

Relatório de projeto de Inteligência Artificial

Cátia Antunes
2212492@ipleiria.pt

Francisco Oliveira
2212481@ipleiria.pt

Resumo

Este relatório apresenta um programa desenvolvido para a gestão eficiente de um armazém, indicando os produtos que devem ser recolhidos pelo *forklift* e, posteriormente levados até à saída. O objetivo principal do programa é otimizar os caminhos percorridos pelos *forklift* e analisar os resultados das experiências.

Palavras-chave

Python, *forklift*, produtos, operadores genéticos, métodos de procura

1. INTRODUÇÃO

A gestão eficiente de um armazém é fundamental para garantir a fluidez das operações logísticas e o atendimento de pedidos de forma rápida e precisa. Um dos desafios enfrentados nesse contexto é a localização dos produtos dentro do armazém. Para otimizar esse processo, foi desenvolvido um programa que permite procurar caminhos, atribuindo um custo a cada. Após o cálculo dos caminhos, o software faculta o melhor caminho, o pior e a média de todos os caminhos.

2. DESCRIÇÃO DOS COMPONENTES

2.1 ESTADO

O estado é composto por uma matriz de $n \times m$ de inteiros, por forma a representar um armazém. Este possui *forklifts*, produtos a recolher e uma saída. Estes *forklifts* apenas se podem deslocar pelas posições vagas, sendo que estas se encontram representadas a branco.

Todos os estados contêm a posição inicial de cada *forklift*, os produtos e a saída.

Para perceber melhor o contexto, são apresentados de seguida algumas variáveis presentes no código:

- *Matrix*: guarda a matriz com a posição dos produtos, posições dos *forklifts* e a saída;
- *Forklift*: guarda a posição (*Cell*) do *forklift*.
- *Goal*: guarda a posição (*Cell*) do objetivo pretendido
- *Exit*: guarda a posição (*Cell*) da saída
- *Line_forklift*, *column_forklift*, *line_exit* e *column_exit*: variáveis que permitem associar uma cor à célula da matriz, sendo que cada cor representa um objeto (Tabela 1).

Tabela 1. Descrição das constantes utilizadas

Item	Número	Cor
<i>Shelf</i>	1	Verde
<i>Product</i>	2	Cinzentos
<i>Exit</i>	3	Azul
<i>Forklift</i>	4	Vermelho
<i>Product_Catch</i>	5	Preto
<i>Empty</i>	0	Branco

2.2 HEURÍSTICA

Para a realização do cálculo da heurística, foi selecionada a *Manhattan Distance*, uma vez que, adequa-se ao cálculo de movimentações num espaço retangular. Tendo em conta o problema em questão, esta heurística torna-se a mais conveniente, uma vez que o *forklift* se desloca em linhas horizontais e verticais com o objetivo de recolher os produtos desejados.

2.3 REPRESENTAÇÃO DOS INDIVÍDUOS UTILIZADOS NO ALGORITMO GENÉTICO

2.3.1 Agente

O agente é composto por um ambiente, uma heurística, uma saída e vetores de montacargas, produtos e pares.

Além disso, foram acrescentadas duas variáveis: *Solution_by_pair*, que representa um dicionário no qual guarda como *key* um par de células e como *value* a solução do par utilizado como *key*; *Worst_global_weight*, representado por um inteiro, inicializado a -1, que permite guardar o pior *fitness* das gerações todas.

2.3.2 Indivíduo

O indivíduo é composto por múltiplas variáveis, tais como:

- *Weight*: inteiro que representa o valor da deslocação do *forklift* desde a posição inicial até à posição de saída.
- *Worst_value*: inteiro que guarda a pior deslocação calculada de um indivíduo de um determinado genoma.
- *All_paths_forklift*: matriz de vetores, no qual cada vetor, de uma só linha, representa o caminho efetuado por um *forklift*.
- *Steps_values*: vetor que guarda a quantidade de movimentos que cada *forklift* tem de realizar, até atingir a saída.
- *Forklifts_and_products*: dicionário que guarda as células que o *forklift* irá percorrer.
- *Genome*: vetor de inteiros, que representa os produtos e os *forklifts*. Para a construção deste vetor, teve-se em consideração que, apenas podia gerar números de forma aleatória entre 1 e o tamanho do vetor, sendo que este tamanho seria composto pela adição do número de produtos com o número de *forklifts* menos 1.

Em relação aos métodos da classe, além dos fornecidos pelo *template*, foram adicionados alguns métodos extra que permitiram efetuar operações auxiliares:

- Método *compute_weight*: recebe um vetor como argumento, sendo que este representa os produtos que um *forklift* recolhe. Com esse argumento é criado um par de células e verifica-se se este existe no dicionário *solution_by_pair* presente no problema. Caso não exista, é necessário inverter as células do par, de forma a obter a lista de movimentos. Desta forma, é calculado o custo e é adicionado um novo vetor à matriz *all_path_forklifts*.
- Método *compute_weight_for_path_of_forklift*: tem como objetivo invocar o método *compute_weight* para cada *forklift* presente no genoma.

2.4 Função de avaliação (função de fitness)

A função de fitness é responsável por calcular a aptidão de um indivíduo. Esta função, percorre o genoma do indivíduo, cujo vetor contém os índices dos produtos a recolher e os *forklifts* que os vão recolher, determinando um peso entre cada percurso efetuado pelo *forklift*. Este peso é calculado usando a soma dos custos entre as deslocações entre cada produto e, por fim, até à saída. O pior valor de peso encontrado no genoma, é atribuído como aptidão do indivíduo.

2.5 Método de criação da população inicial

O método de criação da população inicial consiste na descoberta do tamanho do genoma. Para tal, pode-se utilizar o tamanho do vetor dos *forklifts* menos um, e adicionar o tamanho do vetor dos produtos. Após esta operação, são gerados números aleatórios, entre 1 e o tamanho do genoma. No entanto, é necessário verificar se o número gerado, ainda não se encontra no genoma. Caso não

se encontra presente, este é inserido. Este processo realiza-se até que o vetor esteja completo.

2	1	4	5	3
---	---	---	---	---

Figura 1 - Exemplo de uma criação da população inicial

2.6 Operadores genéticos desenvolvidos

Os operadores genéticos responsáveis pela realização da recombinação genética são:

Recombination 2 – Consiste em selecionar um elemento aleatório do indivíduo 1 e um elemento aleatório do indivíduo 2, certificando que são diferentes até o novo indivíduo ter o tamanho desejado. Após este procedimento que permite obter o primeiro novo indivíduo, procede-se à formação do segundo novo indivíduo, o qual é obtido com os restantes elementos que não foram colocados no primeiro novo indivíduo, sendo estes misturados entre si e acrescentados a este.

Recombination 3 – Consiste em dividir os dois genomas em três partes em que, cada primeiro terço é copiado para o filho correspondente. De seguida esses elementos que foram copiados, são eliminados das cópias criadas. Após isso, vamos buscar os elementos restantes e são colocados num vetor novo. De seguida, é preenchido o primeiro filho. Para tal, é selecionado uma posição aleatória deste novo vetor e colocado esse elemento no filho, caso não esteja presente, removendo-o do vetor novo. Este processo é executado até o primeiro filho estar preenchido. Com os valores restantes do novo vetor, é realizado o mesmo procedimento para preencher o segundo filho.

Os operadores genéticos responsáveis pela realização da mutação genética são:

Mutation 2 – Consiste na seleção de uma posição aleatória do genoma, verificação do valor presente nessa posição e, de seguida, troca do valor presente pelo número aleatório gerado.

Nº aleatório gerado: 3

0 1 2 3 4

3	1	4	5	2
---	---	---	---	---

posição[3] = 5

Verificar onde se encontra o 3 e substituir

5	1	4	3	2
---	---	---	---	---

Figura 2 - Exemplo da mutação 2

Mutation 3 – Consiste na seleção de um elemento aleatório do genoma. Remoção desse mesmo elemento do genoma e baralhamento do mesmo. Por fim, colocação do elemento aleatório retirado, numa posição aleatória do genoma.

Nº aleatório gerado: 3

3	1	4	5	2
---	---	---	---	---

1	4	5	2
---	---	---	---

5	1	4	2
---	---	---	---

Nº aleatório gerado: 1

5	3	1	4	2
---	---	---	---	---

Figura 3 - Exemplo da mutação 3

3. Apresentação e discussão dos resultados obtidos

De forma a elaborar a apresentação e discussão dos resultados obtidos, foi utilizado o problema 3, tendo em conta as colisões entre *forklifts*, ou seja, quando estes se encontram, ao mesmo tempo, na mesma célula da matriz. Caso este acontecimento ocorra, foi adicionado o valor de 10 ao custo total do caminho a percorrer pelo *forklift*.

3.1 Teste Geral

Para a realização da discussão dos resultados dos dados do teste geral, foram realizados 40 *runs*, com os valores constantes para a população (50, 100, 200), gerações (100) e *tournament selection* (2, 4, 6, 8). Relativamente às recombinações foi utilizado a recombinação *pmx*, *recombination2* e *recombination3*. Cada uma com as probabilidades a 0.6, 0.7 e 0.8. Por fim foi utilizado a mutação *insert*, *mutation2*, *mutation3* com as probabilidades a 0.1, 0.2 e 0.3.

Através deste teste, conseguimos perceber que o melhor resultado obtido apresenta as seguintes características:

- População: 200;
- Gerações: 100;
- Método seleção: *Tournament selection* (4);
- Recombinação: *PMX recombination* (0.7);
- Mutação: *Insert Mutation* (0.3);
- Média: 135.75;
- Desvio Padrão: 8.3.

O pior resultado gerado:

- População: 50;
- Gerações: 100;
- Método seleção: *Tournament selection* (2);
- Recombinação: *Recombination2* (0.6);
- Mutação: *Mutation2* (0.1);

- Média: 236.45;
- Desvio Padrão: 9.1.

Ao comparar estes resultados, é importante observar que diferentes configurações nos algoritmos genéticos podem levar a desempenhos distintos. Portanto, torna-se importante encontrar uma configuração adequada dos operadores genéticos e as suas probabilidades de forma a conseguir tirar o maior proveito deste.

3.2 Efeito da variação do tamanho e da população e do número de gerações

Para a realização da discussão dos resultados obtidos, foram realizados 40 *runs*, com os valores constantes para o *Tournament selection* (4), *PMX recombination* (0.8) e *Insert Mutation* (0.025). Em termos de variação da população foram testados os seguintes valores: 50, 100, 200, 400 e 500. Em relação à variação do número de gerações, foram considerados os seguintes valores: 12, 15, 30, 60, 120.

De forma a confirmar o resultado obtido anteriormente, foram selecionados os seguintes valores das combinações realizadas nos testes: população (50) e gerações (120), população (100) e gerações (60), população (200) e gerações (30), população (400) e gerações (15) e população (500) e gerações (12).

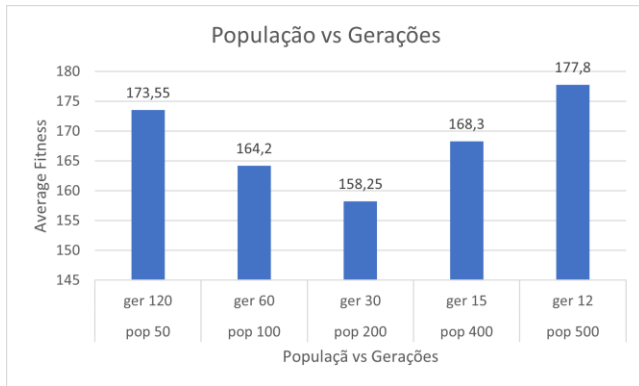
Com base nos resultados da tabela 1 e da figura 1, é possível concluir que as populações que apresenta o melhor resultado (158.25) é a de 200 com 30 gerações. Os piores resultados obtidos foram com: população (500) e geração (12) com um valor de 177.8, e população (50) e geração (120) com 173.55.

Relativamente aos resultados obtidos, é possível concluir que, quanto maior for a população, e se esta utilizar um número de gerações com um valor baixo, maior será a média do *fitness*, uma vez que, como a população apresenta um número elevado e um número baixo de gerações leva a que não seja possível encontrar a solução ótima. O mesmo acontece com populações pequenas e um grande número de gerações. Portanto, é importante encontrar um equilíbrio entre estas duas variáveis de modo a encontrar a solução mais ótima.

Tabela 1 –Valores estatísticos do efeito da variação da população e do número de gerações

População	Nº gerações	Média	Desvio Padrão
50	120	173.55	13.59
100	60	164.2	11.61
200	30	158.25	10.27
400	15	168.3	9.31
500	12	177.8	8.34

Figura 1 – Gráfico de Barras sobre o efeito da Variação da População e das Gerações



3.3 Desempenho relativo dos diferentes operadores genéticos

Para a realização da discussão dos resultados obtidos, foram realizados 40 *runs*, com os valores constantes para a população (200), gerações (50), *Tournament selection* (4). Para a análise dos métodos de recombinação, foram considerados o PMX *recombination*, recombinação 2, recombinação 3, todos com uma probabilidade de 0.8 associada e mutação: *Insert Mutation* (0.2). Em relação à análise dos métodos da mutação foram considerados o *Insert Mutation*, mutação 2 e mutação 3, com uma probabilidade de 0.2 e recombinação com PMX *recombination* (0.8). Uma vez que só pretendemos analisar o seu desempenho em termos de operadores e não a sua probabilidade de acontecer.

Com base na observação da tabela 2 e da figura 2, é possível verificar que, o método de PMX *recombination* com a mutação *Insert* apresentou a média mais baixa (138.8), indicando um desempenho melhor na procura por soluções otimizadas. Além disso, exibiu um desvio padrão mais baixo quando comparado com os outros métodos de recombinação, sugerindo que os resultados são mais consistentes.

Contudo, a *recombination* 2 obteve a média mais alta (225.6), indicando um desempenho inferior na procura por soluções otimizadas.

Com base na observação da tabela 3 e da figura 3, é possível observar que a mutação *Insert* com o PMX *recombination* (0.8) apresentou a média mais baixa (141.4), indicando um desempenho melhor na procura por soluções otimizadas. Além disso, exibiu um desvio padrão mais baixo quando comparado com os outros métodos de recombinação, sugerindo que os resultados são mais consistentes.

Contudo, a *mutation* 3 obteve a média mais alta (163), indicando um desempenho inferior na procura por soluções otimizadas.

Em resumo, os resultados indicam que a escolha dos operadores genéticos pode ter um impacto significativo no desempenho do algoritmo. A recombinação 3 e a mutação *Insert* foram os operadores genéticos que demonstraram desempenhos promissores, alcançando médias de aptidão mais baixas e resultados mais consistentes.

Tabela 2 - Valores estatísticos sobre o desempenho relativo dos operadores de recombinação

Recombinação	Mutação	Média	Desvio Padrão
<i>Recombination</i> 2 (0.8)	<i>Insert Mutation</i> (0.2)	225.6	9.13
<i>Recombination</i> 3 (0.8)	<i>Insert Mutation</i> (0.2)	189.2	7.15
PMX <i>Recombination</i> (0.8)	<i>Insert Mutation</i> (0.2)	138.8	6.54

Tabela 3 - Valores estatísticos sobre o desempenho relativo dos operadores de mutação

Recombinação	Mutação	Média	Desvio Padrão
PMX <i>Recombination</i> (0.8)	<i>Mutation</i> 3 (0.2)	163.0	14.19
PMX <i>Recombination</i> (0.8)	<i>Mutation</i> 2 (0.2)	157.15	10.76
PMX <i>Recombination</i> (0.8)	<i>Insert Mutation</i> (0.2)	141.4	9.84

Figura 2 – Gráfico de Barras sobre os operadores de recombinação

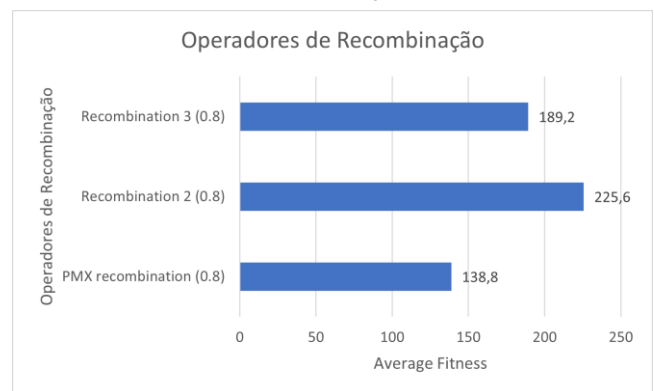
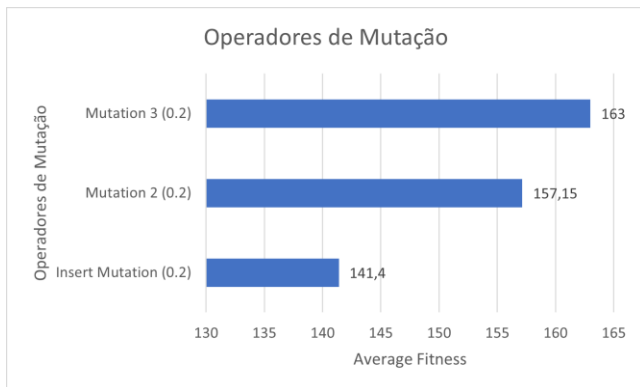


Figura 3 – Gráfico de Barras sobre os operadores de Mutação



3.4 Efeito da variação das probabilidades dos operadores de recombinação

Para a discussão destes resultados foram considerados os valores contantes para as variáveis da população (200), número de gerações (50) e *Tournament selection* (4).

Ao analisar os resultados presentes na tabela 4, é possível observar que relativamente ao operador de recombinação *PMX*, à medida que a taxa vai aumentando, a média diminui gradualmente de 150.9 para 138.1. Isto indica que um valor mais alto de taxa de recombinação resulta em soluções mais ótimas.

No caso da recombinação 2 e 3, à medida que a taxa aumenta, a média também aumenta de 154.6 para 226.6. Isso sugere que, uma taxa mais alta não é tão eficaz em produzir soluções mais ótimas. Estes resultados podem derivar do facto de o operador não ser o mais adequado para este tipo de problema ou os parâmetros de entrada (população, número de gerações e *tournament selection* (4)), não terem sido os ideais.

Sendo que o objetivo é otimizar a solução do problema dos *forklifts* e, os valores de média menores apresentam soluções mais ótimas podemos concluir que para o operador de recombinação *PMX* é preferível uma taxa de recombinação mais alta.

Tabela 4 –Valores estatísticos sobre o efeito da variação das probabilidades no operador genético de Recombinação

Recombinação	Mutação	Média	Desvio Padrão
PMX Recombination (0.1)	Insert Mutation (0.2)	150.9	10.94
PMX Recombination (0.2)	Insert Mutation (0.2)	147.5	13.23
PMX Recombination (0.3)	Insert Mutation (0.2)	148.15	9.95
PMX Recombination (0.4)	Insert Mutation (0.2)	146.3	11.82

PMX Recombination (0.5)	Insert Mutation (0.2)	141.85	9.54
PMX Recombination (0.6)	Insert Mutation (0.2)	140.25	8.10
PMX Recombination (0.7)	Insert Mutation (0.2)	142.55	8.40
PMX Recombination (0.8)	Insert Mutation (0.2)	138.6	9.31
PMX Recombination (0.9)	Insert Mutation (0.2)	138.1	8.43
Recombination 2 (0.1)	Insert Mutation (0.2)	154.6	11.85
Recombination 2 (0.2)	Insert Mutation (0.2)	155.55	13.64
Recombination 2 (0.3)	Insert Mutation (0.2)	162.5	13.13
Recombination 2 (0.4)	Insert Mutation (0.2)	165.3	11.35
Recombination 2 (0.5)	Insert Mutation (0.2)	177.35	11.62
Recombination 2 (0.6)	Insert Mutation (0.2)	191.45	10.97
Recombination 2 (0.7)	Insert Mutation (0.2)	218.5	9.27
Recombination 2 (0.8)	Insert Mutation (0.2)	221.95	8.50
Recombination 2 (0.9)	Insert Mutation (0.2)	226.6	8.77
Recombination 3 (0.1)	Insert Mutation (0.2)	153.9	15.72
Recombination 3 (0.2)	Insert Mutation (0.2)	157	11.14
Recombination 3 (0.3)	Insert Mutation (0.2)	157	11.19
Recombination 3 (0.4)	Insert Mutation (0.2)	165.5	9.91

Recombination 3 (0.5)	Insert Mutation (0.2)	169.7	8.37
Recombination 3 (0.6)	Insert Mutation (0.2)	182.6	8.06
Recombination 3 (0.7)	Insert Mutation (0.2)	185.25	9.06
Recombination 3 (0.8)	Insert Mutation (0.2)	187.6	9.91
Recombination 3 (0.9)	Insert Mutation (0.2)	187.6	8.61



Figura 4 -Gráfico de Barras sobre os Operadores de Recombinação vs Probabilidade de Recombinação

3.5 Efeito da variação das probabilidades dos operadores de mutação

Para a discussão destes resultados foram considerados os valores contantes para as variáveis da população (200), número de gerações (50) e *Tournament selection* (4).

Ao analisar os resultados presentes na tabela 5, é possível analisar que à medida que a taxa de mutação aumenta de 0.1 para 0.9, a média aumenta gradualmente de 141.8 para 175.65. Isso indica que uma probabilidade de mutação mais baixa resulta em soluções mais ótimas.

No caso da mutação 2, observou-se que as médias se mantêm constantes à medida que se aumenta a probabilidade de mutação. Isso indica que a taxa de mutação não tem um impacto significativo nos resultados obtidos para este operador.

No caso da mutação 3, é possível observar que a média aumenta consideravelmente à medida que se aumenta a probabilidade, variando entre 162.9 nos valores mais baixos para 225.5 nos

valores mais altos, indicando que uma probabilidade mais alta, leva a uma solução menos ótima.

Tabela 5 - Valores estatísticos sobre o efeito da variação das probabilidades no operador genético de Recombinação

Recombinação	Mutação	Média	Desvio Padrão
PMX Recombination (0.8)	Insert Mutation (0.1)	141.8	9.03
PMX Recombination (0.8)	Insert Mutation (0.2)	138.7	7.7
PMX Recombination (0.8)	Insert Mutation (0.3)	137.15	6.78
PMX Recombination (0.8)	Insert Mutation (0.4)	138.55	7.51
PMX Recombination (0.8)	Insert Mutation (0.5)	140.95	6.72
PMX Recombination (0.8)	Insert Mutation (0.6)	149.15	11.41
PMX Recombination (0.8)	Insert Mutation (0.7)	161.4	9.61
PMX Recombination (0.8)	Insert Mutation (0.8)	169.2	8.92
PMX Recombination (0.8)	Insert Mutation (0.9)	175.65	7.96
PMX Recombination (0.8)	Mutation 2 (0.1)	158.8	12.54
PMX Recombination (0.8)	Mutation 2 (0.2)	155.55	12.32
PMX Recombination (0.8)	Mutation 2 (0.3)	156.75	11.69
PMX Recombination (0.8)	Mutation 2 (0.4)	155.7	9.66
PMX Recombination (0.8)	Mutation 2 (0.5)	156.25	10.36
PMX Recombination (0.8)	Mutation 2 (0.6)	154.4	9.36
PMX Recombination (0.8)	Mutation 2 (0.7)	152.25	7.13

(0.8)			
PMX Recombination (0.8)	Mutation 2 (0.8)	153.65	7.44
PMX Recombination (0.8)	Mutation 2 (0.9)	154.6	6.45
PMX Recombination (0.8)	Mutation 3 (0.1)	162.9	10.99
PMX Recombination (0.8)	Mutation 3 (0.2)	166.3	11.63
PMX Recombination (0.8)	Mutation 3 (0.3)	166.5	15.16
PMX Recombination (0.8)	Mutation 3 (0.4)	162.65	12.14
PMX Recombination (0.8)	Mutation 3 (0.5)	179.35	14.97
PMX Recombination (0.8)	Mutation 3 (0.6)	210.85	8.39
PMX Recombination (0.8)	Mutation 3 (0.7)	218.3	6.94
PMX Recombination (0.8)	Mutation 3 (0.8)	219.45	9.68
PMX Recombination (0.8)	Mutation 3 (0.9)	225.5	7.49

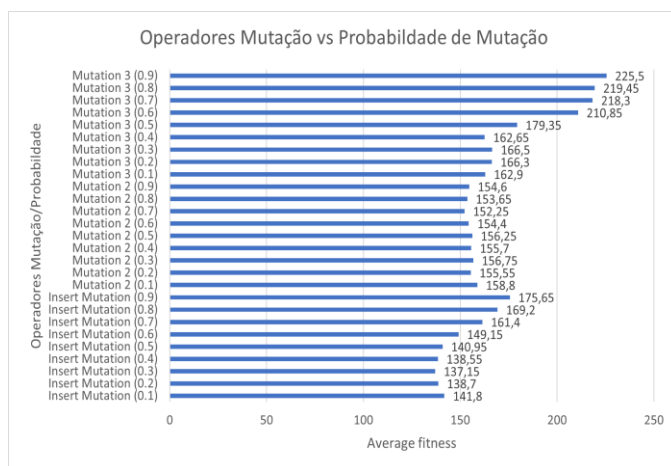


Figura 5 Gráfico de Barras sobre os Operadores de Mutação vs Probabilidade de Mutação

3.6 Efeito da variação do tamanho do torneio

Para a discussão destes resultados foram considerados os valores contantes para as variáveis da população (200), bem como o número de gerações (5), método de recombinação *PMX* (0.8) e a mutação *Insert* (0.2).

O método de seleção por torneio é um método de seleção onde um número fixo de indivíduos é selecionado aleatoriamente da população para competir entre si. O indivíduo com melhor aptidão dentro do torneio é escolhido como vencedor e selecionado para reprodução.

Ao analisar os resultados da tabela 6, verifica-se que à medida que o tamanho do torneio aumenta de 2 para 10, a média diminui gradualmente de 186.5 para 142.6. Isso indica que um tamanho de torneio maior resulta em soluções mais ótimas. Podemos também verificar que a melhor solução é alcançada com um tamanho do torneio 6, apresentando ainda um desvio padrão baixo, indicando que esta tipo de método apresenta resultados mais consistentes.

Portanto, analisando apenas o efeito da variação do torneio podemos concluir que escolher o *tournament selection* (6) pode ser a escolha adequada para obter soluções mais otimizadas.

Tabela 6 - Valores estatísticos sobre o efeito da variação do tamanho do torneio

Método seleção	Média	Desvio Padrão
<i>Tournament selection</i> (2)	186.5	10.72
<i>Tournament selection</i> (4)	140.25	6.73
<i>Tournament selection</i> (6)	139.8	7.87
<i>Tournament selection</i> (8)	141.35	9.27
<i>Tournament selection</i> (10)	142.6	11.96

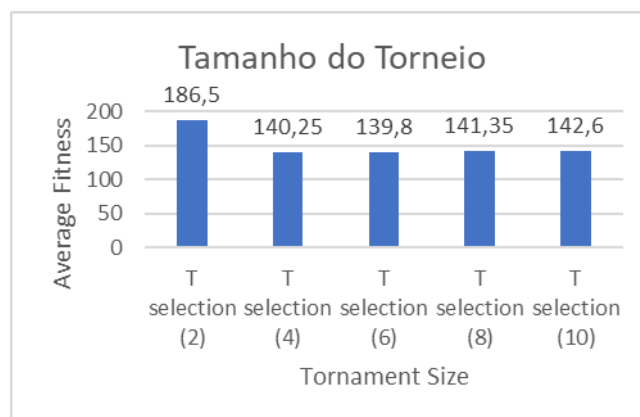


Figura 6 – Gráfico de barras sobre o tamanho do torneio

4. Conclusão

Com base nos resultados apresentados, foram investigados diversos aspetos relacionados com o desempenho do algoritmo genético na resolução do problema.

Populações intermédias (200) e menor número de gerações (30) resultaram em médias menores, indicando melhores soluções. Além disso, os operadores de recombinação e mutação também desempenharam papéis importantes. Sobre o efeito da variação da probabilidade da recombinação, o método de recombinação *PMX* com uma probabilidade de 0.9 e a mutação *Insert* com uma probabilidade de 0.2 demonstraram melhores soluções. Já sobre o efeito da variação da probabilidade da mutação, o método de recombinação *PMX* (0.8) e a mutação *Insert* (0.3) apresentou a melhor solução. Tendo em conta o método de seleção de torneio, os que apresentaram melhores resultados foram os de tamanho 4 e 6, com médias de 140.25 e 139.8, respetivamente. É importante ressaltar que para este problema, foi utilizado uma forma de penalização caso os *forklifts* se encontrassem na mesma célula, acrescentado mais 10 pontos ao *fitness* do indivíduo.

Em conclusão, foram investigados diversos aspetos relacionados com o desempenho do algoritmo genético na resolução do problema. Desta forma, verificou-se que o tamanho da população e o número de gerações têm um impacto significativo na qualidade das soluções encontradas. Além disso, observou-se que, diferentes operadores genéticos, como a recombinação e a mutação, podem influenciar a eficácia do algoritmo e, diferentes combinações de probabilidades, podem levar a resultados distintos. Relativamente ao tamanho do torneio, conclui-se que diferentes tamanhos também originam resultados diferentes.

No geral, este estudo forneceu uma visão abrangente sobre a aplicação de algoritmos genéticos, destacando a importância de ajustar os parâmetros e operadores genéticos face às características e objetivos do problema. Os resultados obtidos demonstraram a capacidade de encontrar soluções eficientes, mas também a necessidade de ajustar cuidadosamente os parâmetros para otimizar o desempenho.