

# Introdução à Inteligência Artificial

## Trabalho 2

Professor Alneu de Andrade Lopes  
PAE Nicolás Roque dos Santos  
PAE Paulo Ricardo Viviurka do Carmo

**Data de Entrega 29/06**

### 1. Descrição

Este trabalho de implementação tem como objetivo testar o funcionamento dos diferentes algoritmos de busca aprendidos (busca em largura, busca em profundidade, busca best first, bem como os algoritmos A e A\*) no curso. Para a análise, devem ser construídos grafos-knn com diferentes parâmetros, conforme especificados abaixo, para simular mapas com diferentes tamanhos, números de vértices e número de arestas entre vértices, nos quais os algoritmos serão testados. Além da implementação, os tempos de execução e caminhos gerados devem ser guardados de forma a disponibilizar dados para as comparações entre os algoritmos nos diferentes cenários. Observe que as médias de várias execuções com diferentes parâmetros deverão ser usadas.

Visando criar um ambiente propício de testes, os grafos-knn foram escolhidos pois possibilitam a geração de diferentes configurações e sua topologia final é semelhante a mapas, ou seja, para alcançar longas distâncias são utilizados os nós vizinhos. Testes poderão ser realizados com outros tipos de grafos com regras geradoras (como as redes aleatórias) terão avaliação extra, devido ao diferencial da análise do comportamento dos algoritmos em diferentes topologias.

Os grafos-knn são gerados de acordo com o procedimento indicado abaixo:

```
procedimento grafo_knn( $v$ ,  $k$ )  
  lista_vertices <-  $v$  vértices com coordenadas ( $x$ ,  $y$ ) randômicas;  
  lista_arestas <- cada vértice  $v$  recebe  $k$  arestas conectadas aos vértices  
    com as menores distâncias euclidianas ( $v1$ ,  $v2$ , distância);  
fim
```

As buscas devem ser realizadas de forma a encontrar um **caminho** entre um ponto **inicial** e um **final**. Ambos devem ser definidos previamente a cada cenário onde os algoritmos de busca devem ser aplicados, de forma a manter o ambiente de testes constante. Os diferentes cenários de implementação correspondem a utilização das seguintes combinações de parâmetros de formação dos grafos-knn:

- Número de vértices ( $v$ ): Por exemplo, usar os conjuntos de 500, 5000 e 10000 para a análise.
- Valor do  $K$ , que afeta o número de arestas total, ( $k$ ): 3, 5 e 7 (pelo menos)

O tamanho dos eixos  $x$  e  $y$  deve ser correspondente ao número de vértices  $v$ , por exemplo: se  $v = 5000$ , então  $(x, y) = (5000, 5000)$ , aumentando as possibilidades de distribuição dos vértices de acordo com o tamanho do grafo.

Por exemplo, no cenário com  $v=7000$  e  $k=3$  serão enumerados os parâmetros e retornos necessários para os demais requerimentos do trabalho:

- Para gerar o grafo-knn:
  - $v = 7000$ ;  $k = 3$ ;  $x = 7000$ ;  $y = 7000$ ;
  - retorno: grafo-knn para a execução do cenário.
- Para a execução dos algoritmos:
  - ponto de *início* e *fim* para os algoritmos buscarem um caminho;
  - retornos:
    - caminho gerado por cada algoritmo;
    - tempos de execução e tamanhos dos caminhos de cada algoritmo.
- Gerar visualizações para a análise posterior dos algoritmos.

**Estes passos devem ser realizados para todas as combinações de  $v$  e  $k$ .**

## 2. Informações

- Este trabalho deverá ser desenvolvido em um grupo composto por 2 pessoas.
- A implementação deverá ser realizada preferencialmente em Python.
- Além do código-fonte, um relatório de até 10 páginas deverá ser elaborado. O relatório deverá ter as seguintes informações: introdução, descrição da implementação, resultados contendo visualizações dos caminhos e do tempo de execução de cada algoritmo, e uma discussão sobre os resultados.
- Após a entrega do código e do relatório, cada grupo deverá apresentar o trabalho desenvolvido em uma das datas presentes na Seção 4.
- **Trabalhos plagiados receberão nota 0.**

## 3. Entrega

Um dos integrantes do grupo deverá criar uma pasta com o nome “T2” em seu escaninho no Tidia e inserir o seguinte arquivo na pasta:

- Arquivo compactado (.rar, .zip ou .tar.gz) contendo:
  - Código-fonte e demais arquivos utilizados na implementação.
    - **Inserir informações sobre a execução do código.**
  - Relatório em formato pdf contendo as informações definidas na Seção 2.
    - **Relatórios em outros formatos não serão revisados.**
  - Slides da apresentação do trabalho em formato pdf.
    - **Os slides poderão ser minimamente alterados entre a entrega do trabalho e a apresentação.**

## 4. Datas Importantes

- **Data de entrega: 29/06**

- **Datas de apresentação: Início 29/06 e datas conforme planilha que será compartilhada futuramente.**