

Intelligence Artificielle

TP 5 : Optimisation des Trajets des Bus Universitaires avec A*

Objectifs du TP

- Construire un graphe pondéré orienté à partir d'un dataset réel
- Implémenter A* pour la recherche du chemin optimal
- Implémenter une approche Greedy Best-First
- Comparer coûts, performance et nombre de nœuds explorés

Problème Étudié

L'université dispose d'un réseau interne de bus reliant plusieurs points stratégiques du campus (résidence universitaire, bibliothèques, bâtiments de cours, administration, restauration, sport...). Chaque déplacement est associé à un temps de trajet réel en minutes, dépendant du trafic et de la distance.

Ce TP a pour objectif d'implémenter un système permettant de calculer automatiquement le chemin le plus rapide entre deux arrêts, en utilisant l'algorithme A*, puis de comparer avec une recherche gloutonne (Greedy Best-First Search).

Dataset: bus_network.csv

Un fichier externe représente les trajets directs entre arrêts :

Colonne	Signification
from_stop	Arrêt de départ
to_stop	Arrêt d'arrivée
time_min	Durée du trajet (minutes)
h_from, h_to	Estimations heuristiques de distance vers un bâtiment cible

```
from_stop,to_stop,time_min,h_from,h_to
RES,P1,5,18,15
RES,RU,7,18,14
P1,SCI,6,15,9
P1,BIB,4,15,8
RU,BIB,3,14,8
RU,SPT,6,14,12
BIB,SCI,4,8,9
BIB,ADM,5,8,6
SCI,INFO,7,9,0
SCI,ADM,4,9,6
ADM,INFO,5,6,0
SPT,ADM,8,12,6
P2,SPT,5,16,12
P2,P1,6,16,15
```

Travail demandé

1. Charger le fichier CSV dans un DataFrame.
2. Construire une structure Python sous forme de dictionnaire

- ```
graph = {
 "RES": {"P1":5, "RU":7},
 ...
}
3. Stocker séparément les heuristiques dans un dictionnaire (vous pouvez utiliser l'algorithme Dijkstra depuis le but INFO sur graphe inversé)
heuristic = {
 "RES": 18,
 ...
}
4. Implémentation de l'algorithme A*
5. Implémentation Greedy Best-First
6. Tester :
```

| <b>Départ</b> | <b>Destination</b> |
|---------------|--------------------|
| RES           | INFO               |