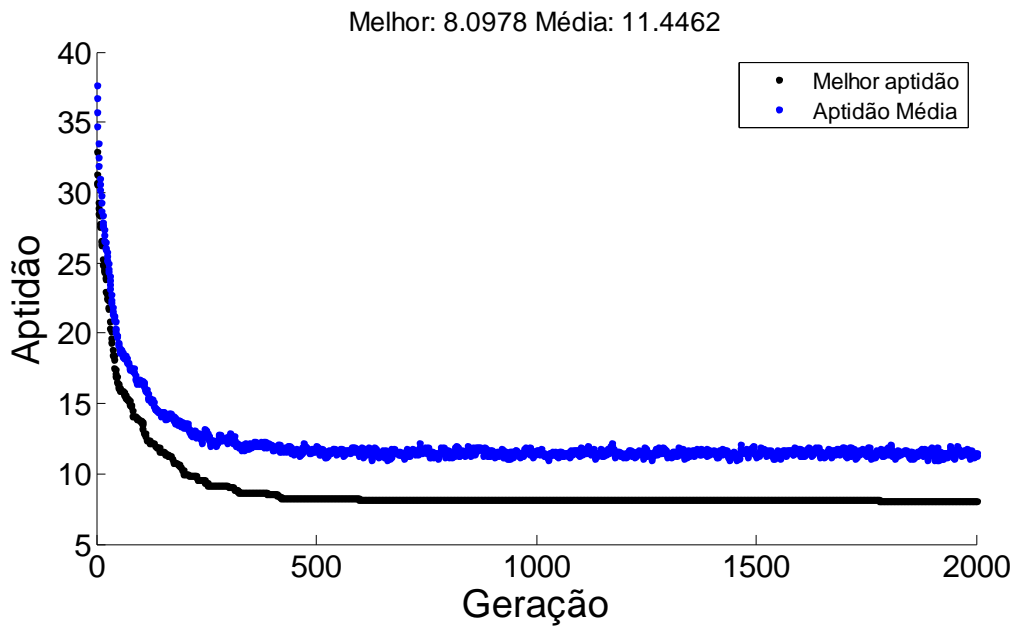


Figura 4 – Variação da maior aptidão e aptidão média dos elementos da população em cada geração para o problema do caixeiro viajante com 60 cidades.

Conclusão

O algoritmo genético utilizado se mostrou um método efetivo para encontrar o valor mínimo do polinômio. Com poucas gerações já foi possível observar que o valor mínimo foi encontrado. O algoritmo genético é uma boa alternativa principalmente em problemas que não apresentam uma solução única e problemas de otimização difíceis para os algoritmos tradicionais, como é o caso do problema do caixeiro viajante. Para muitos problemas do mundo real não é necessário encontrar a solução ótima, uma boa solução em grande parte dos problemas é suficiente, e o algoritmo genético é uma boa opção para encontrar esse tipo de solução.