

Facility Location Problem

Garnaud Raphaël

30 avril 2017

Introduction

Pendant ce projet nous allons essayer de définir l'algorithme ayant les meilleures performances, et de résoudre celui fourni dénommé D1.1. Pour cela nous allons jouer avec quelques paramètres. Tout d'abord le choix de l'installation à étudier, ensuite l'ordre des nœuds dans l'arbre de branchement à traiter et enfin le calcul de la borne inférieure. Pour choisir un algorithme (modèle) par rapport à un autre, on choisira celui dont le temps de résolution est minimal et nous verrons ensuite le temps d'apparition de la solution.

Quatre installations et trois villes

On commence par étudier le paramètre du choix de la règle de branchement. On observe que le modèle le plus performant est le modèle 2, celui utilisant les installations les plus coûteuses (en terme de coût de placement) en premier, bien que le modèle trouvant la solution en premier soit le modèle considérant les installations les moins coûteuses en premier. Cela peut s'expliquer par le fait que le modèle 3 va aller au plus simple en prenant les moins chers, il aura donc la solution plus rapidement. Mais ensuite il faut qu'il vérifie toutes les autres possibilités, ce qui prendrait donc plus de temps.

La surprise de ce résultat est le manque d'efficacité du modèle 4 sensé être le plus performant, cela pourrait s'expliquer par le temps de calcul supplémentaire dû aux différentes boucles dans la fonction `choix_borne_inf`. On peut espérer que ce manque d'efficacité n'existe que pour les données faciles, et que pour les prochaines instances, il gagne en performance par rapport aux autres. Pour l'instant, on retient donc que le paramètre le plus performant est la règle de branchement utilisant l'installation la plus coûteuse.

Cette expérience permet également de vérifier que les différents algorithmes ou modèles donnent bien la bonne solution ici 510.

Modèles	Temps de résolution	Apparition	Noeuds traités	Solutions générées
Modèle 1	0.000774 s	0.000179	6	0
Modèle 2	0.000753 s	0.000244 s	8	1
Modèle 3	0.000761 s	$7.9 \cdot 10^{-5}$ s	9	0
Modèle 4	0.000851 s	0.000282 s	7	0

Trente emplacements et quatre-vingts villes

On étudie ensuite le deuxième paramètre : l'ordre de traitement des nœuds de branchements, c'est à dire quel est le prochain nœud que l'on va étudier.

On observe que le modèle le plus performant, avec cette donnée (de la taille de notre objectif), est encore le modèle 2. Ce qui semble là encore étrange puisque le deuxième modèle ne bénéficie pas de l'amélioration du "meilleur d'abord". Cela pourrait encore s'expliquer par l'utilisation de boucles supplémentaires. Ainsi le meilleur paramètre pour les nœuds de branchements semble être le Depth-first.

Malheureusement, le résultat semble également confirmer que ma fonction `choix_borne_inf` est mal écrite, vu qu'elle n'est pas plus performante avec des données plus difficiles. Ainsi, avec les algorithmes que j'ai écrits, le meilleur paramètre du choix de la règle de branchement est celui utilisant l'installation la plus coûteuse.

Néanmoins, d'un point de vu du nombre de nœuds traités, le modèle 5 est plus performant que le modèle 2. Pour finir même si nous n'avons pas la solution, vu que cette instance a été créée par la génération aléatoire de Blide-Karup, les 5 modèles fournissent la même solution : 23301, nous pouvons donc dire que ces modèles sont cohérents entre eux.

Modèles	Temps de résolution	Apparition	Nœuds traités	Solutions générées
Modèle 1	1.54519 s	1.1656 s	13322	0
Modèle 2	0.44856 s	0.026186 s	3194	10
Modèle 3	5.13368 s	4.91228 s	41636	1
Modèle 4	1.52783 s	1.51182 s	10055	12
Modèle 5	1.52268	0.89774 s	4041	1

Quatre-vingts emplacements et quatre-vingts villes

Pour finir, nous étudions le paramètre du calcul de la borne inférieure avec des instances beaucoup plus grande. Au vu des observations, il semblerait que le modèle 2 reste plus performant que le modèle 6. Cela semble difficile à expliquer, vu que la seule "perte de temps" est une condition "if" supplémentaire, ce qui demande très peu de temps de calcul.

Nous pouvons constater que le modèle 6 est moins performant, en ce sens qu'il traite plus de nœuds, ce que l'amélioration de la borne inférieure aurait justement du changer, cela semble prouver qu'il y a un problème dans le calcul de la borne inférieure. Néanmoins, on peut constater que le modèle 6 trouve un peu plus rapidement la solution que le modèle 2.

Enfin, les deux modèles donnent bien la même solution : 18108.

Modèles	Temps de résolution	Apparition	Nœuds traités	Solutions générées
Modèle 2	20.6499 s	0.075181 s	76594	7
Modèle 6	21.4709 s	0.080172 s	76601	7

Conclusion

Pour conclure après plusieurs essais avec les deux derniers modèles pour résoudre l'instance D1.1, cela n'a pas fonctionné (même en y consacrant 20 minutes). Nous pouvons expliquer cette absence de solution par les nombreux problèmes que nous avons pu voir dans ces algorithmes. En même temps, nous pouvons trouver étrange que ces deux algorithmes ne trouvent pas de solutions alors qu'ils en trouvent pour des instances plus grandes (80,80).

Correspondance des modèles

Modèles	Règle de branchement	Prochain nœud à traiter	Borne inférieure
Modèle 1	première installation indéterminée	Depth-first	sans payer d'installations indéterminées
Modèle 2	la plus couteuse	Depth-first	sans payer d'installations indéterminées
Modèle 3	la moins couteuse	Depth-first	sans payer d'installations indéterminées
Modèle 4	connecté au plus de villes	Depth-first	sans payer d'installations indéterminées
Modèle 5	la plus couteuse	best first	sans payer d'installations indéterminées
Modèle 6	la plus couteuse	Depth-first	paye l'indéterminée la moins chère