BERGER LUC
DOYEN Samuel
ETIEVANT Séverin
MONTEIRO Adrian
I1, Groupe L

Complexité

La complexité est un moyen de prédire grossièrement le temps d'exécution d'un algorithme. Grossièrement car le temps d'exécution est encadré d'un minorant et d'un majorant correspondant respectivement au cas d'exécution le plus rapide et le plus lent testé.

Mesure du temps

La mesure du temps est on ne peut plus simple. Pour faire une analogie, ça ressemble à démarrer un chronomètre en début d'algorithme et l'arrêter une fois celui-ci terminé. En réalité, la programmation est un peu plus complexe puisqu'elle demande d'enregistrer les temps auxquels l'algorithme a démarré et cessé de fonctionner puis de faire une soustraction entre les deux pour obtenir le temps d'exécution.

Pour rentrer un peu plus dans les détails, nous séparons les valeurs de mesure de temps en deux temps :

D'une part, nous enregistrons dans 3 listes différentes l'ensemble des temps individuels pour chaque exécution des algorithmes Ford-Fulkerson, Pousser-réétiqueter, Flot minimum pour chaque valeur de n sur 100 itérations, ce qui nous servira pour la suite de l'analyse.

D'autre part, nous additionnons ces temps sur les 100 itérations pour chaque valeur de n pour obtenir un temps global. On réalise cette étape pour chaque algorithme demandé et on obtient le résultat suivant :

(Avec N le nombre de sommets)

```
N : 10
Ford-Fulkerson total : 0.012821 s
Push-Relabel total   : 0.048202 s
Min-Cost Flow total  : 0.055685 s

N : 20
Ford-Fulkerson total : 0.051551 s
Push-Relabel total   : 0.279874 s
Min-Cost Flow total  : 0.450873 s

N : 40
Ford-Fulkerson total : 0.207024 s
Push-Relabel total   : 2.475591 s
Min-Cost Flow total  : 2.738462 s

N : 60
Ford-Fulkerson total : 0.510705 s
Push-Relabel total   : 7.565460 s
Min-Cost Flow total  : 8.784432 s
```

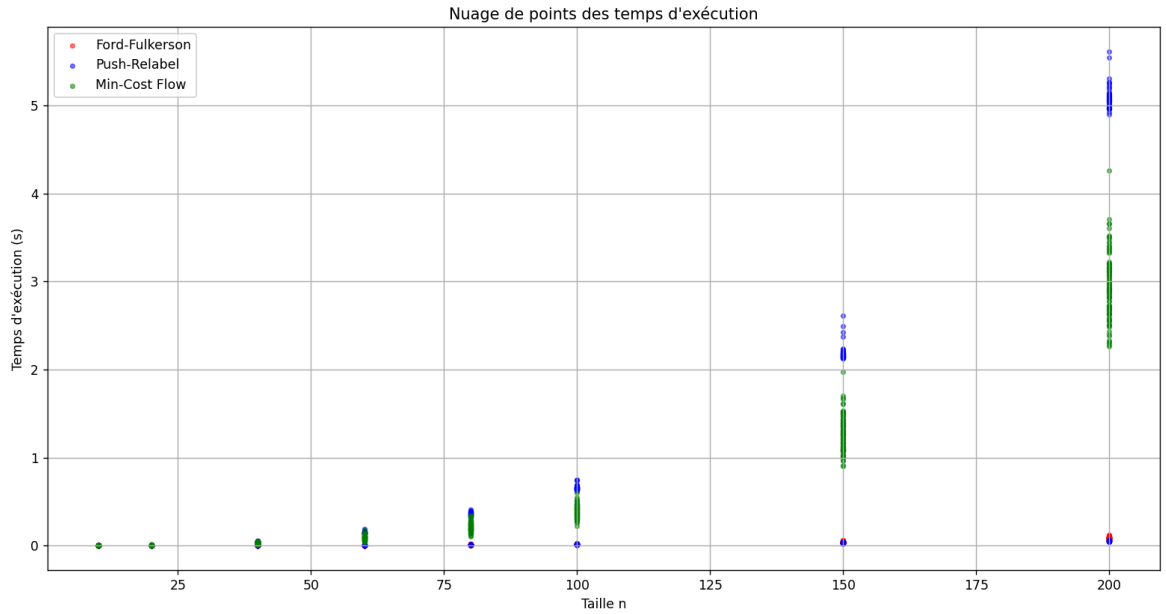
```
N : 80
Ford-Fulkerson total : 0.971784 s
Push-Relabel total   : 15.497426 s
Min-Cost Flow total  : 21.296005 s

N : 100
Ford-Fulkerson total : 1.643307 s
Push-Relabel total   : 34.835170 s
Min-Cost Flow total  : 38.861677 s

N : 150
Ford-Fulkerson total : 4.131040 s
Push-Relabel total   : 102.535982 s
Min-Cost Flow total  : 127.148195 s

N : 200
Ford-Fulkerson total : 8.531761 s
Push-Relabel total   : 262.196721 s
Min-Cost Flow total  : 294.888983 s
```

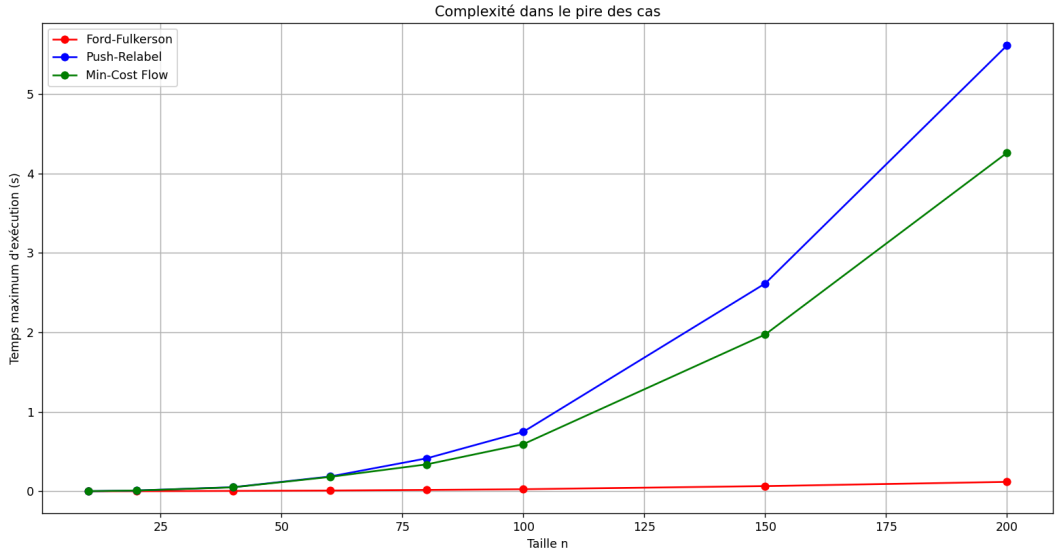
Nuage de point



Voici le nuage de points obtenu. Il représente la dispersion du temps d'exécution de chaque algorithme en fonction du nombre de sommets testés. Vous pouvez observer en rouge l'algorithme de Ford-Fulkerson, en bleu Pousser-Réétiqueter, et en vert le Flot minimal.
On remarque que pour chaque taille de flot testé, Pousser-réétiqueter sera toujours au-dessus du flot minimal qui sera lui-même au-dessus de Ford-Fulkerson à l'exception de rares itérations du plus lent qui à certaines occasions obtient un temps d'exécution équivalent au plus rapide.

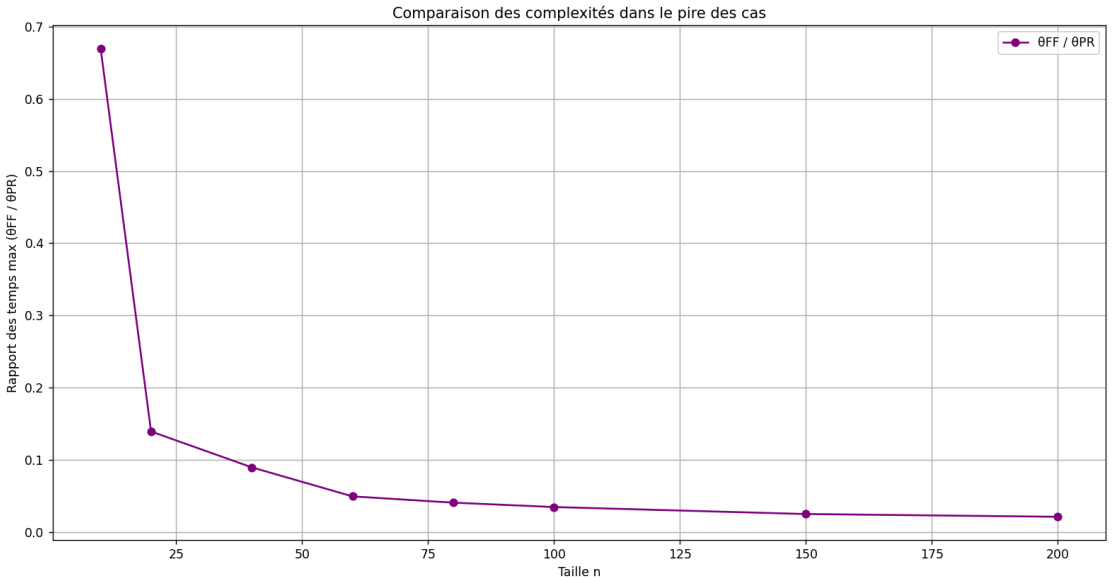
La complexité dans le pire des cas par algorithme

Nous récupérons ensuite les listes de tous les temps mesurés qui contiennent 100 valeurs par taille n. Puis pour chaque algorithme (Ford-Fulkerson, Pousser-réétiqueter, Flot minimum), on extrait la valeur maximale observée par chaque n. Nous traçons ensuite ces valeurs maximales en fonction de n pour visualiser le temps d'exécutions dans le pire des cas par algorithme :



- Graphiquement, on observe pour l'algorithme Ford-Fulkerson est linéaire, et donc de complexité O(n), en effet, il n'y a pas beaucoup de variation entre les différentes tailles testées. En revanche, le cours nous indique que la complexité de cet algorithme est polynomiale $O(VE^2)$. On peut proposer une remarque face à cette divergence. Nous n'avons peut-être pas testé avec suffisamment de sommets pour pouvoir constater la réelle évolution de la courbe de Ford-Fulkerson.
- À première vue, on observe également que l'algorithme de Pousser-Réétiqueter que sa complexité a une allure d'exponentielle ($O(k^n)$). En comparant avec le cours, on peut affirmer que c'est bien le cas puisqu'il est indiqué que Pousser-réétiqueter est censé avoir une complexité exponentielle.
- Pour le troisième et dernier algorithme, le Flot minimum, on observe une fonction exponentielle de complexité $O(k^n)$. Cependant, il n'y a aucune information quant à la complexité calculée mathématiquement pour cet algorithme dans le cours. On ne peut donc pas comparer notre résultat.

Comparaison de la complexité dans le pire des cas



L'ensemble des observations préalablements faites se répercutent sur ce graphique qui étudie le rapport de la complexité de Ford-Fulkerson sur celle de Pousser-réétiqueter en fonction du nombre de sommets. On rappelle que la complexité du dernier algorithme est exponentielle, ce qui permet nous permet d'observer en connaissance de cause la décroissance caractéristique d'une exponentielle inverse.

Conclusion

D’après ce que nous avons pu apprendre du cours sur les différents algorithmes, le classement des complexités de chacun ne semble pas anormal étant donné que Pousser-réétiqueter est un algorithme qui demande parfois de réaliser beaucoup plus d’actions pour arriver à bout d’un problème de flot maximal. Là où celui de Ford-Fulkerson arrive à un même résultat en moins d’actions. Ainsi, on peut conclure que lorsqu’il ne s’agit pas d’un problème de flot prenant en compte des coûts, la méthode de Ford-Fulkerson, bien que moins intuitive et imagée que Pousser-réétiqueter, est plus efficace lorsqu’il s’agît de traiter de gros volumes de données (même si nos machines peinent à passer les 1000 sommets).