

Inteligência Artificial

Docente Joaquim Gonçalves

Licenciatura em Engenharia Informática Médica

2023/2024

**Algoritmos de Pesquisa**

*Profundidade, largura e A\**

23544 - Ana Beatriz Machado Carvalho

23548 – Ana Margarida Maia Pinto

23552 - Diana Alexandra da Costa Dinis

Índice

[1. Introdução 1](#_Toc151729877)

[1.1. Enquadramento 1](#_Toc151729878)

[1.2. Estrutura do documento 1](#_Toc151729879)

[2. Contexto 2](#_Toc151729880)

[2.1. Estratégia 2](#_Toc151729881)

[3. Algoritmos de Pesquisa 3](#_Toc151729882)

[3.1. Largura 3](#_Toc151729883)

[3.2. Profundidade 3](#_Toc151729884)

[3.3. A\* 4](#_Toc151729885)

[3.3.1. Heurística 4](#_Toc151729886)

[4. Testes e Resultados 5](#_Toc151729887)

[5. Conclusão 6](#_Toc151729888)

[6. Bibliografia 7](#_Toc151729889)

**Índice de Figuras**

[Figura 1 - Representação gráfica do algoritmo de pesquisa em largura 3](#_Toc151729890)

[Figura 2 - Representação gráfica do algoritmo de pesquisa em profundidade 4](#_Toc151729891)

**Índice de Tabelas**

**Não foi encontrada nenhuma entrada do índice de ilustrações.**

**Lista de siglas e acrónimos**

**FIFO** *First In First Out*

**FILO** *First In Last Out*

**IA** Inteligência Artificial

# Introdução

## Enquadramento

Este trabalho enquadra-se da unidade curricular de Inteligência Artificial lecionada pelo docente Joaquim Gonçalves da Licenciatura de Engenharia Informática Médica, no Instituto Politécnico do Cávado e do Ave

## Estrutura do documento

Este documento encontra-se dividido em essencialmente cinco capítulos. Inicia com o contexto do trabalho, onde estará exposto os objetivos do trabalho. Segue-se com o capítulo dos algoritmos de pesquisa, onde estará uma introdução teórica a cada um dos algoritmos escolhidos. A seguir, o capítulo dos testes e resultados, onde os algoritmos serão aplicados ao problema em Python e, por fim, a conclusão, onde os resultados serão comparados e será escolhido o melhor algoritmo para cada situação.

# Contexto

Este capítulo servirá essencialmente para contextualizar o trabalho prático, com a definição dos objetivos deste.

O trabalho terá como plano um armazém automático, onde um *robot* irá receber um pedido de encomenda de um item e terá de calcular uma rota desde a sua posição até ao item encomendado, que estará disponível numa determinada coordenada.

A fase inicial do trabalho será a implementação de um único *robot* com um único objetivo.

A fase seguinte será a implementação de um único *robot* com dois objetivos.

## Estratégia

Sendo um problema de pesquisa, irão ser aplicados os algoritmos de pesquisa em largura e profundidade para os não informados e o algoritmo A\* para o algoritmo informado.

Terá de ser implementado, em *Python*, uma representação do armazém, assim como os algoritmos que o *robot* terá de modo a conseguir uma solução ao problema. Serão avaliados o tempo que o algoritmo demora a encontrar uma solução e se essa solução será a solução ótima.

A solução será o caminho encontrado pelos algoritmos desde a posição inicial até ao objetivo, ou seja, até ao item no armazém. Uma solução pode ser considerada a solução ótima quando esta for o menor caminho, com o menor custo.

Os algoritmos de pesquisa, de modo a conseguirem prosseguir com a procura, utilizam uma *frontier*, que irá gerir os nós a expandir, de modo que o algoritmo saiba as opções que tem para prosseguir com a procura.

A forma como a *frontier* é montada depende do algoritmo.

# Algoritmos de Pesquisa

Este capítulo tratará da introdução teórica a cada algoritmo escolhido para o desenvolvimento do trabalho prático.

## Largura

O algoritmo de pesquisa em largura irá seguir múltiplas direções ao mesmo tempo, explorando vários caminhos simultaneamente. Para tornar isso possível, a *frontier* será montada como uma *queue*, tendo como método o FIFO – *First In First Out*.

Assim, todos os novos caminhos possíveis são adicionados à fila e expandidos conforme a ordem que foram adicionados. Montando uma árvore, visível na Figura 1, observa-se que se expande todos os nós do mesmo nível antes de passar para outro nível:

Uma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamente

Figura 1 - Representação gráfica do algoritmo de pesquisa em largura

## Profundidade

O algoritmo de pesquisa em profundidade é o oposto do algoritmo de pesquisa em largura. Enquanto o algoritmo em largura exausta todos os nós a expandir do mesmo nível, seguindo vários caminhos ao mesmo tempo, o algoritmo em profundidade segue apenas um caminho até ao fim, representado na Figura 2.

Uma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamenteUma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamente

Figura 2 - Representação gráfica do algoritmo de pesquisa em profundidade

Para isso, a estratégia deste algoritmo é programada para agir como uma *stack*, agindo em FILO – *First In Last Out*. Com isso, ao expandir os nós, o último a ser adicionado à *frontier* irá ser o próximo a ser expandido.

## A\*

### Heurística

# Testes e Resultados

# Conclusão

# Bibliografia

<https://www.geeksforgeeks.org/depth-first-traversal-dfs-on-a-2d-array/>

<https://www.baeldung.com/cs/dfs-vs-bfs-vs-dijkstra>

<https://www.pygame.org/tags/algorithm>

https://www.pygame.org/docs/ref/time.html

<https://cs50.harvard.edu/ai/2023/notes/0/>

https://www.techwithtim.net/tutorials/game-development-with-python/snake-pygame/snake-tutorial-1

https://www.inf.ed.ac.uk/teaching/courses/inf2d/timetable/05\_SearchB\_wbg\_4up.pdf

https://en.wikipedia.org/wiki/Breadth-first\_search