15/05/2018

Algorithmique Avancée

Projet n° 2 : Algorithme de Karger

Jérémy JUNAC

Johann MORTARA

Analyse de l'algorithme

Idée de l'algorithme

L'algorithme que nous étudions est l'algorithme de Karger. L'objectif est de trouver la coupe minimum d'un graphe par contractions successives d'arêtes choisies de façon aléatoire. Ici, nous nous positionnons dans le cas d'un graphe non-orienté non-pondéré. Lorsque le graphe ne possède plus que deux sommets, on ne peut plus contracter d'arêtes ; la valeur de la coupe minimum correspond au nombre d'arêtes entre ces deux sommets.

Implémentation "classique"

Dans sa version classique, l'algorithme sélectionne de façon aléatoire une arête du graphe et la contracte, jusqu'à ce qu'il ne reste plus que deux sommets ; la valeur de la coupe minimum correspond au nombre d'arêtes entre ces deux sommets.

Implémentation "améliorée"

Dans sa version améliorée, l'algorithme classique est lancé sur le graphe jusqu'à ce qu'il ne reste plus que $\frac{n}{\sqrt{2}}$ sommets. Ensuite, deux récursions sont réalisées sur le graphe obtenu précédemment, jusqu'à ne plus obtenir que deux sommets. La valeur de la coupe minimum correspond à la plus petite des valeurs obtenues.

Notre implémentation

Le déroulement de notre implémentation est identique à celui de la version améliorée, à deux différences près :

- l'algorithme classique est lancé sur le graphe jusqu'à ce qu'il ne reste plus que $\frac{n}{a}$ sommets.
- b branchements sont effectuées, puis cet algorithme est répété jusqu'à ne plus obtenir que deux sommets.

La démarche expérimentale ayant abouti à ces valeurs est décrite dans la prochaine partie.

Résultats obtenus

En analysant la description qui a été faite de l'algorithme en première partie, nous remarquons que plus a est grand, plus le nombre d'arêtes qui sont contractées avant le lancement des b récursions est grand également. Cela signifie donc moins de lancements de récursions, et donc autant de résultats en moins, ce qui limite par conséquent nos chances d'obtenir la valeur correcte de la coupe minimum. Une faible valeur de a semble donc appropriée.

De même, plus b est grand (donc plus le nombre de récursions lancées et grand) et plus nous obtenons de résultats en sortie de l'algorithme, et donc plus nous avons de chance d'obtenir la valeur correcte de la coupe minimum, ce qui est tout à notre avantage.

Proposition de valeurs pour a et b

Recherche d'une valeur optimale de b

Pour confirmer cette théorie, nous avons lancé l'algorithme récursif sur une grande suite de graphes générés aléatoirement. La figure 1 représente un résumé de nos statistiques.

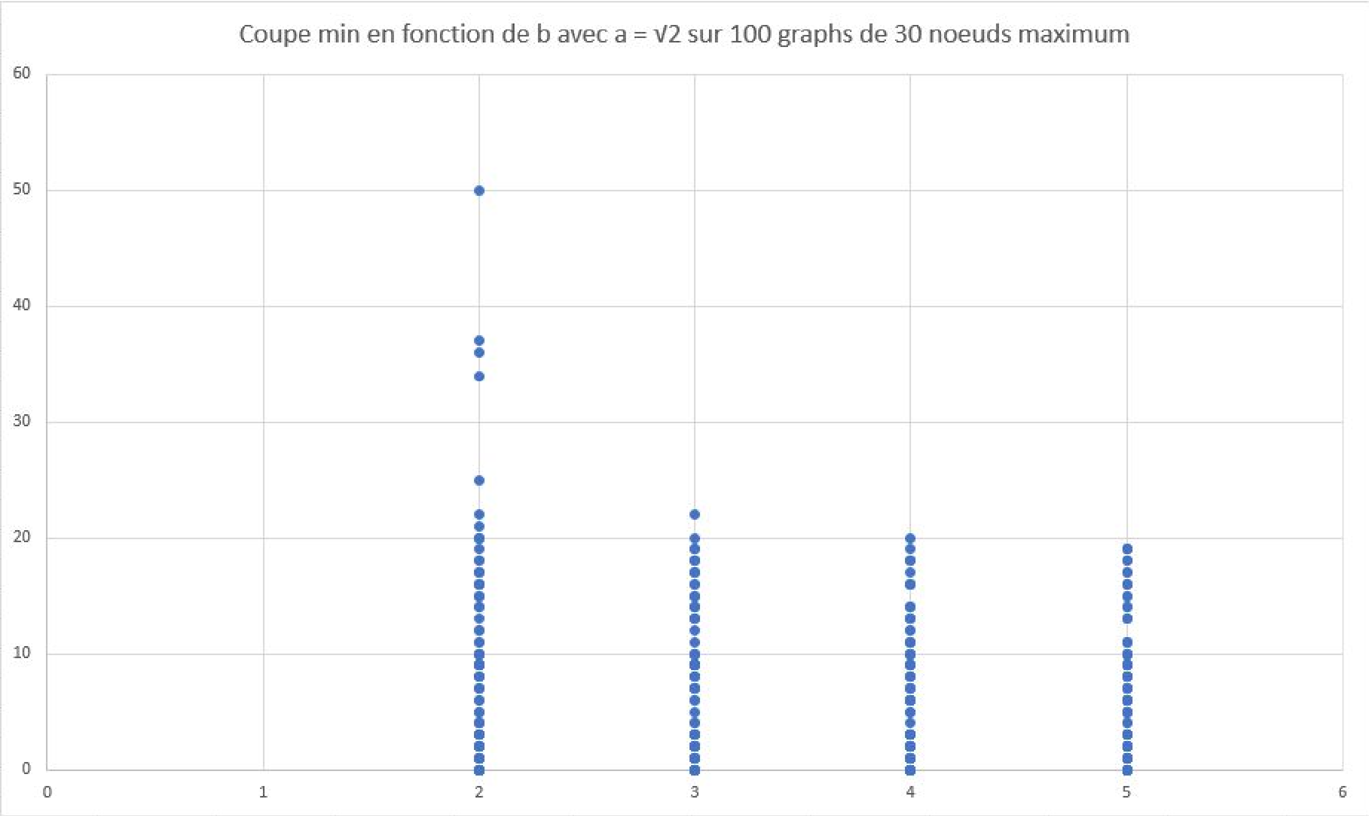

Figure 1

Nous avons fixé $a = \sqrt{2}$ et fait varier la valeur de b. Nous remarquons bien que plus b et grand, moins nous obtenons de valeurs de coupe minimum élevées, et donc plus l'écart entre la valeur exacte et la valeur obtenue sera faible.

Recherche d'une valeur optimale de a

Nous allons ici aussi tenter de confirmer la théorie. La figure 2 représente un résumé de nos statistiques.

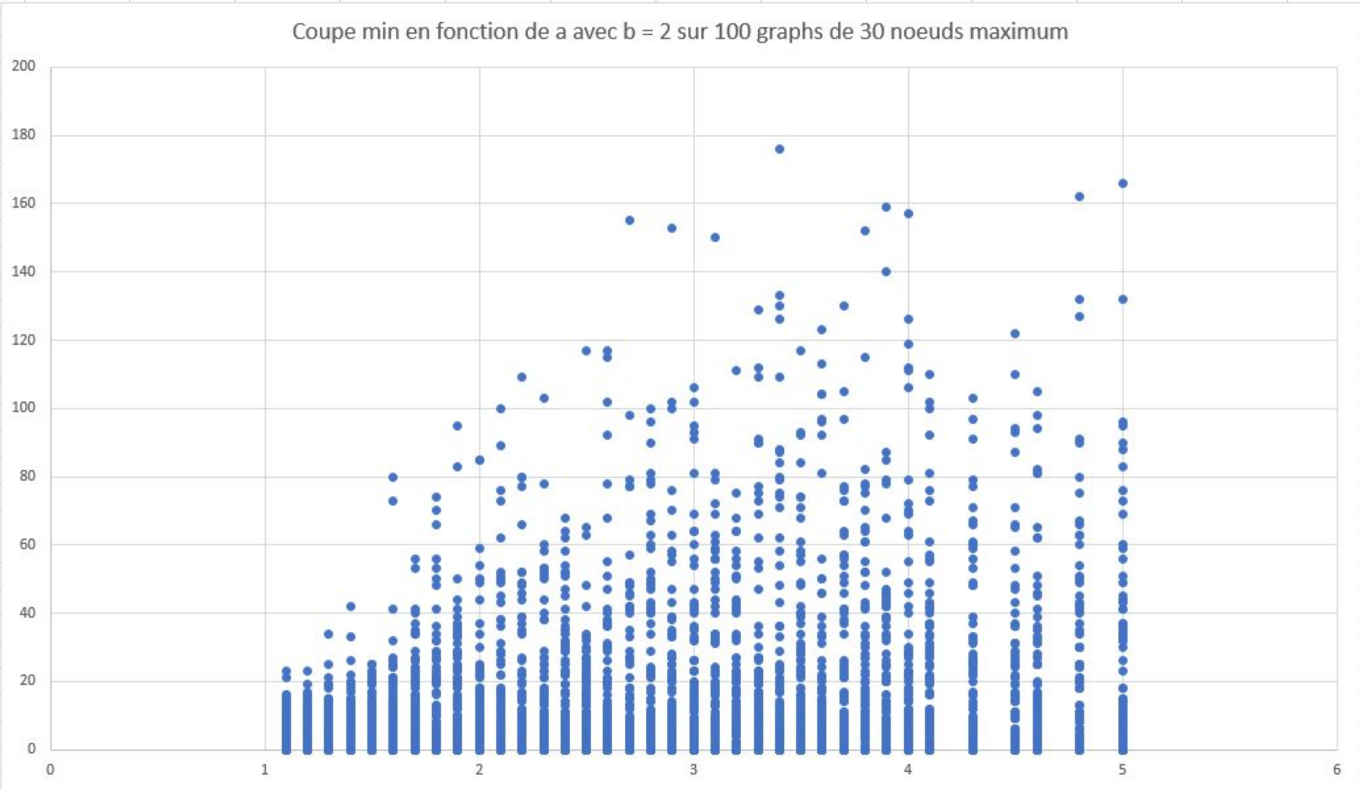

Figure 2

Là encore, la théorie est confirmée, et nous voyons bien que plus a est petit, plus nous avons de chance d'obtenir une coupe minimale proche de la valeur exacte.

Choix des valeurs de a et b

Ici, un dilemme se pose. Nous pouvons soit privilégier les performances en prenant a grand et b petit, soit privilégier la précision des valeurs en prenant a petit et b grand. Là encore, nous allons nous appuyer sur l'expérimentation pour trouver les valeurs proposant le meilleur compromis.

.

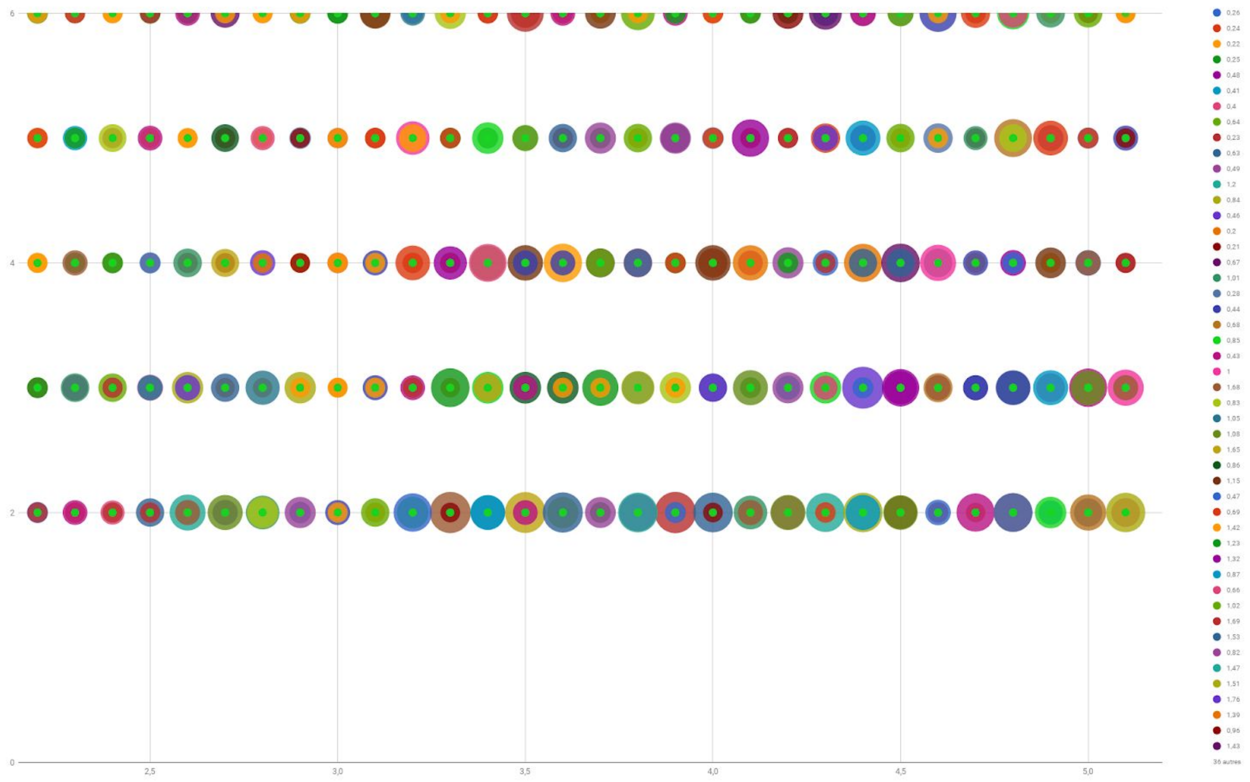

Figure 3 : diagramme à bulles représentant la valeur de la coupe minimum (une boule plus petite équivaut à un meilleur résultat) en fonction des valeurs de a (en ordonnées) et de b (en abscisses)

NB : Afin de faciliter la lecture du graphique, celui-ci a été effectué sur les 600 premières valeurs obtenues en sortie de l'algorithme.

En nous basant sur la partie de graphique que nous voyons, et les données disponibles dans le fichier `interesting_results/search_a_b_maxv_30_10_graphs.csv`, nous observons que $a = 5$ et $b = 3$ semblent être de bonnes valeurs.

Comparaison des algorithmes

Maintenant que nous avons obtenu des valeurs pour a et b, nous pouvons à présent comparer les 3 versions de l'algorithme que nous rappelons brièvement ici :

- version classique
- version améliorée avec $a = \sqrt{2}$ et $b = 2$
- version améliorée avec $a = 3$ et $b = 5$

Pour comparer ces algorithmes, nous les avons appliqués sur une série de 1000 graphes générés aléatoirement de manière uniforme grâce au modèle de Erdős–Rényi.

Algorithme	Coupe minimale	Nombre de contractions
Version classique	42,358	17,954
Version améliorée avec $a = \sqrt{2}$ et $b = 2$	8,853	97,911
Version améliorée avec $a = 3$ et $b = 5$	28,756	21,837

Tableau 1 : Résumé de l'exécution des différents algorithmes sur 1000 graphes générés aléatoirement de manière uniforme avec maximum 40 noeuds

Ce tableau nous montre un des résultats auxquels nous nous attendions. La version classique est peu précise mais extrêmement efficace en nombre de contractions.

Un résultat un peu plus intéressant et inattendu, la différence entre les deux versions améliorées. En effet, selon les résultat d'au-dessus, nous avons choisi nos a et b de manière à avoir un compromis entre performance et précision. Au final, on remarque qu'avec $a = 3$ et $b = 5$, nous sommes à mi-chemin entre la version classique et la version améliorée avec $a = \sqrt{2}$ et $b = 2$ en terme de précision, mais nous sommes très proche du Karger classique pour le nombre de contractions. En diminuant les performances de 22%, nous gagnons donc 68% de précision. Cette algorithme est donc extrêmement efficace. On observe aussi qu'un a plus grand (un peu moins du double) peut être compensé par un b plus grand (un peu plus du double), avec une meilleur efficacité. Il serait intéressant de voir si cette limite se confirme avec des grands nombres.

Quant à la version améliorée avec $a = \sqrt{2}$ et $b = 2$, elle tient toutes ses promesses en donnant des résultats assez précis (par rapport aux autres versions de l'algorithme de Karger), mais avec un coût non négligeable. Là encore, il serait intéressant de pousser l'expérience plus loin et de comparer, en particulier cette version, mais aussi les autres, à un algorithme qui donne le bon résultat à coup sûr (comme par exemple l'algorithme de Stoer–Wagner). Cela nous permettrait d'observer quel est la marge d'erreur de ces algorithmes probabilistes de manière factuelle.

Conclusion

Au travers de cette analyse, nous avons pu voir que le choix d'un algorithme précis pour résoudre ce problème nécessite la prise en compte de plusieurs paramètres. En effet, dépendant de ce que l'on cherche à améliorer, notre choix ne sera pas le même. Si l'on veut minimiser le nombre de contractions, l'algorithme classique est le bon choix ; il évite tout appel récursif, mais ne renvoie qu'un seul résultat. Les algorithmes améliorés renvoient plusieurs valeurs, ce qui permet une comparaison et une meilleure probabilité d'obtenir le résultat correct. Là encore, le choix des valeurs pour a et b sera motivé par une volonté de privilégier soit la rapidité d'exécution (a grand et b petit), soit la probabilité d'obtenir la véritable valeur de la coupe minimale (a petit et b grand).