

# Relatório de Inteligência Artificial

Andreia Filipa Gomes Fernandes

Número: 2160844

2160844y.ipleiria.pt

## SUMÁRIO

Este projeto foi efetuado no desenvolvimento de algoritmos genéticos que determinam as ações efetuadas por um agente no decorrer do jogo do *Catch*.

Tendo como objetivo apanhar os objetos sem colidir com os obstáculos, tentando otimizar a distância percorrida/o número de passos até sair ela porta. Para tal, foram desenvolvidos algoritmos genéticos de mutação e de recombinação.

Para maximizar a performance do agente deve-se optar por um tamanho de população elevado, o tamanho do torneio deverá ser da ordem de 10% do total de indivíduos de uma população.

Os operadores de mutação deverão ser aplicados com uma probabilidade de 0,01%, com uma probabilidade de recombinação mais elevada.

## 1. INTRODUÇÃO

O número máximo de iterações encontra-se pré-definido num *dataset*, e poderá ser modificado para a realização de experiências que visam estudar a sua influência na performance de um agente. Considera-se que o agente consegue percecionar os seus quatro sentidos adjacentes (Norte, Sul, Este, Oeste).

Cumprindo o requisito principal de maximizar o desempenho do agente, foi considerado em cada controlador que, o agente tem como objetivo deslocar-se até à porta optimizando o número de passos e a distância percorrida.

## 2. DESCRIÇÃO DO ESTADO E DA HEURÍSTICA

### 2.1. Descrição do estado

O agente tem acesso à localização das caixas na grelha e desloca-se, sempre que possível, para essa posição.

O controlador decide sempre que possível pelo movimento que corresponda ao caminho mais curto em direção às caixas, até à porta, analisando as linhas e as colunas.

O agente desloca-se para Norte, caso não seja possível desloca-se para Oeste, Este ou Sul.

O jogo foi “manipulado” para que o agente não colida com as paredes. Esta situação foi considerada de acordo com o pressuposto inicial de que se o agente perceciona as quatro direções adjacentes, então sabe onde está a parede quando esta está adjacente, e não vai contra a parede.

### 2.2 Descrição da heurística

A heurística permite melhorar a eficiência da procura.

Neste caso, a heurística desenvolvida permite calcular a distância existente entre o agente (o carinho) e o objetivo (chegar à porta).

Calculo da heurística:

$|Linha\ do\ Catch - Linha\ do\ Objeto| + coluna\ do\ Catch - Linha\ do\ Objeto|$

## 3. DESCRIÇÃO DOS INDIVÍDUOS

### UTILIZADOS NO ALGORITMO GENÉTICO

Um individuo é representado por um “problema” e pela “fitness”. O Problema contem uma *LinkedList* com todas as caixas na ordem que elas serão apanhadas, com o custo de cada par (distancia da cell1 à cell2), contem o “catch”. A fitness corresponde ao valor do individuo como solução, quanto mais baixa a fitness melhor.

## 4. DESCRIÇÃO DA FUNÇÃO DE AVALIAÇÃO UTILIZADA

A função de avaliação calcula o valor de fitness de cada indivíduo, movimentos efetuados e movimentos com desvio (ocorrem quando o agente se afasta dos objetos).

O valor de desvio é usado para evitar situações em que o agente anda em círculos, sempre nas mesmas posições sem nunca apanhar objetos mas ainda assim é considerado muito bom.

O fitness é calculado Segundo a formula:

$\text{fitness} = \text{valorFitness} ? 0 : (\text{fitness} < \text{valorFitness}) ? 1 : -1$

## 5. DESCRIÇÃO DA CRIAÇÃO DA POPULAÇÃO INICIAL

A população inicial é composta por um conjunto de indivíduos do qual o total pode ser definido na interface pelo utilizador.

Verificou-se que este total tem influência na performance final evidenciada pelos indivíduos, sendo os resultados apresentados e analisados mais à frente.

À medida que a população vai evoluindo, através de seleção e aplicação de operadores genéticos, estes valores vão sofrendo alterações para a performance do agente seja melhor.

## 6. OPERADORES GENÉTICOS

O objetivo da aplicação de operadores genéticos a várias gerações é conseguir gerar novas populações para que os indivíduos possam melhorar a sua performance, permitindo explorar zonas distintas do espaço de procura.

Os operadores são aplicados de forma estocástica e permitem que a população se diversifique, mas mantendo características das gerações anteriores.

## 6.1 Recombinação

A recombinação é o operador responsável pelo cruzamento dos genes dos indivíduos selecionados de uma população.

Por norma, este operador é considerado predominante, de modo a que os seus valores de probabilidade devem ser mais altos, comparativamente à mutação.

### 6.1.1. Recombinação Order 1

Uma faixa de genes consecutivos do pai 1 cai e os valores restantes são colocados no filho na ordem em que aparecem no pai 2.

Parent 1: 8 4 7 3 6 2 5 1 9 0

Parent 2: 0 1 2 3 4 5 6 7 8 9

Child 1: 0 4 7 3 6 2 5 1 8 9

### 6.1.2. Recombinação de Um Corte (One Cut)

É escolhido um ponto de corte nos organismos progenitores de forma aleatória e a partir desse ponto são trocados os genes dos indivíduos. E resultam filhos.

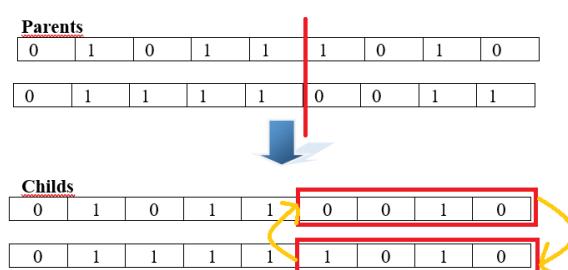


Figura 1 - Esquematização da recombinação de um corte

## 6.2 Mutação

A mutação, normalmente aplicada depois da recombinação, é o operador responsável por introduzir novas características na população, influenciando fortemente a exploração de outras zonas do espaço.

A sua aplicação ocorre ao nível do gene e a probabilidade associada à sua ocorrência deve ser baixa.

Este operador pode ser considerado secundário, e é necessário ter em conta que não há garantias que os indivíduos resultantes sejam mais aptos ou que possibilitem a evolução da população, mas no entanto sabe-se que se a probabilidade da mutação for alta a população seguinte apresenta poucas ou nenhuma características em comum com a população que a gerou o que torna a pesquisa aleatória.

### 6.2.1. Mutação Inversa

Na mutação é selecionado um subconjunto de genes, a sequência é invertida inteira no subconjunto

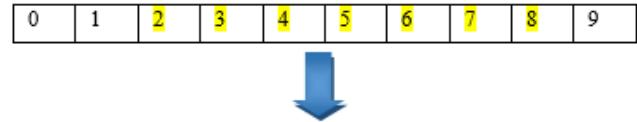


Figura 2 - Esquematização da Mutação Inversa

### 6.2.2. Mutação Swap

São selecionados duas posições aleatoriamente e os valores são trocados.

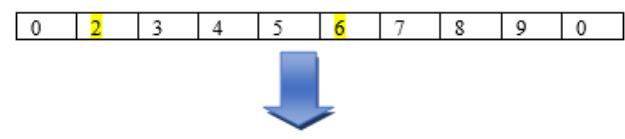


Figura 3 - Esquematização da Mutação Swap

## 7. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

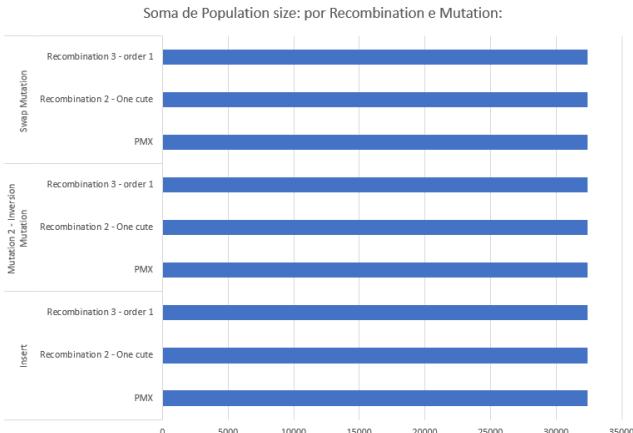
Fechas o programa, dizes que não foram feitos os testes até ao fim nesse dataset

E conclui com os testes inacabadas

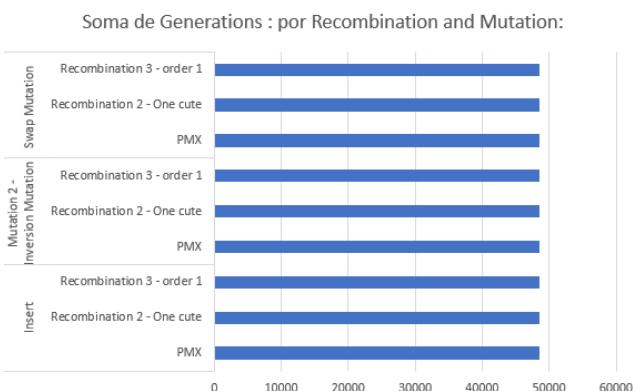
### 7.1. Análise do DataSet1

Para o DataSet1, foram recolhidos dados com populações de 100, 200 e 300 indivíduos, com gerações de 100 e 500, foi usado um tournament de 2 e 4. Na recombinação foram aplicadas probabilidades de 0.1 e 0.4, e na da mutação foram aplicadas probabilidades de 0.2, 0.35 e 0.4. Para o DataSet foram usadas 30 simulações para os 6 algoritmos genéticos.

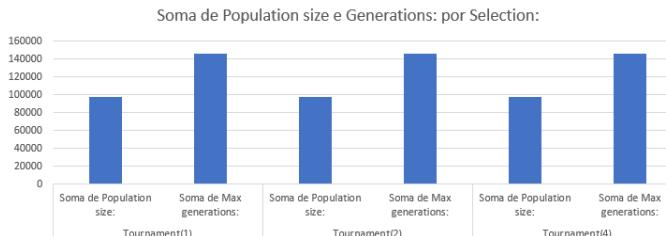
Verificou-se uma performance constante, porque o puzzle para este DataSet é pequeno, por isso não tem muito espaço de procura. O fitness e o average têm sempre um valor de 16.0 com um desvio de 0.0, ou seja encontra sempre a melhor solução, e o caminho explorado é sempre o mesmo: 1 → 2 → 3.



**Gráfico 1 - Soma da população por Recombinação e Mutação**



**Gráfico 2 - Soma das Gerações Por Mutação/Recombinação**

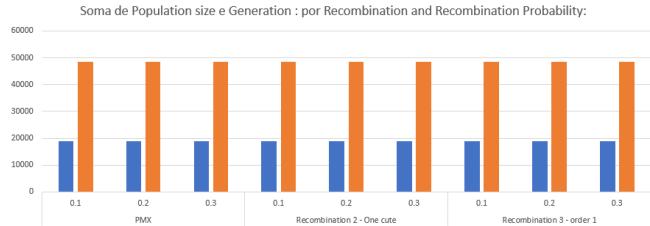


**Gráfico 3 - Soma da população/geração por Tournamant**

## 7.2. Análise do DataSet2

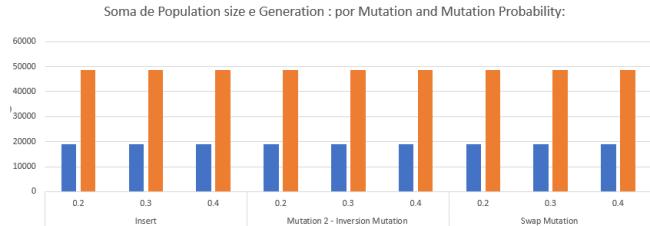
Para o DataSet2, foram recolhidos dados com populações de 50, 100 e 200 indivíduos, com gerações de 100 e 500, foi usado um tournament de 1, 2 e 4. Na recombinação foram aplicadas probabilidades de 0.1, 0.2 e 0.3, e na da mutação foram aplicadas probabilidades de 0.2, 0.3 e 0.4. Para o DataSet foram usadas 30 simulações para os 6 algoritmos genéticos.

Verificou-se uma performance constante, porque o puzzle para este DataSet é pequeno, por isso não tem muito espaço de procura. O fitness e o average têm sempre um valor de 19.0 com um desvio de 0.0, ou seja encontra sempre a melhor solução, e o caminho explorado é sempre o mesmo: 1 → 3 → 4 → 2.



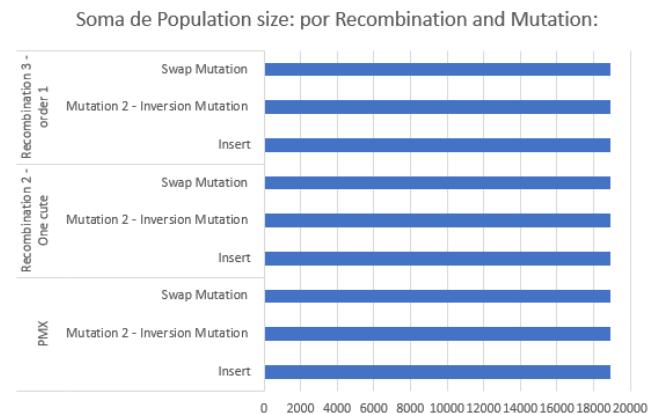
**Gráfico 4 - População e Geração por Recombinação e Probabilidade de Recombinação**

(Max Generation – laranja, Population size - azul)

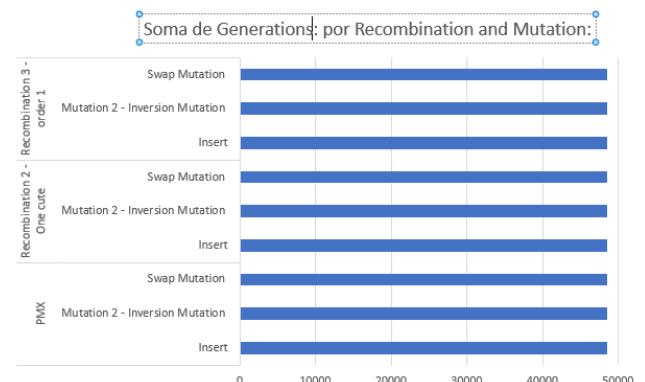


**Gráfico 5 - População e Geração por Mutação e Probabilidade de Mutação**

(Max Generation – laranja, Population size - azul)



**Gráfico 6 - População Por Mutação e Recombinação**



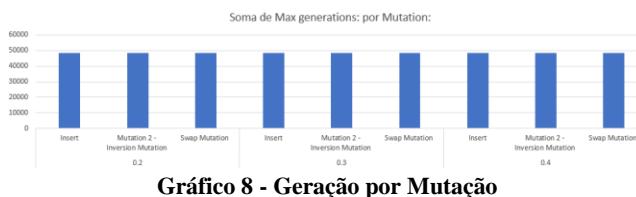
**Gráfico 7 - Geração por Recombinação e Mutação**

### 7.3. Análise do DataSet3

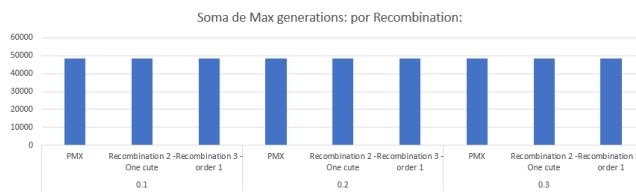
Para o DataSet3, foram recolhidos dados com populações de 50, 100 e 200 indivíduos, com gerações de 100 e 500, foi usado um tournament de 1, 2 e 4. Na recombinação foram aplicadas probabilidades de 0.1, 0.2 e 0.3, e na da mutação foram aplicadas probabilidades de 0.2, 0.3 e 0.4. Para o DataSet foram usadas 30 simulações para os 6 algoritmos genéticos.

Verificou-se uma quase performance constante, porque o puzzle para este DataSet é pequeno, por isso não tem muito espaço de procura. O fitness tem sempre um valor de 17.0 e o average tem quase sempre um valor de 17.0 com um desvio de 0.0, ou seja encontra quase sempre a melhor solução, e o caminho explorado é quase sempre o mesmo: 1 → 2 → 3 → 4.

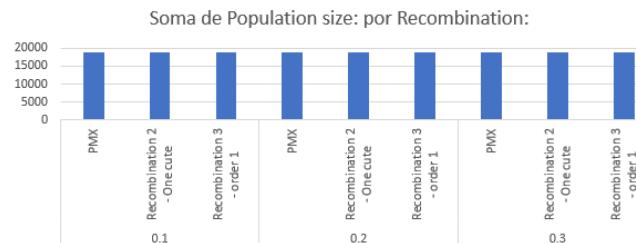
Em 1459 resultados há 31 variações, por isso não se pode considerar que influencia os resultados em grande escala. O average quando há um desvio é de aproximadamente 17.13 e de 17.06.



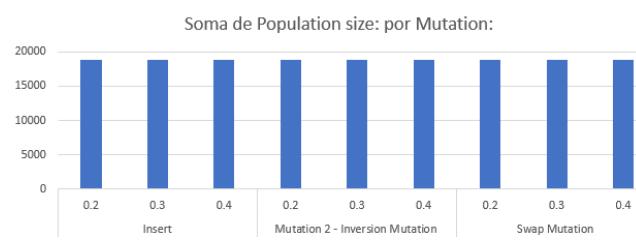
### **Gráfico 8 - Geração por Mutação**



### **Gráfico 9 - Geração por Recombinação**



### **Gráfico 10 - População por Recombinação**



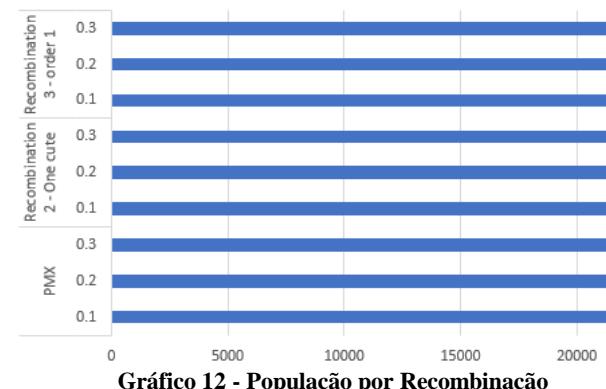
**Gráfico 11 - População por Mutação**

## 7.4. Análise do DataSet4

Para o DataSet4, foram recolhidos dados com populações de 100, 200 e 300 indivíduos, com gerações de 100 e 500, foi usado um tournament de 2 e 4. Na recombinação foram aplicadas probabilidades de 0,1, 0,2 e 0,3, e na da mutação foram aplicadas probabilidades de 0,2, 0,3 e 0,4. Para o DataSet foram usadas 30 simulações para os 6 algoritmos genéticos.

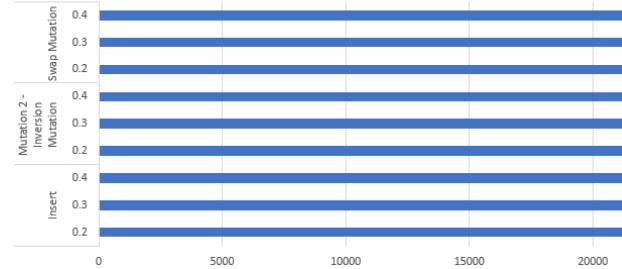
Verificou-se uma quase performance constante, porque o puzzle para este DataSet é pequeno, por isso não tem muito espaço de procura. O fitness tem sempre um valor de 15.0 e o average tem quase um valor de 15.0 com um desvio de 0.0, ou seja encontra quase sempre a melhor solução, e o caminho explorado é sempre o mesmo: 5 → 6 → 3 → 4 → 2 → 1.

Soma de Population size: por Recombination:



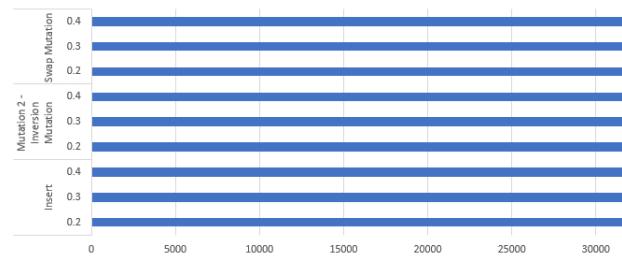
### **Gráfico 12 - População por Recombinação**

Soma de Population size: por Mutation:



**Gráfico 13 - População por Mutação**

Soma de Generations: por Mutation:



#### **Gráfico 14 - Geração por Mutação**

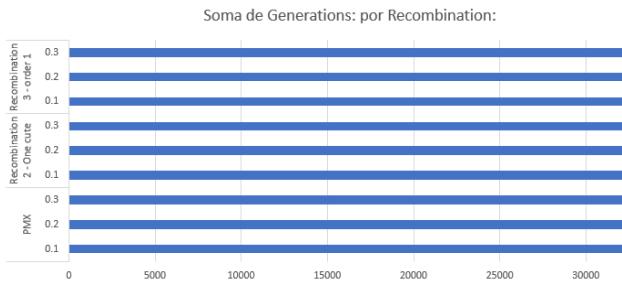


Gráfico 15 - Geração por Recombinação

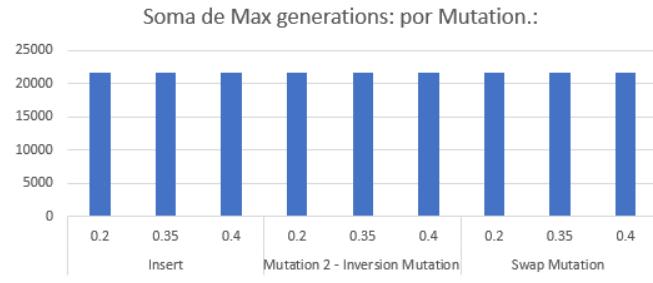


Gráfico 18 - Geração por Mutação

## 7.5. Análise do DataSet5

Para o DataSet5, foram recolhidos dados com populações de 100, 200 e 300 indivíduos, com gerações de 100 e 500, foi usado um tournament de 2 e 4. Na recombinação foram aplicadas probabilidades de 0.1 e 0.2 e na da mutação foram aplicadas probabilidades de 0.2, 0.35 e 0.4. Para o DataSet foram usadas 30 simulações para os 6 algoritmos genéticos.

O puzzle para este DataSet é mediano, ou seja o agente tem mais espaço de procura, o que faz com que os resultados se desviam ligeiramente do resultado ideal. O desvio varia entre 0.6 e 5.72. O melhor caminho obtido foi com um average de 45.0 (desvio de 0.0). O pior resultado foi 51.33, com um desvio de 5.72. Os melhores resultados foram sempre obtidos com a mutação Inversa. Os piores resultados variaram entre a mutação swap e a insert. O fitness obtido foi sempre de 45.0.

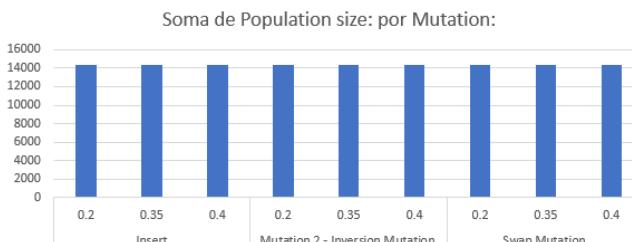


Gráfico 16 - População por Mutação

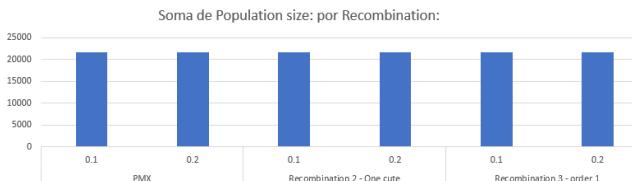


Gráfico 17 - População por Recombinação

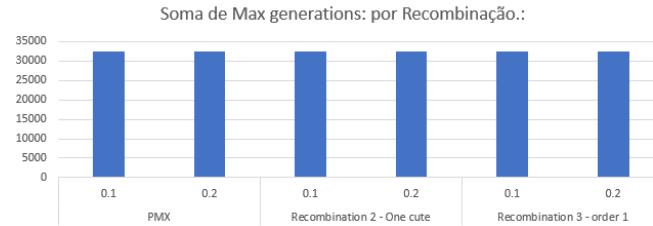


Gráfico 19 - Geração por Recombinação

## 7.6. Análise do DataSet6

Teste não terminou a tempo, como era um teste grande e faltavam 45 linhas, não foi considerado que influenciasse os resultados. O dataSet6 sofre diversas variações, porque é grande, ou seja o indivíduo tem mais espaço de procura.

Para o DataSet6, foram recolhidos dados com populações de 100, 200 e 400 indivíduos, com gerações de 100, 500 e 700, foi usado um tournament de 2 e 4. Na recombinação foram aplicadas probabilidades de 0.2, 0.3, 0.4 e 0.5 e na da mutação foram aplicadas probabilidades de 0.2, 0.3 e 0.4.

Para o DataSet foram usadas 30 simulações para os 6 algoritmos genéticos.

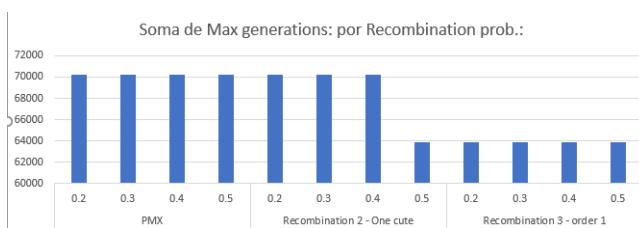
O melhor resultado obtido foi de 91.6 para uma população de 400 e 200, geração de 700, tournament 4, com a recombinação de Um Corte e order 1 e a mutação Inversa, a probabilidade da recombinação foi de 0.0.2 e 0.4 e a probabilidade da mutação foi de 0.4.

O fitness obtido foi entre 89.0 e 177.0.

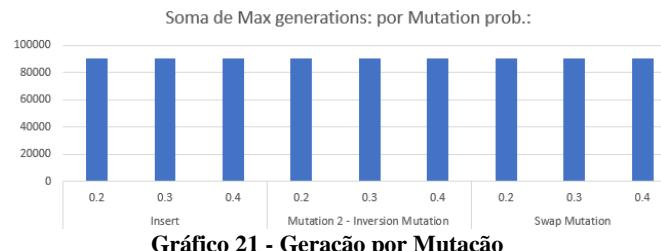
## Soma de Population size: por Recombination and Mutation:



Gráfico 19 - População por Recombinação e População



**Gráfico 20 - Geração por Recombinação**



**Gráfico 21 - Geração por Mutação**

## 7.7. Análise do DataSet7

Teste não terminou a tempo, como era um teste grande e faltavam 173 linhas, não foi considerado que influenciasse os resultados. O dataSet7 sofre variações, porque é grande, ou seja o indivíduo tem mais espaço de procura.

Para o DataSet7, foram recolhidos dados com populações de 200 e 400 indivíduos, com gerações de 250 e 500, foi usado um tournament de 2, 4 e 5. Na recombinação foram aplicadas probabilidades de 0.1, 0.2 e 0.3. Na da mutação foram aplicadas probabilidades de 0.2, 0.4, 0.5.

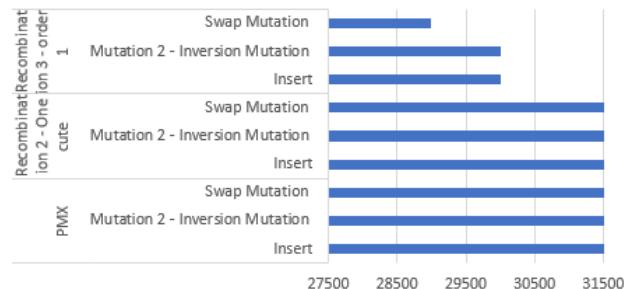
Para o DataSet foram usadas 30 simulações para os 6 algoritmos genéticos.

O puzzle para este DataSet é grande, o que faz com que o agente tenha um espaço de procura muito maior. Por consequência disso os resultados são variados.

O melhor resultado obtido foi 88.4 para uma população de 400, geração 250, tournament 5, com a recombinação de order 1 e a mutação Inversa, a probabilidade da recombinação foi de 0.1, e a probabilidade da mutação foi de 0.5.

O fitness obtido foi entre 86.0 e 148.0.

## Soma de Max generations: por Recombination e Mutation:



**Gráfico 22 - Geração por Recombinação e Mutação**

## 7.8. Análise do DataSet8

Teste não terminou a tempo, como era um teste grande e faltavam 64 linhas, não foi considerado que influenciasse os resultados. O dataSet8 sofre diversas variações, porque é bastante grande, ou seja o indivíduo tem mais espaço de procura.

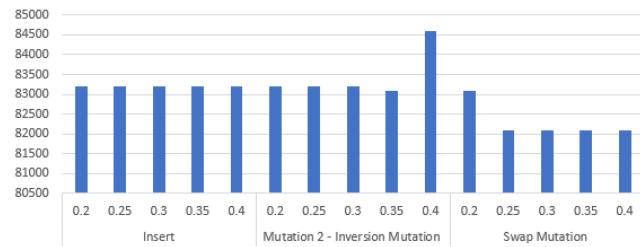
Para o DataSet8, foram recolhidos dados com populações de 50, 100 e 400 indivíduos, com gerações de 100, 500 e 100, foi usado um tournament de 3 e 4. Na recombinação foram aplicadas probabilidades de 0.2, 0.3, 0.4 e 0.5 e na da mutação foram aplicadas probabilidades de 0.2, 0.25, 0.3, 0.35 e 0.4.

Para o DataSet foram usadas 50 simulações para os 6 algoritmos genéticos.

O melhor resultado obtido foi de 101.76 para uma população de 400, geração 1000, tournament 4, com a recombinação de Um Corte e a mutação Inversa, a probabilidade da recombinação foi de 0.1, e a probabilidade da mutação foi de 0.4

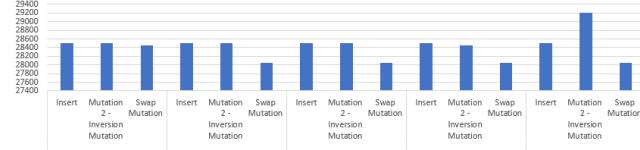
O fitness obtido foi entre 98.0 e 218.0.

## Soma de Max generations: por Mutation :



**Gráfico 23 - Geração por Mutação**

## Soma de População: por Mutation :



**Gráfico 24- População por Mutação**

Soma de Max generations: por Recombination:



Gráfico 25 - Geração por Recombinação

## 8. Conclusão

Da analise dos testes todos, pode-se concluir que quando mais alto o tournament, melhor a performance do indivíduo.

O fitness também tem melhor solução quando a população e a geração são mais elevadas.

E a mutação com que se obtém melhor resultados é a mutação Inversa. Relativamente às Recombinações obtem-se melhores resultados com várias.

## 9. REFERENCIAS

- [1] <http://lipas.uwasa.fi/~timan/EA/EAlecture3b.pdf>
- [2] <http://www.rubicite.com/Tutorials/GeneticAlgorithms/CrossoverOperators/CycleCrossoverOperator.aspx>

	Melhor Fitness	Melhor Average	Desvio	Probabilidade da Mutação	Probabilidade de Recombinação	Geração	População
DataSet1	16.0	16.0	0.0	0.2, 0.3 e 0.4	0.1 e 0.2, 0.3	100 e 500	100, 200 e 300
DataSet2	19.0	19.0	0.0	0.2, 0.3, 0.4	0.1, 0.2, 0.3	100 e 500	50, 100, 200
DataSet3	17.0	17.0	0.0	0.2, 0.3, 0.4	0.1, 0.2, 0.3	100, 500	50, 100, 200
DataSet4	15.0	15.0	0.0	0.2, 0.3, 0.4	0.1, 0.2, 0.3	100, 500	100, 200, 300
DataSet5	45.0	45.0	0.0	0.2, 0.35 e 0.4	0.1 e 0.2	100, 500	100, 200, 300
DataSet6							
DataSet7							
DataSet8							