

Etude pratique de la complexité : Algorithme d'Edmonds-Karp

Samy Braik

10 Décembre 2023

1 Introduction

On a vu dans la partie sur l'étude théorique de la complexité de l'algorithme d'Edmonds-Karp que celui-ci fonctionnait en temps $O(|V| \cdot |E|^2)$ sur un graphe $G = (V, E)$.

Ainsi, étant donné que la complexité dépend de deux paramètres, j'ai successivement fixé un des deux paramètres pour pouvoir mesurer le temps d'exécution seulement par rapport au second paramètre.

2 Complexité par rapport au nombre d'arcs

En fixant le nombre de noeuds à 200 et en faisant augmenter le nombre d'arcs de 3 à 970, on obtient bien un temps d'exécution quadratique par rapport au nombre d'arcs, comme en atteste la figure 1.

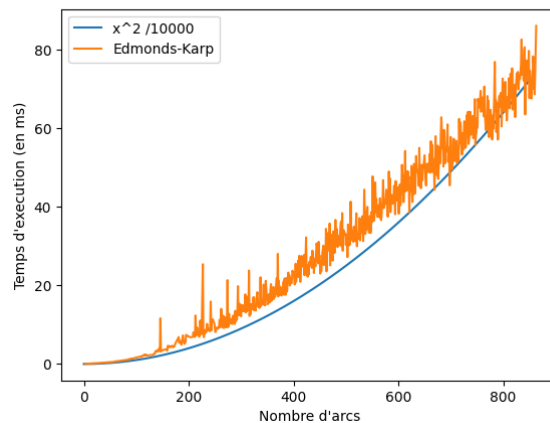


Figure 1: Temps d'exécution par rapport au nombre d'arcs

3 Complexité par rapport au nombre de noeuds

De la même manière que précédemment, en fixant le nombre d'arcs à 100 et en faisant augmenter le nombre de noeuds de 1000 à 4500, on obtient un temps d'exécution linéaire.

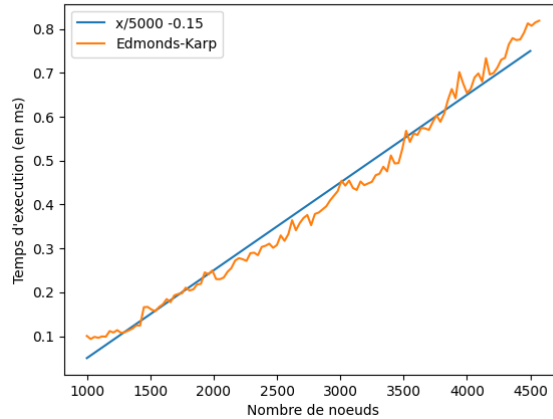


Figure 2: Temps d'exécution par rapport au nombre de noeuds

4 Conclusion et précisions

Ainsi, la complexité observée lors de cette étude pratique est cohérente avec la complexité obtenue lors de l'étude théorique de l'algorithme.

J'ai pu cependant observer que le temps d'exécution tendait à se stabiliser après avoir suffisamment augmenté le nombre d'arcs, tout en gardant le nombre de noeuds constant.

L'algorithme a été implémenté comme méthode de la classe Graph qui est définie dans le fichier "Graph.py". Le fichier nommé "testNodeConstant.py" teste l'algorithme d'Edmonds-Karp sur les graphes contenus dans les fichiers texte du dossier "testNC", c'est-à-dire sur les graphes ayant un nombre constant de noeuds mais un nombre variant d'arcs.

Le fichier nommé "testEdgeConstant.py" teste quant à lui l'algorithme sur les graphes contenus dans les fichiers texte du dossier "testEC", c'est-à-dire sur les graphes ayant un nombre constant d'arcs mais un nombre variant de noeuds.