Relatório - Projeto De Inteligência Artificial

Pedro Vicente
Estudante de Engenharia Informática
do Politécnico de Leiria
Estudante Nº2210716

João Franco
Estudante de Engenharia Informática
do Politécnico de Leiria
Estudante Nº2211083

RESUMO

Este relatório apresenta o desenvolvimento e implementação de um aplicativo para otimizar o processo de coleta em armazéns usando algoritmos genéticos e o algoritmo de busca A*. O objetivo do aplicativo é minimizar o tempo de entrega, a distância total percorrida e as colisões entre os agentes. O relatório fornece uma visão geral da descrição do problema, da solução implementada, da metodologia de avaliação e dos resultados obtidos a partir de vários experimentos. Além disso, discute os desafios enfrentados durante o projeto e sugere possíveis extensões futuras.

CATEGORIAS E DESCRITORES DE ASSUNTO

I.2.9: **[Inteligência Artificial]** - *Resolução de Problemas, Métodos de Controle e Busca*

I.2.8: [**Resolução de Problemas, Métodos de Controle e Busca**] - *Métodos Heurísticos*

TERMOS GERAIS

Algoritmos, Gestão, Medição, Desempenho, Design, Economia.

PALAVRAS-CHAVE

Coleta, Algoritmo de busca A*.

1. INTRODUÇÃO

A recolha eficiente de produtos em armazéns, conhecida como picking, é crucial para garantir entregas pontuais e reduzir custos operacionais. Este relatório descreve o desenvolvimento de um aplicativo que aborda o problema de otimização de picking usando uma combinação de algoritmos genéticos e o algoritmo de busca A*. O aplicativo permite que os usuários insiram instâncias do problema e visualize a simulação da melhor solução encontrada.

2. DESCRIÇÃO DO PROBLEMA

O problema envolve a distribuição de coletas (produtos) para agentes responsáveis por sua coleta. Cada agente deve coletar os produtos atribuídos em uma ordem específica para minimizar o tempo de entrega do último produto e reduzir a distância total percorrida. O armazém é representado como uma matriz, onde espaços vazios, prateleiras, produtos, pontos de entrega e agentes são denotados por diferentes valores. O objetivo é encontrar rotas ótimas para os agentes coletarem os produtos e entregá-los nos pontos de entrega designados, evitando colisões entre os agentes.

3. ABORDAGEM DA SOLUÇÃO

O problema envolve a distribuição de picks (produtos) para agentes responsáveis por sua coleta. Cada agente deve coletar os produtos atribuídos numa ordem específica para minimizar o tempo de entrega do último produto e reduzir a distância total percorrida. O armazém é representado como uma matriz, onde espaços vazios, prateleiras, produtos, pontos de entrega e agentes são representados por diferentes valores. O objetivo é encontrar rotas ótimas para os agentes recolherem os produtos e entregá-los nos pontos de entrega designados, evitando colisões entre os agentes.

4. IMPLEMENTATION DETAILS

A implementação do projeto envolveu diversos componentes-chave:

4.1 Classe WarehouseState

A classe WarehouseState foi desenvolvida para representar um estado do problema. Esta contêm as posições atuais dos agentes e produtos, bem como o estado da matriz do armazém.

4.2 Classe WarehouseProblemSearch

A classe WarehouseProblemSearch representa o problema para o algoritmo de busca A*. Esta define o estado inicial, a função de sucessor e a função de teste de objetivo.

4.3 Classe HeuristicWarehouse

A classe HeuristicWarehouse fornece uma heurística para o algoritmo de busca A*. Esta estima o custo de um estado dado até o estado objetivo, orientando a busca em direção a caminhos que lhe pareçam mais eficientes segundo a heurística aplicada.

4.4 Classe WarehouseProblemGA

A classe WarehouseProblemGA representa o problema para o algoritmo genético. Esta define a representação de indivíduos, a função de avaliação (fitness), a criação da população inicial e os operadores genéticos.

4.5 Classe WarehouseIndividual

A classe WarehouseIndividual representa um indivíduo do algoritmo genético. Esta armazena a distribuição de produtos pelos agentes e a ordem de recolha dos produtos.

4.6 Operadores de Recombinação e de Mutação

As classes Recombination2 e Recombination3 implementam operadores de recombinação específicos para o problema e a representação dos indivíduos. Por outro lado, as classes Mutation2

e Mutation3 implementam operadores de mutação também adequados para o problema e a representação utilizada.

5. DESCRIÇÃO DO ESTADO

O estado do problema é representado pela classe WarehouseState. Ela contém as posições atuais dos agentes, dos produtos e da saida, bem como o estado da matriz do armazém. A matriz do armazém representa a estrutura do armazém, as posições dos produtos, a posição do ponto de entrega e as posições iniciais dos agentes. Cada célula na matriz é atribuída a um valor: 0 para espaço vazio, 1 para uma prateleira vazia, 2 para uma prateleira com um produto, 3 para o ponto de entrega e 4 para um agente. Nesta classe foram também implementados os métodos que permitem saber em que direções o agente se pode deslocar e os métodos para deslocar o agente.

6. DESCRIÇÃO DA HEURÍSTICA

A heurística para o algoritmo de busca A* é implementada na classe HeuristicWarehouse. A heurística fornece uma estimativa do custo de um estado dado para o estado objetivo, orientando a busca para caminhos mais promissores. No contexto do problema de otimização de picking, a heurística que escolhemos foi a distância euclidiana implementada na classe HeuristicWarehouse.

7. DESCRIÇÃO DA REPRESENTAÇÃO DOS INDIVÍDUOS

Os indivíduos no algoritmo genético representam uma distribuição de produtos pelos agentes, bem como a ordem em que os agentes devem recolher os produtos atribuídos a eles. O genoma do individuo, é composto por números inteiros e únicos, que representam a ordem pela qual os agentes(forklifts) devem efetuar os picks, sendo que para separar que picks pertencem a cada agente, são usados números maiores do que o número de picks.

8. DESCRIÇÃO DA FUNÇÃO DE FITNESS

A função de fitness, é utilizada para determinar o quão bom é um indivíduo em termos de desempenho. No contexto do problema de otimização de picking, a função de avaliação calcula a distância total percorrida pelos agentes e o número de colisões entre eles. O objetivo é minimizar o tempo de entrega, a distância percorrida e as colisões. A função de fitness é implementada na classe WarehouseProblemGA e é usada para avaliar e comparar os indivíduos durante o processo de seleção e evolução do algoritmo genético. Para equilibrar a carga entre os agentes, a função de fitness, depois de calcular custo do caminho total e penalizar cada colisão com um custo de 100, multiplica o valor do fitness pelo custo do caminho do agente com maior custo.

9. DESCRIÇÃO DO MÉTODO DE CRIAÇÃO DA POPULAÇÃO INICIAL

A população inicial é criada aleatoriamente, onde cada indivíduo que gera a população inicial criando objetos WarehouseIndividual. A criação de um individuo está definida no construtor da IntVectorIndividual, superclasse da classe WarehouseIndividual que inicia o genoma a 0 e o percorre inserindo números de 1 ao número de produtos somado com o numero de agentes, sem repetições.

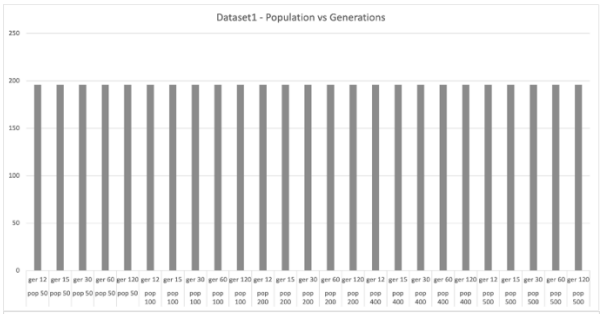
10. OPERADORES GENÉTICOS DESENVOLVIDOS

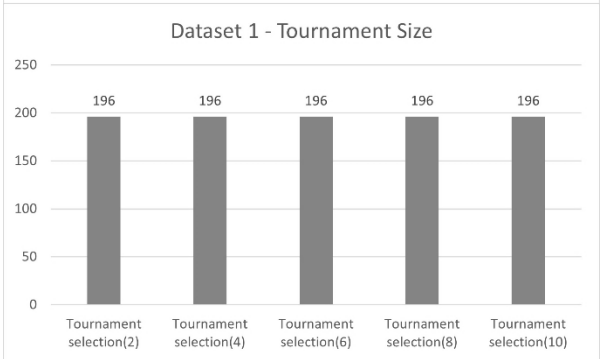
Os operadores genéticos são responsáveis pela variação e evolução dos indivíduos na população. Na nossa implementação, desenvolvemos dois operadores de recombinação, a recombinação de 2 cortes e a recombinação uniforme, adequados ao problema e à representação utilizada para os indivíduos. Estes operadores realizam a troca de produtos entre os agentes durante o crossover. Também desenvolvemos dois operadores de mutação, a mutação de troca (swap mutation) e mutação de deslocamento (Shift mutation), que fazem pequenas alterações nas distribuições de produtos dos indivíduos, trocando a ordem de produtos e a que agente pertence cada produto. A classe WarehouseProblemGA incorpora esses operadores genéticos para a evolução dos indivíduos.

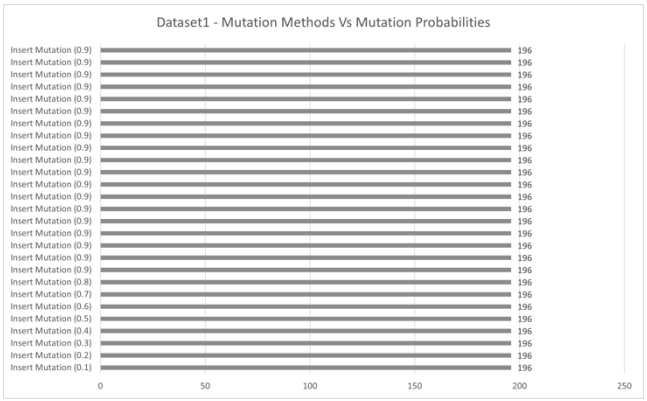
11. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Para avaliar o desempenho da solução implementada, realizamos uma série de testes para estudar diversos aspetos, incluindo o efeito do tamanho da população e do número de gerações, o desempenho de diferentes operadores genéticos, o impacto da variação das probabilidades dos operadores genéticos e a influência do tamanho do torneio. Os resultados obtidos a partir desses experimentos foram analisados e apresentados sob a forma de gráficos. Por motivos que não compreendemos, as experiências relativas á mutação e á recombinação não correram como o esperado e apenas usaram os métodos já implementados no projeto base.

11.1 Dataset 1

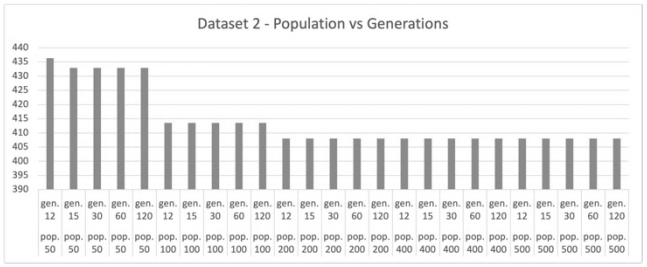


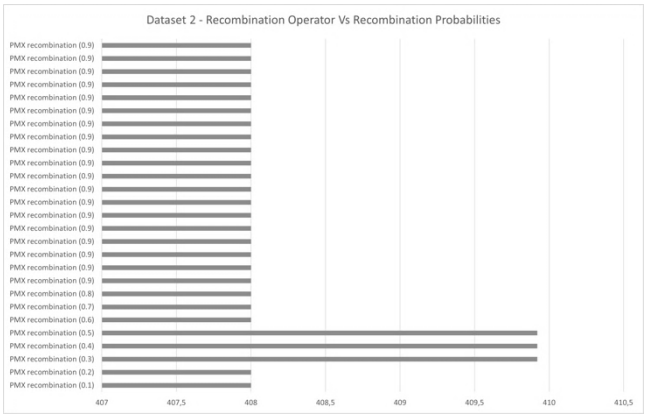


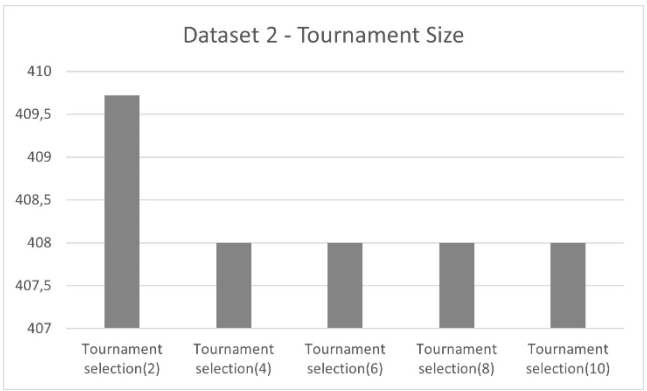


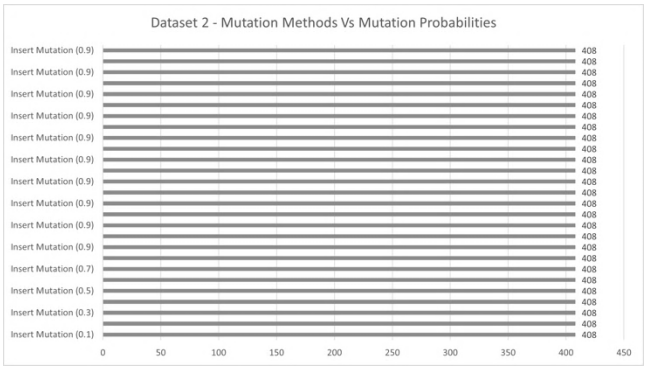
Devido á simplicidade do problema 1, o tamanho da população, o número de gerações, os métodos de recombinação, os métodos de mutação e respetivas probabilidades não influenciam.

11.2 Dataset 2



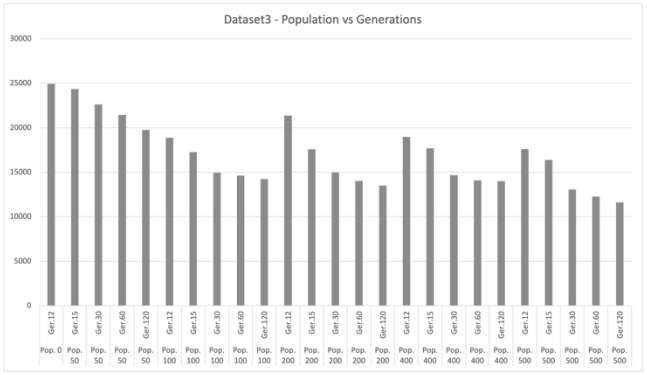


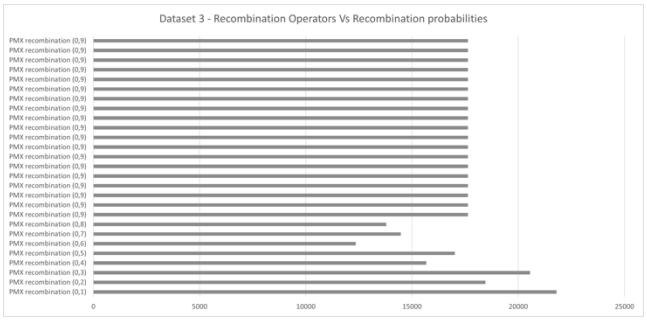


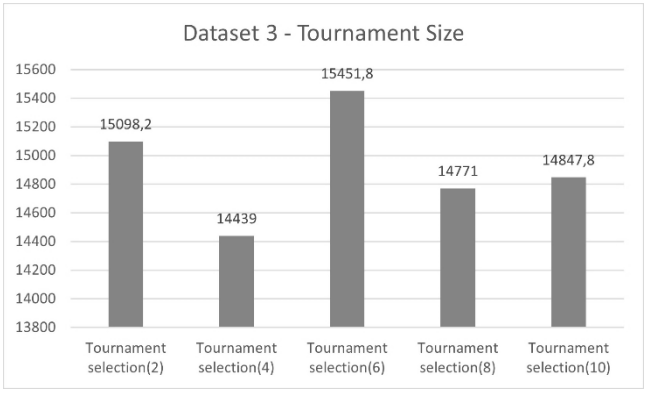


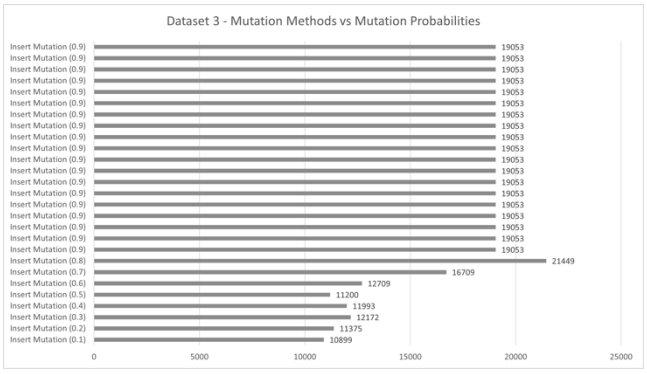
Este dataset revelou ser mais influenciável pelo tamanho da população, pela quantidade de gerações e pelo tamanho do sorteio sendo que a quantidade de gerações tem pouca influência no fitness.

11.3 Dataset 3



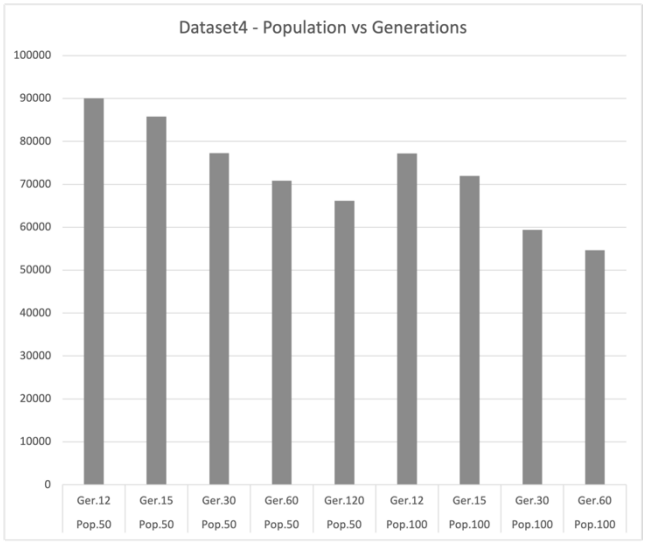


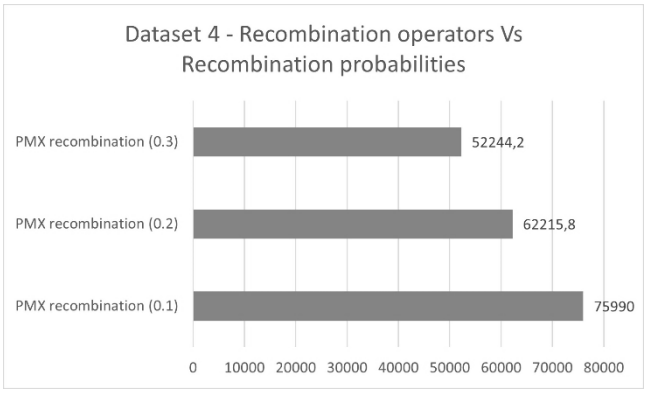


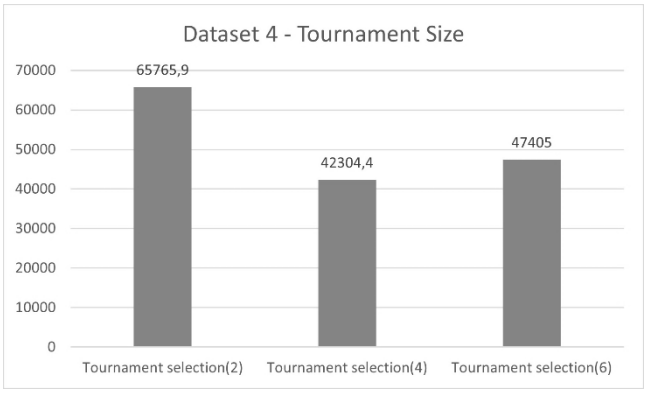


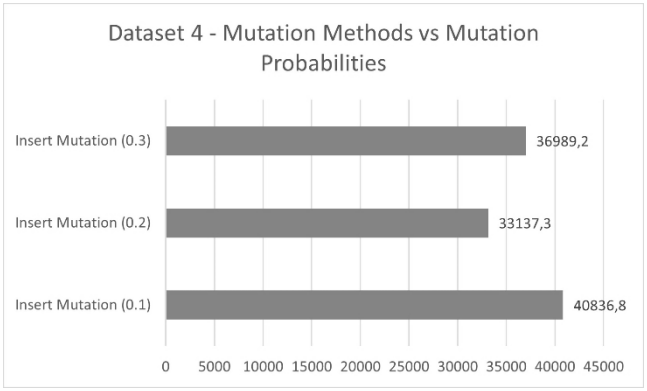
Este dataset revelou ser mais influenciável pelo tamanho da população e pela quantidade de gerações sendo que quanto maior a população e quanto mais gerações existirem, melhores serão os resultados finais do fitness. O tamanho do torneio também tem uma influência positiva no fitness, apesar de o tamanho de torneio com melhores resultados ser 4.

11.4 Dataset4









Devido ao nosso poder de processamento limitado, com o tempo disponível, não nos foi possível terminar o dataset 4, mas segundo os dados recolhidos, a melhor probabilidade de mutação foi 0.2 e quanto maior for a população, o número de gerações e a probabilidade de recombinação, melhor será o fitness.

12. EXTRAS IMPLEMENTADOS NO PROJETO

Foram implementados dois extras no Projeto sendo eles a deteção de colisões, havendo uma penalização de 100 por colisão, e a implementação de barras de progresso e de percentagem no titulo da janela, de forma a dar ao utilizador do programa um feedback do estado dos cálculos efetuados.

13. CONCLUSÃO

Em conclusão, este relatório apresentou o desenvolvimento e implementação de uma aplicação para otimização de picking em armazéns usando algoritmos genéticos e o algoritmo de busca A*. A solução proposta mostrou-se eficaz na minimização do tempo de entrega, da distância percorrida e das colisões entre os agentes. No entanto, existem oportunidades para melhorias e extensões futuras, como a incorporação de outros algoritmos de busca ou a consideração de restrições adicionais. Em geral, o projeto proporcionou uma compreensão aprofundada do problema de otimização de picking e das técnicas aplicadas para sua resolução.

14. AGRADECIMENTOS

Agradecemos ao Professor Carlos Grilo pela orientação e apoio durante o desenvolvimento deste projeto.

15. REFERÊNCIAS

[1] Russell, Stuart, and Peter Norvig. Artificial Intelligence: A Modern Approach. 3rd ed., Pearson, 2016.

[2] Russell, Stuart, and Peter Norvig. Artificial Intelligence: A Modern Approach. 4th ed., Pearson, 2021.