



Inteligência Artificial

2010/2011

# Implementação de um Algoritmo Genético

-Determinação de circuitos económicos de visita obrigatória a  $n$  pontos com retorno-

The Salesman Problem

André Gonçalves, nº 28451

David Seixo, nº 28316

Lara Luís, nº 28608

Grupo 6

## 1. Introdução

Com este trabalho pretende-se a resolução do problema do caixeiro viajante, recorrendo ao uso de algoritmos genéticos para obter uma solução que se aproximará o mais possível da óptima. Assim, para a implementação do algoritmo foi tido em conta tudo o que é essencial definir para a utilização deste género de algoritmos: representação dos indivíduos da população; método de selecção; operadores de recombinação e mutação; função de mérito e também a definição das características da população (como tamanho, etc).

Para tal, foi implementado um tipo de crossover (OX – order based crossover) e quatro tipos de diferentes mutações: troca, inserção, deslocamento e inversão.

## 2. Modo de funcionamento do algoritmo

A construção da classe representativa do algoritmo genético foi onde encontramos alguns dos maiores problemas na elaboração do projecto. A implementação do método de pesquisa (`search()`) foi realizada recorrendo ao exemplo descrito nos slides das aulas teóricas. Ou seja, a cada iteração, são seleccionados dois indivíduos da população, são gerados números aleatórios, verificando com base nesses números se é realizado crossover, e mutações aos filhos resultantes do crossover, ou no caso de não haver crossover, aos dois indivíduos inicialmente seleccionados. Foram no entanto adicionadas algumas propriedades ao nosso método para tornar a pesquisa pelo melhor indivíduo mais eficiente. Antes do crossover é garantido que nenhum dos pais tem uma fitness igual ao outro, deste modo, garantimos efectivamente que o crossover não produz filhos iguais aos pais.

No entanto, isto produz variações no que toca aos piores indivíduos que podem ser observadas no gráfico, uma vez que as mutação ao qual é obrigado o indivíduo podem conferir-lhe uma fitness pior que a que ele tinha.

São também realizadas as comparações para obter quais os melhores e piores indivíduos e respectivas fitness, e também algumas comparações que têm em conta o valor de fitness dos filhos gerados na respectiva geração e o último melhor indivíduo guardado. Assim, pretende saber-se qual entre os três (dois filhos e o anterior melhor indivíduo) são realmente os dois melhores indivíduos da geração e apenas esses são passados para a próxima.

Antes da conversão da população na nova população é realizada uma mutação do melhor indivíduo. Se o novo indivíduo gerado for melhor que o anterior passa a ser o melhor indivíduo e é adicionado na população. Caso contrario, é descartado.

### 3. Os melhores resultados

- Cinco linha

Crossover	Mutação	# elite	# população	# ciclos	P.Crossover	P.mutação
OX2	Inversão	1	2	10	0.2	0.001

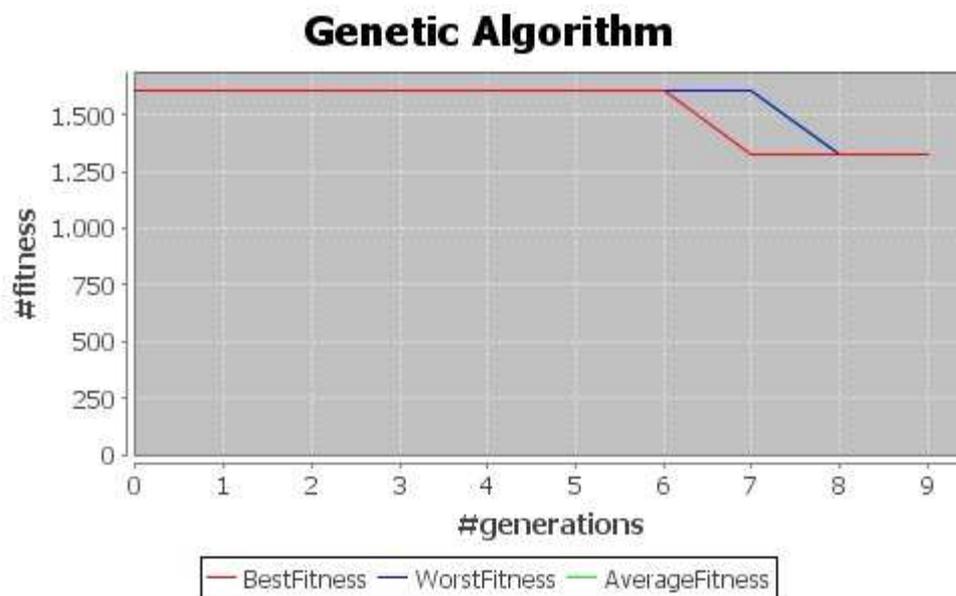
**Melhor cromossoma:** [ 2; 1; 0; 4; 3 ] **fitness:** 1328.0

**Geração em que ocorreu:** 0

**Pior cromossoma:** [ 1; 2; 0; 4; 3 ] **fitness:** 1610.0

**Tempo de execução:** 0 segundos

**Gráfico com curvas de aprendizagem:**



Depois de vários testes com diferentes valores de parâmetros e de tipos de mutação, conclui-se que o valor óptimo é sempre atingido na primeira geração, em zero segundos. Apenas há uma variação no resultado referente à fitness do pior indivíduo. Não há parâmetro ou método de mutação que seja maior influência neste aspecto pois em todos os casos analisados se revelaram essas variações.

- Cinquenta anéis

Crossover	Mutação	# elite	# população	# ciclos	P.Crossover	P.mutação
OX2	Inversão	10	100	400	0.6	0.001

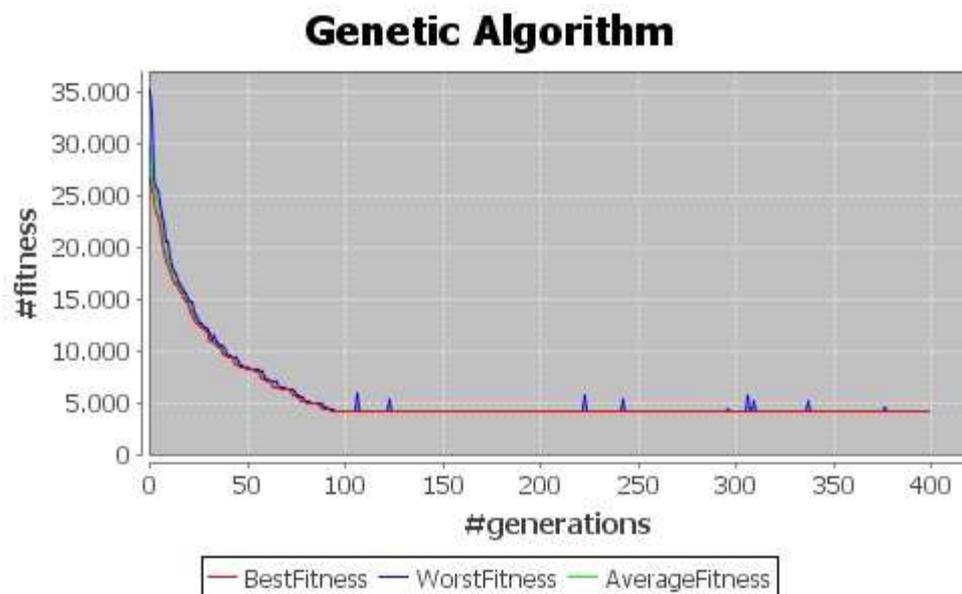
**Melhor cromossoma:** [ 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32 ] **fitness: 4116.0**

**Geração em que ocorreu:** 92

**Pior cromossoma:** [ 40; 0; 12; 9; 37; 23; 14; 38; 7; 27; 10; 48; 11; 34; 21; 6; 29; 22; 32; 18; 41; 13; 5; 28; 19; 45; 43; 26; 15; 2; 36; 47; 46; 25; 42; 1; 17; 3; 30; 4; 31; 35; 44; 8; 33; 49; 16; 20; 39; 24 ] **fitness: 38114.0**

**Tempo de execução:** 27 segundos

**Gráfico com curvas de aprendizagem:**



Os melhores resultados foram obtidos ao usar o operador de mutação de inversão.

## 4. Resultados experimentais

- Resultados médios

Exemplo 1) Entre os testes realizados, uma das alterações na parametrização foi a de redução do valor de elite. Esta pequena variação ficou com um número de gerações intermédio em comparação com os melhores e os piores resultados.

Crossover	Mutação	# elite	# população	# ciclos	P.Crossover	P.mutação
OX2	Troca	5	100	700	0.6	0.001

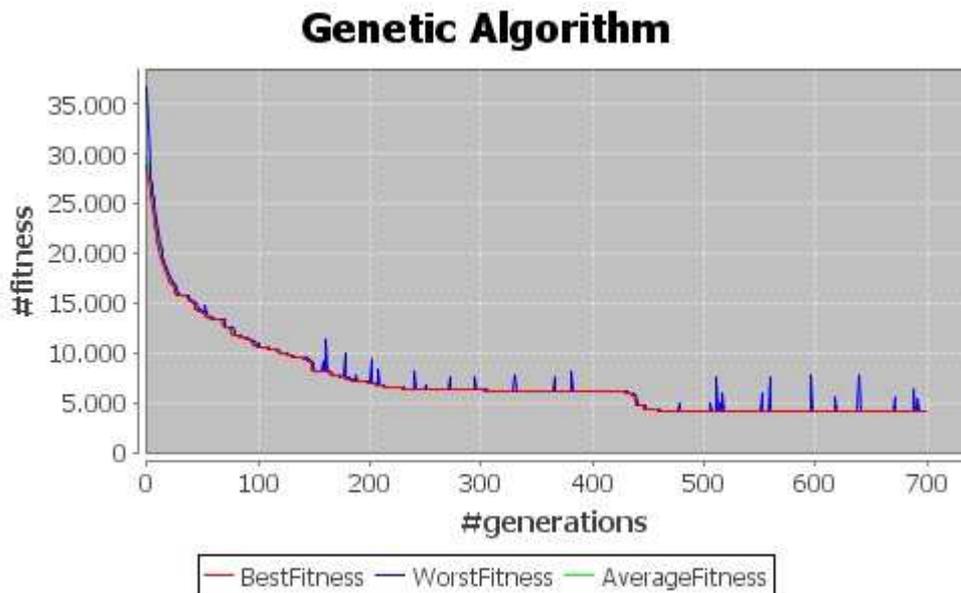
**Melhor cromossoma:** [ 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9 ] **fitness: 4116.0**

**Geração em que ocorreu:** 468

**Pior cromossoma:** [ 31; 44; 9; 26; 10; 40; 7; 49; 23; 45; 18; 11; 27; 35; 16; 28; 8; 1; 29; 41; 39; 14; 32; 12; 2; 33; 5; 46; 25; 38; 17; 19; 22; 4; 36; 21; 47; 37; 0; 24; 48; 30; 15; 20; 43; 34; 3; 13; 42; 6 ] **fitness: 39587.0**

**Tempo de execução:** 33 segundos

Gráfico com curvas de aprendizagem:



Exemplo 2) Outro dos testes realizados foi tentar compreender qual a influência dos operadores de mutação diferentes. Aqui, vê-se o resultado entre o operador de inserção e de deslocamento.

Crossover	Mutação	# elite	# população	# ciclos	P.Crossover	P.mutação
OX2	Inserção	10	100	1000	0.6	0.001

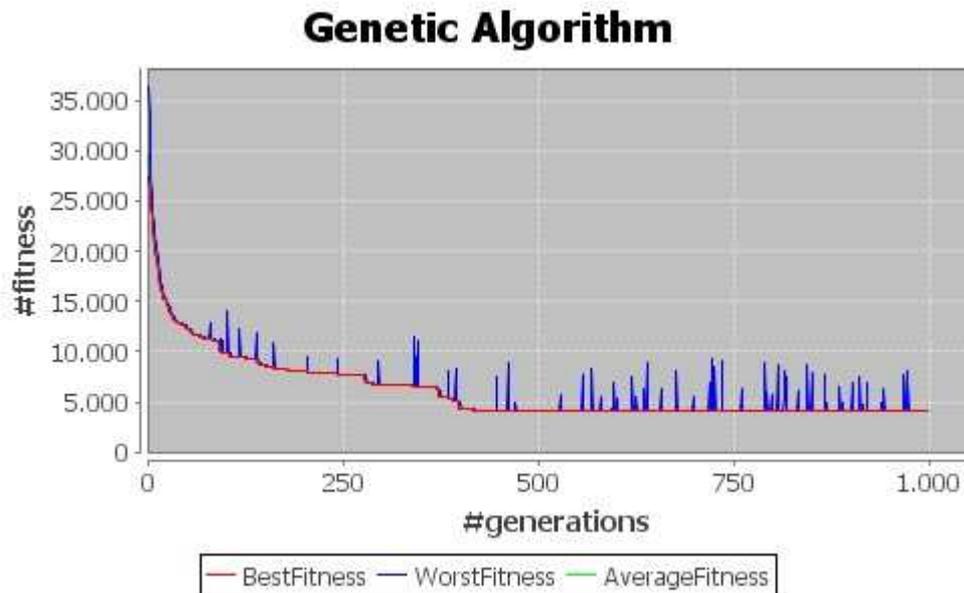
**Melhor cromossoma:** [ 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35;  
36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 0; 1; 2;  
3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20;  
21; 22; 23; 24 ] **fitness:** 4116.0

**Geração em que ocorreu:** 415

**Pior cromossoma:** [ 18; 29; 13; 43; 22; 28; 3; 44; 11; 25; 49; 34;  
30; 33; 16; 2; 19; 24; 20; 31; 48; 5; 46; 41; 21; 6; 37; 27; 15;  
42; 14; 8; 40; 10; 45; 39; 26; 0; 35; 4; 32; 7; 23; 1; 38; 17;  
36; 9; 12; 47 ] **fitness:** 38170.0

**Tempo de execução:** 47 segundos

**Gráfico com curvas de aprendizagem:**



Crossover	Mutação	# elite	# população	# ciclos	P.Crossover	P.mutação
OX2	Deslocamento	10	100	700	0.6	0.001

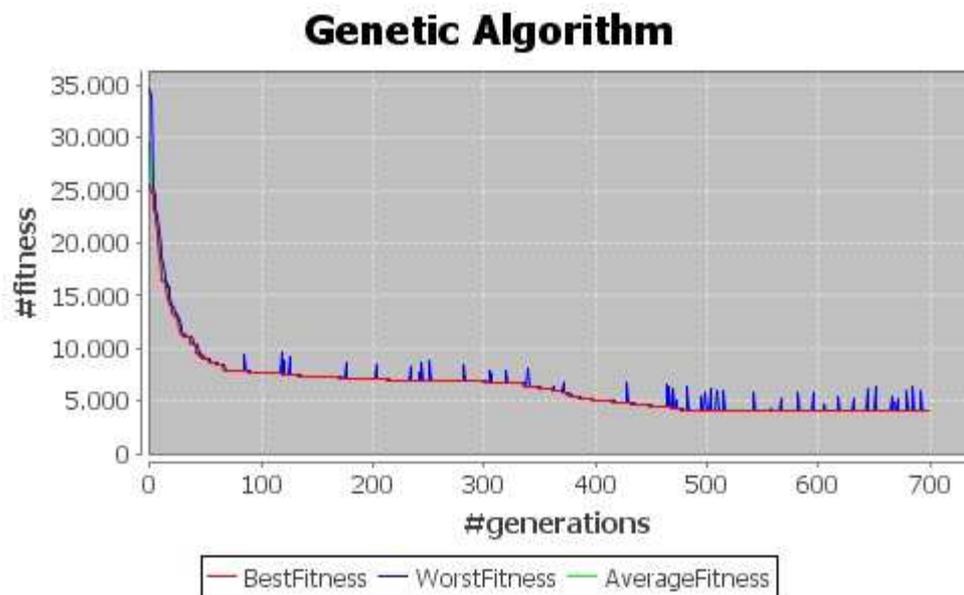
**Melhor cromossoma:** [ 5; 4; 3; 2; 1; 0; 49; 48; 47; 46; 45; 44; 43; 42; 41; 40; 39; 38; 37; 36; 35; 34; 33; 32; 31; 30; 29; 28; 27; 26; 25; 24; 23; 22; 21; 20; 19; 18; 17; 16; 15; 14; 13; 12; 11; 10; 9; 8; 7; 6 ] **fitness:** 4116.0

**Geração em que ocorreu:** 476

**Pior cromossoma:** [ 18; 29; 13; 43; 22; 28; 3; 44; 11; 25; 49; 34; 30; 33; 16; 2; 19; 24; 20; 31; 48; 5; 46; 41; 21; 6; 37; 27; 15; 42; 14; 8; 40; 10; 45; 39; 26; 0; 35; 4; 32; 7; 23; 1; 38; 17; 36; 9; 12; 47 ] **fitness:** 38297.0

**Tempo de execução:** 38 segundos

**Gráfico com curvas de aprendizagem:**



Assim, os resultados obtidos não revelaram grandes diferenças entre os dois operadores pois ambos chegaram à solução óptima num número semelhante de iterações.

Exemplo 3) Tentou combinar-se uma parametrização favorável com uma desfavorável, neste caso, reduzindo o crossover e utilizando o operador de mutação de inversão.

Crossover	Mutação	# elite	# população	# ciclos	P.Crossover	P.mutação
OX2	Inversão	10	100	700	0.3	0.001

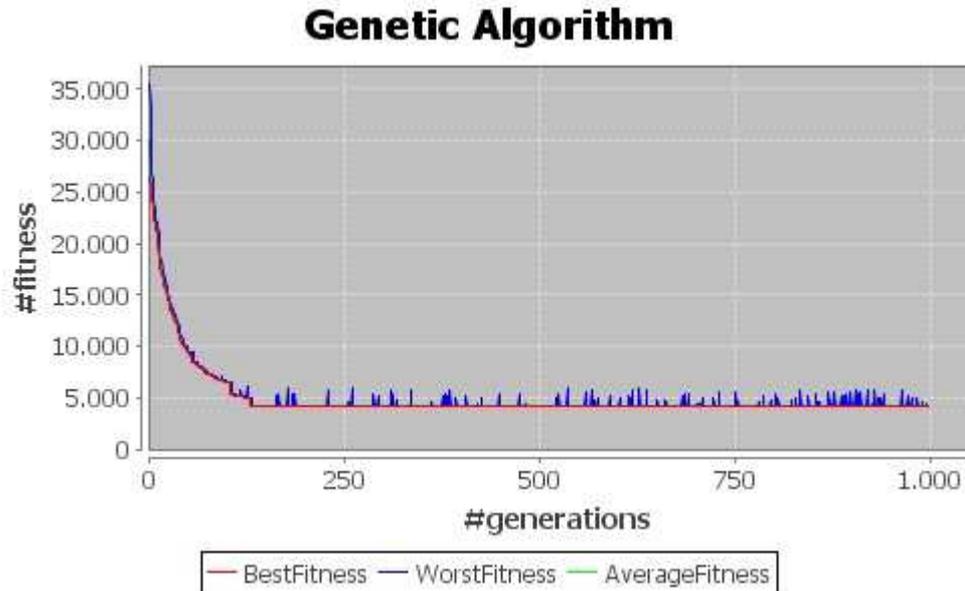
**Melhor cromossoma:** [ 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 0; 1; 2; 3; 4; 5; 6 ] **fitness:** 4116.0

**Geração em que ocorreu:** 153

**Pior cromossoma:** [ 12; 7; 36; 10; 18; 27; 38; 29; 34; 49; 32; 13; 3; 26; 44; 30; 16; 2; 0; 47; 23; 43; 35; 14; 48; 21; 17; 8; 46; 41; 11; 5; 45; 28; 1; 22; 37; 24; 6; 33; 9; 4; 25; 15; 40; 20; 31; 42; 19; 39 ]      **fitness:** 37634.0

**Tempo de execução:** 61 segundos

**Gráfico com curvas de aprendizagem:**



Pode-se concluir que esta combinação resultou num número de gerações intermédio comparando com os melhores e piores resultados.

- **Piores resultados**

Exemplo 1) Alguns dos piores resultados foram obtidos quando foi aumentado o parâmetro de probabilidade de mutação, fazendo com que haja mais probabilidade de ocorrer a mutação nos filhos.

Crossover	Mutação	# elite	# população	# ciclos	P.Crossover	P.mutação
OX2	Troca	10	100	1200	0.6	0.4

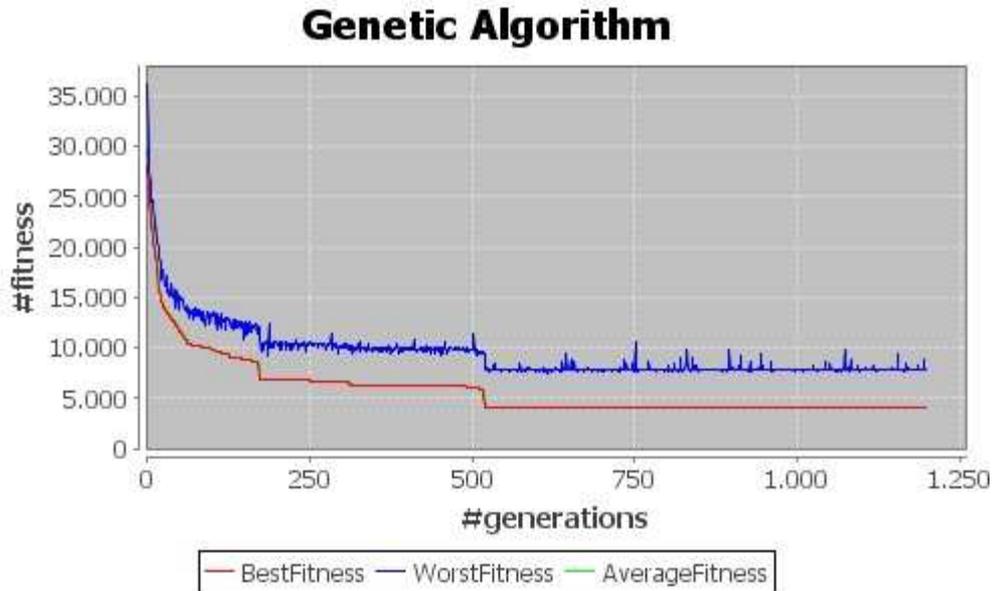
**Melhor cromossoma:** [ 18; 17; 16; 15; 14; 13; 12; 11; 10; 9; 8; 7; 6; 5; 4; 3; 2; 1; 0; 49; 48; 47; 46; 45; 44; 43; 42; 41; 40; 39; 38; 37; 36; 35; 34; 33; 32; 31; 30; 29; 28; 27; 26; 25; 24; 23; 22; 21; 20; 19 ]      **fitness:** 4116.0

**Geração em que ocorreu:** 521

**Pior cromossoma:** [ 32; 36; 3; 41; 30; 23; 13; 6; 12; 31; 0; 18; 29; 8; 20; 42; 26; 1; 34; 16; 38; 14; 44; 39; 48; 28; 11; 17; 47; 19; 27; 9; 24; 43; 37; 15; 2; 7; 22; 49; 33; 40; 25; 46; 35; 45; 5; 21; 4; 10 ] **fitness:** 37979.0

**Tempo de execução:** 147 segundos

**Gráfico com curvas de aprendizagem:**



Exemplo 2) Outros maus resultados surgiram ao reduzir a probabilidade de crossover, levando a uma menor recombinação nos cromossomas, levando a uma variedade de combinações de cromossomas mais demoradas o que leva a um número mais elevado de ciclos para atingir o valor óptimo.

Crossover	Mutação	# elite	# população	# ciclos	P.Crossover	P.mutação
OX2	Troca	10	100	1000	0.2	0.001

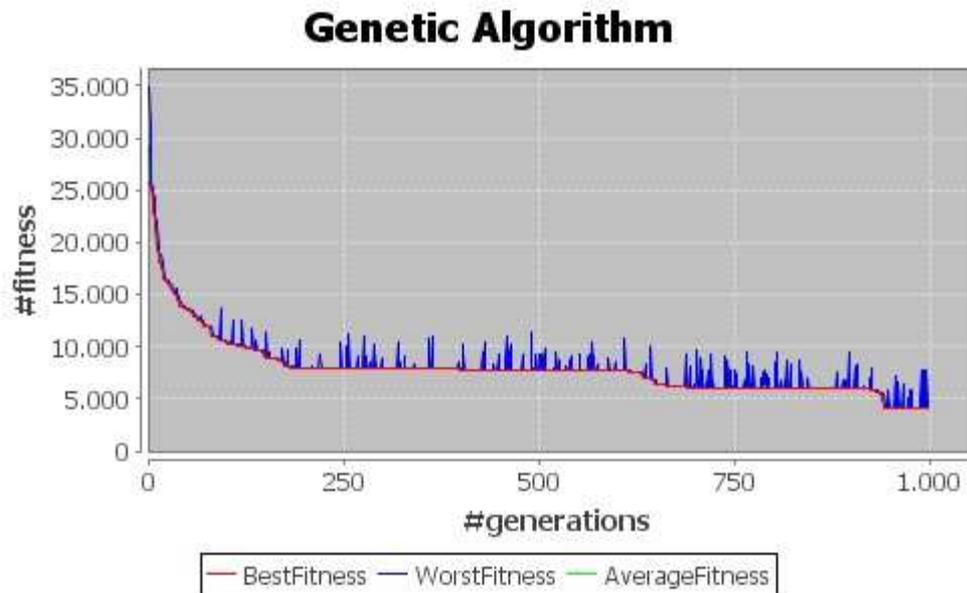
**Melhor cromossoma:** [ 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23 ] **fitness:** 4116.0

**Geração em que ocorreu:** 942

**Pior cromossoma:** [ 29; 36; 18; 37; 0; 13; 20; 42; 40; 33; 19; 48; 30; 15; 41; 5; 25; 9; 38; 45; 27; 8; 14; 17; 44; 1; 26; 3; 39; 6; 4; 16; 34; 10; 31; 47; 28; 49; 23; 46; 32; 2; 12; 11; 21; 35; 7; 43; 22; 24 ] **fitness:** 37744.0

**Tempo de execução:** 60 segundos

Gráfico com curvas de aprendizagem:



## 5. Hill-Climbing

O método Hill-Climbing realiza modificações para obter os cromossomas vizinhos, tentando encontrar um vizinho com melhor fitness que o actual para poder realizar a substituição. Este método facilita a determinação de uma solução obtida embora isso não seja garantido.

Relativamente à aplicação deste método ao problema aqui tratado, o Hill-Climbing ou Simulated Annealing, este iria decrementar o número de gerações até ser encontrado o melhor individuo mas poderia não trazer mais vantagens pois a sua implementação poderia até ser bastante eficiente a encontrar a solução óptima mas isto já é feito de maneira muito satisfatória sem a necessidade da utilização de Hill-Climbing, recorrendo apenas a alguns valores favoráveis e ao operador de mutação de inversão como já referido anteriormente. Este método poderia ter a consequência de aumentar o tempo de execução do programa.

Assim, devido ao pouco tempo que tivemos para implementar funções extra no código, optámos por implementar os métodos de mutação e não Hill-Climbing ou Simulated Annealing.

## 6. Conclusões

Todas as conclusões relativamente às experiências realizadas encontram-se junto ao exemplo em questão.

Para finalizar resta apenas mencionar que, com alguma pena, não foram realizados todos os operadores de crossover por falta de tempo.