Logotipo

Descrição gerada automaticamente

Licenciatura em Engenharia Informática (BLEI)

Unidade Curricular de Inteligência Artificial

3.º Ano

**AGENTE DE PESQUISA APLICADA**

**À DETERMINAÇÃO** **DO MELHOR TRAJETO DE UM ROBÔ**

Docente Responsável

Prof. Dr. António Castro

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Trabalho elaborado por:

**João Madail, Joaquim Ferreira e Lara Figueiredo**

Vila Nova de Gaia, 23 de Novembro de 2022

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SUMÁRIO**

Um agente inteligente, é aquele que toma a melhor ação possível face um problema. Definir a melhor rota numa viagem é um excelente exemplo de um problema cujo recurso a agentes inteligentes é uma otima solução. No presente relatório, apresentamos um estudo comparativo entre 4 algoritmos de pesquisa resultado do seguinte objetivo. Determinar o melhor percurso a realizar por um robô, que se movimenta num centro de distribuição, onde deve recolher encomendas, procurando minimizar a distância percorrida. Os algortimos implementados são o A\*, Dijkstra, BFS (Breadth First Search) E DFS (Depth First Search), utilizados como sendo 4 soluções distintas. Como condicionantes, robô em questão tem uma capacidade limitada, sendo que, caso necessário, o mesmo poderá ter que efetuar um desvio para largar encomendas num ponto de recolha, assim, possibilitando a recolha de todas as encomendas. Como estrutura de dados, recorreu-se ao uso de Nós. Para cada simulação, são guardados o número de nós visitados, o custo de solução e o tempo de execução, sendo os dados agrupados por Algoritmo. Este processo é repetido 3 vezes, cada uma para um ambiente diferente. De modo a facilitar a criação e seleção de ambiente, é possível gravar e importar novos ambientes.

**ÍNDICE**

[**1.** **INTRODUÇÃO** 5](#_Toc120369303)

[**2.** **ESTADO DA ARTE** 5](#_Toc120369304)

[**2.1** **Algoritmos** 5](#_Toc120369305)

[**2.2 Problemas complexos resolvidos com IA** 6](#_Toc120369306)

[**2.3 Algoritmos de Otimização** 6](#_Toc120369307)

[**2.3. 1 Algoritmo Astar (A\*)** 6](#_Toc120369308)

[**2.3.2 Algoritmo Dijkstra** 6](#_Toc120369309)

[**2.3.3 DFS (Depth First Search)** 7](#_Toc120369310)

[**2.3.4 BFS (Breadth First Search)** 7](#_Toc120369311)

[**3.** **DESCRIÇÃO DO PROJETO** 9](#_Toc120369312)

[**3.1 Descrição do Problema** 9](#_Toc120369313)

[**3.2 Descrição da Resolução do Problema** 10](#_Toc120369314)

[**4.** **RESULTADOS** 10](#_Toc120369315)

[**4.1** **Estudo Comparativo** 13](#_Toc120369316)

[**5.** **CONCLUSÕES** 18](#_Toc120369317)

[**6.** **REFERÊNCIAS BIBLIOGRÁFICAS** 19](#_Toc120369318)

**ÍNDICE DE FIGURAS**

[Figura 1 – Algoritmo BFS 7](#_Toc120369286)

[Figura 2 – Demonstração de um trecho de código 10](#_Toc120369287)

[Figura 3 – Imput para o Gride 11](#_Toc120369288)

[Figura 4 – Número de encomenda e escolha do algoritmo 11](#_Toc120369289)

[Figura 5 – Desenhar armazém e estatísitca com resultados obtidos 12](#_Toc120369290)

[Figura 6 - Gride obtido após inserção de dados 12](#_Toc120369291)

[Figura 7 - Primeiro Ambiente 13](#_Toc120369292)

[Figura 8 - Segundo Ambiente 14](#_Toc120369293)

[Figura 9 - Terceiro Ambiente 14](#_Toc120369294)

[Figura 10 - Desempenho após três cenários 15](file:///C:\Users\joaom\Downloads\RelatórioAI%20Final%20(1).docx#_Toc120369295)

[Figura 11 - Tempo de execução para os três cenários 16](file:///C:\Users\joaom\Downloads\RelatórioAI%20Final%20(1).docx#_Toc120369296)

[Figura 12 - Custo para o cenário 1 16](#_Toc120369297)

[Figura 13 - Custo para o cenário 2 17](#_Toc120369298)

[Figura 14 - Custo para o cenário 3 17](#_Toc120369299)

# **INTRODUÇÃO**

Os conceitos e ferramentas de inteligência artificial (IA) vêm conduzindo mudanças em nossa sociedade. A IA atualmente compreende diferentes áreas que incluem aprendizagem de máquina, visão computacional, processamento de linguagem natural, reconhecimento de padrões em imagens, robótica, entre outras. Os avanços recentes em IA têm viabilizado a criação e o aperfeiçoamento de aplicações que vão desde veículos autônomos, diagnóstico médico, assistência física a deficientes e idosos, à segurança pública e indústria de entretenimento (Coneglian, 2020).

As técnicas de IA, associadas à abundante quantidade de dados digitais e ao omnipresente poder de processamento paralelo entregue pela computação na nuvem, deverão, sem dúvida, suprir a alta demanda pública por serviços que podem ser otimizados, utilizando IA, por exemplo, havendo uma demanda por um serviço de limpeza, pode-se usar IA para criar um Robô aspirador inteligente (Santos, 2017).

Partindo deste pressuposto, no âmbito da disciplina de Inteligência Artificial foi proposto a realização de um projeto com o tema Agente de Pesquisa aplicada à determinação do Trajeto de um Robô. De acordo com a definição adotada pela Robotic Industries Association (RIA), o robô é um equipamento multifuncional e reprogramável, projetado para movimentar materiais, peças, ferramentas ou dispositivos especializados através de movimentos programados para a execução de uma infinidade de tarefas (Ziebarth *et.al*, 2020).

Portanto, pretende-se neste estudo desenvolver um Agente que seja capaz de para determinar o melhor percurso a realizar por um robô, que se movimenta num centro de distribuição, onde deve recolher encomendas, procurando minimizar a distância percorrida.

# **ESTADO DA ARTE**

## **Algoritmos**

Algoritmo pode ser definido como uma linguagem intermediária entre a linguagem humana e as linguagens de programação. É utilizado para representar a solução de um problema. Descrevem instruções a serem executadas pelos computadores (Zanatto, 2021).

## **2.2 Problemas complexos resolvidos com IA**

A inteligência artificial é desenvolvida para que os dispositivos criados pelo homem possam desempenhar determinadas funções sem a interferência humana. Isso implica em perceber variáveis, tomar decisões e [resolver problemas](https://fia.com.br/blog/resolucao-de-problemas-nas-empresas/" \t "_blank). Enfim, operar em uma lógica que remete ao raciocínio (Gebert *et al*., 2018)

## **2.3 Algoritmos de Otimização**

### **2.3. 1 Algoritmo Astar (A\*)**

É um algoritmo de busca de caminho que encontra sempre o melhor caminho entre dois pontos, que pode ser o caminho de menor distância, menor gasto energético, menor tempo de execução, entre outros. Mas tem a contrapartida de ser computacionalmente pesado, sendo necessário ter em consideração vários aspetos na sua implementação tais como o poder de cálculo do processador, a limitação de memória e o tipo de heurística a utilizar (Pedro, 2012).

### **2.3.2 Algoritmo Dijkstra**

Criado por Edsger Dijkstra em 1959, o algoritmo de Dijkstra é um dos principais algoritmos utilizados para resolver problemas de caminhos em grafos. Sendo apenas possível ser aplicado em grafos ponderados em que as arestas são positivas, um grafo é chamado ponderado se todas as suas arestas são representadas por um peso. Este algoritmo tem como objetivo encontrar o caminho mais econômico entre tais dois vértices (Baltazar; Pereira, 2018). No algoritmo de Dijkstra enquanto houver algum vértice para o qual a distância mínima ainda não foi calculada, é escolhido dentre os disponíveis um vértice em que a distância calculada até o momento é mínima.

### **2.3.3 DFS (Depth First Search)**

A ideia da DFS é a partir de um vértice inicial, escolher uma aresta e a partir dela percorrer o mais "fundo"possível todos os vértices acessíveis antes de verificar a próxima aresta (Goldbarg, 2017).O algoritmo inicia atribuindo como não visitado a todos os vértices do grafo, a seguir é escolhido como início da busca o vértice inicial do grafo e então é executada a função DFS responsável pela busca. A função DFS recebe o vértice e o marca como visitado. Após isso, executa para todos os vértices não visitados e adjacentes a ele o método DFS (Feofiloff, 2018*).*

### **2.3.4 BFS (Breadth First Search)**

No algoritmo de busca em amplitude é realizada a busca de todos os elementos adjacentes ao vértice inicial antes de buscar os outros elementos. (Goldbarg, 2017). A busca pode ser representada como uma busca em níveis, no grafo da Figura 1 pode ser vista a ordem em que os vértices são visitados, primeiramente são percorridos os vértices dentro do quadrado vermelho, após isso são encontrados os vértices do quadrado laranja e por fim os vértices do quadrado azul são marcados como visitado.

Diagrama, Esquemático

Descrição gerada automaticamente

Figura 1 – Algoritmo BFS

# **DESCRIÇÃO DO PROJETO**

Num centro de distribuição, existe um robô responsável pela recolha de encomendas, que se encontram espalhadas pelo centro, que são depois colocadas num espaço de recolha. O robô possui capacidade limitada, pelo que pode ter de efetuar mais que uma viagem até ao espaço de recolha, para colocar todas as encomendas. O robô possui dimensão e pode andar em frente ou rodar, parado, em qualquer direção. O espaço onde o robô se movimenta (centro) é conhecido e contém obstáculos (de posição também conhecida), que o robô terá de contornar. A localização das encomendas a recolher é conhecida.

A localização do espaço de recolha também é conhecida. Deve ser possível especificar o espaço onde o robô se movimenta (posição dos obstáculos), a posição de todas as encomendas a recolher, a posição do espaço de recolha, a posição inicial do robô e o número de encomendas que pode recolher ao mesmo tempo (capacidade).

## **3.1 Descrição do Problema**

É necessário utilizar pelo menos dois métodos de pesquisa (um deles o A\* e os outros à escolha) para encontrar o melhor percurso a realizar pelo robô. Neste caso, além do algoritmo já pré-definido escolheu-se os seguintes algoritmos: Dijkstra,DFS, BFS a fim de realizar um estudo comparativo entre os quatro algoritmos e descobrir o algoritmo mais eficiente para resolver o problema.

## **3.2 Descrição da Resolução do Problema**

O projeto está estruturado em três classes principais, sendo uma delas o main, no qual estão todos os métodos principais de execução dos algoritmos.

Texto

Descrição gerada automaticamente

Figura 2 – Demonstração de um trecho de código

# **RESULTADOS**

O projeto foi desenvolvido utilizando a linguagem Phyton, pois possibilita a criação de modelos de Inteligência Artificial. Também conta com um enorme número de bibliotecas para IA, tanto para algoritmos de machine learning, deep learning ou até mesmo visão computacional. Foi necessário a criação de um repositório para acesso remoto do grupo ao código desenvolvido, portanto foi escolhido o GitHub pois todos os elementos do grupo já tinham conhecimentos prévios a cerca da ferramenta.

O projeto possui uma interface gráfica que permite ao utilizador configurar os seguintes parâmetros: largura, comprimento e tamanho da célula para definir o gride.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 3 – Imput para o Gride

Também é possível determinar o números de encomendas a recolher, escolher o algoritmo para ser utilizado, posicionar o robô, as encomendas, inserir obstáculos, determinar ponto de entrega.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 4 – Número de encomenda e escolha do algoritmo

Após a inserção dos dados, é possível desenhar o armazém com os todos os atributos definidos ou também inserir armazém com dados já pré-definidos. O resultado final gera a estatística com número de nós visitados, custo da solução e tempo de execução do algoritmo.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Figura 5 – Desenhar armazém e estatísitca com resultados obtidos

Para ilustrar os resultados obtidos, é necessário executar cada algoritmo separadamente. Os dados são exportados para um arquivo de formato csv.

Dado os parâmetros: largura: 2, comprimento: 20, tamanho da célula: 25, número de encomendas: 2. É possível obter o seguinte gride

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

Figura 6 - Gride obtido após inserção de dados

## **Estudo Comparativo**

A fim de testar o impacto dos obstáculos, da capacidade das encomendas e do número de encomendas a recolher na definição da melhor solução, criaram-se 3 ambientes diferentes.

O primeiro ambiente, definido por número total de encomendas de 5, uma capacidade de 2 encomendas e uma pequena parede/obstáculo, foi criado com o propósito de testar a influência do número total de encomendas.

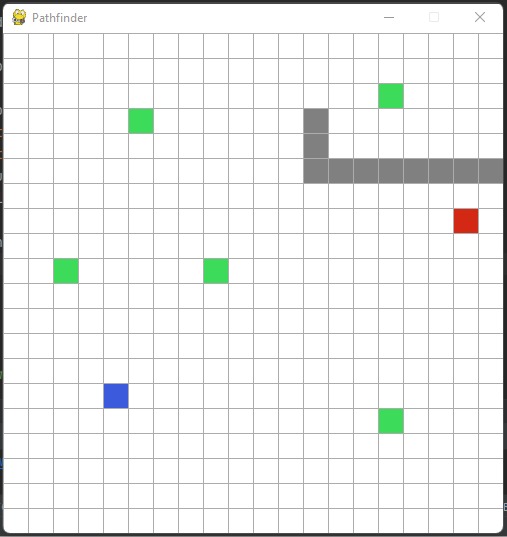


Figura 7 - Primeiro Ambiente

No segundo ambiente, aumentámos a capacidade para 3, o número de encomendas e tabém o tamanho da parade/obstaculo com o intuito de testar a influência dos obstáculos

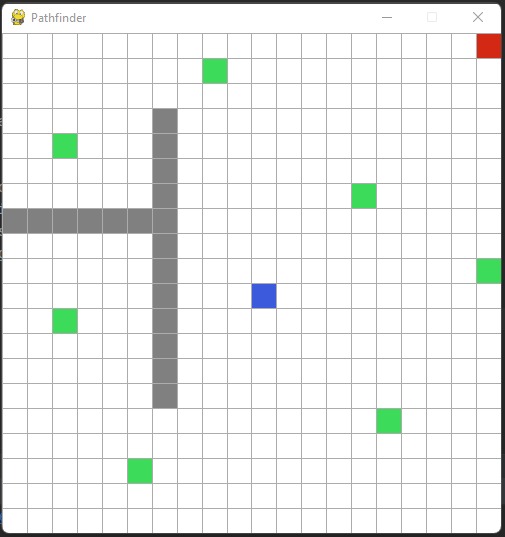


Figura 8 - Segundo Ambiente

No terceiro ambiente, criado com o objetivo de testar a influência da capacidade, reduziu-se a capacidade para 1, obrigando o robo a deslocar-se diretamente para o ponto de recolha assim que recolhia cada encomenda.

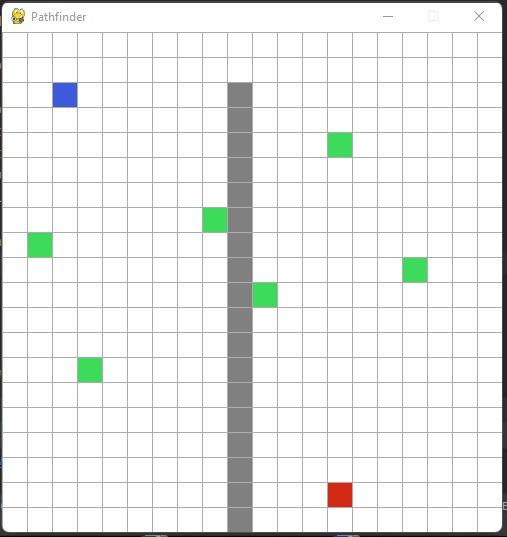


Figura 9 - Terceiro Ambiente

No primeiro ambiente o algoritmo A\* demorou 12.152261 segundos (em média 65.656% mais rápido que os restantes algoritmos), no segundo 34.733131 segundos (em média 42.950% mais rápido que os restantes algoritmos) e no terceiro 40.745206 segundos (em media 34,779% mais rápido que os restantes algoritmos).

Figura 10 - Desempenho após três cenários

Foi possível concluir que, número total de encomendas pouco influenciou o tempo de execução do A\*, contudo, condicionando a procura da solução ideal com a redução da capacidade, ou aumentando a quantidade de obstáculos, foi possível registar uma perda de performance.

Embora fosse consideravelmente mais rápido que os restantes algoritmos em todos os ambientes, o A\* demonstrou que à medida de que a necessidade de aumentar os nós visitados vai aumentando, a disparidade entre o seu tempo de execução e o dos restantes vai diminuindo. É possível fazer esta análise observando os gráficos abaixo.

Figura 11 - Tempo de execução para os três cenários

~

Figura 12 - Custo para o cenário 1

Figura 13 - Custo para o cenário 2

Figura 14 - Custo para o cenário 3

# **CONCLUSÕES**

Neste trabalho foi apresentada uma análise comparativa da implementação dos quatro algoritmos de busca no qual o objetivo foi encontrar o melhor percurso a realizar pelo robô. De um modo geral após análises de alguns cenários, foi possível observar que o A\* e o Dijkstra demonstraram desempenho semelhantes relativamente ao custo da solução, porém o A\* tem um menor tempo de execução. Nesta mesma análise, foi possível concluir que, para a problemática em questão, algoritmos como o DFS e BFS não são as melhores opções dado a complexidade temporal por de trás das grandes tempos de execução.

Em termos da implementação de código do projeto, verificamos que pode-se melhorar a aplicação no futuro, implementando a hipótese do robot quando vai buscar uma encomenda, caso encontre outra encomenda no seu caminho, recolhe esta e só depois retorna a que previamente ia buscar.

# **REFERÊNCIAS BIBLIOGRÁFICAS**

Apostila de Lógica de Programação. Instituto Federal Fluminense Campus Centro. Maio de 2019.

J. Bughin, E. Hazan, Five Management Strategies for Getting the Most from AI, McKinsey.com and MIT Sloan Management Review, 2017 (https: //www.mckinsey. com/mgi/overview/in-the-news/five-management- -strategies-for-getting-the-most-from-ai).

Mohamed, H. A self learner’s guide to shortest path algorithms, with implementations in Python. Disponível em <https://towardsdatascience.com/a-self-learners-guide-to-shortest-path-algorithms-with-implementations-in-python-a084f60f43dc> Acesso em 22 de Novembro de 2022.

P. Gerbert, S. Ramachandran, J-H. Mohr, M. Spira, The big leap toward AI at scale, BCG Henderson Institute, 2018 (https://www.bcg.com/de-de/publications/ 2018/ big-leap-toward-ai-scale.aspx).

Pedro Luís Cerqueira Gomes da Costa et al. Planeamento cooperativo de tarefas e trajectórias em múltiplos robôs. 2012.

Saraiva, C. Recuperação da informação com abordagem semântica utilizando linguagem natural: a inteligência artificial na ciência da informação. Trabalho de Conclusão de Curso. UNESP, 2020.

Zanatto, N.E. GRAPHEDU: Sistema web para visualização de algoritmos de grafos. Trabalho de Conclusão de Curso. Universidade Caxias do Sul, 2021.

Ziebarth, J. A; Novaes, S. F.; Nonato, L.G. Rumo a uma política de Estado para inteligência artificial. p 37-48. 2020 (https://www.revistas.usp.br/revusp/article/view/167914).