Mini Projet

February 28, 2023

1 Algorithme complétement aléatoire

FullRandom

Tout d'abord pour tester notre implémentation et commencer à tater le terrain, nous avons codé un algorithme qui choisit à chaque demande un technicien aléatoire. Même si cet algorithme n'est pas très intéressant, nous avons effectuer des tests dessus et nous obtenons un ratio de compétitivité de 6.38 (moyenne sur 100 essais). Complexité : $O(log(N))$ (coût de la fonction randint)

2 Algorithme du technicien le plus proche

Closest

Nous avons ensuite décidé d'utiliser un algorithme glouton pour continuer à essayer de comprendre un peu mieux nos instances : à chaque demande reçu, on choisit cette fois-ci le technicien le plus proche. Cet algorithme a été bien plus concluant et nous obtenons un ratio de compétitivité de 2,84 (déterministe) Complexité : $O(N)$

3 Algorithme du technicien le plus proche randomisé

RandomClosest

Cette algorithme consiste à d'abord vérifier si un technicien n'est pas déjà sur place, si oui on le séléctionne, sinon on en prend un au hasard parmis tous les autres techniciens. Nous avons obtenu un ratio de compétitivité de 1.66 (moyenne sur 100 essais). Cet algorithme, pourtant très naïf marche largement mieux que le full random, mais encore plus intéressant : il est bien au dessus d'un algorithme glouton classique que l'on pourrait utiliser sur tout types d'instances. Complexité : $O(N)$

4 Algorithme pondéré aléatoire

WeightedRandom

Cet algorithme choisit un technicien aléatoirement, avec une probabilité dépendante de la distance séparant un technicien du site. On obtient un ratio de compétitivité de 1,55, ce résultat est encore très intéressant : on fait mieux que RandomClosest. Complexité : $O(N)$

5 Algorithme periodique

Periodic

Les algos précédents étaient tous pour des cas assez généraux. En observant nos instances nous avons remarqué que les demandes se répétaient (ABCABCABC...). Le problème de nos algorithmes précédent est que dans le cas où les demandes sont sur des sites proches l'un de l'autre, un seul technicien effectue tous les trajets. Nous avons donc ajouté une condition qui est, de d'abord sortir tous les techniciens encore en (0,0) (la base) puis de faire traiter la demande suivante par le technicien le plus proche (donc en général le technicien est déjà sur le site). Ainsi vu que les demandes sont des périodes, les techniciens n'auront, pour la plupart des cas, pas à bouger après leurs premier déplacement. Et les résultats sont impressionnant pour un algorithme deterministe : on a un ratio de compétitivité de 1,08 (et de 1 sur des instance ayant des périodes de taille inférieur au nombre de technicien). Complexité : $O(N)$

6 Algorithme periodique aléatoire

PeriodicRandom

On garde l'idée de d'abord sortir tous les techniciens, une fois les techniciens sortis, si on a une demande venant d'un site non couvert (distance entre technicien et demande égale à 0) alors on choisit un technicien aléatoire. On obitent un ratio de 1,48, ce qui est peu concluant surtout pour un algorithme basé sur des instances. l'algorithme periodique n'est donc pas intéressant à randomiser. Complexité : $O(N)$

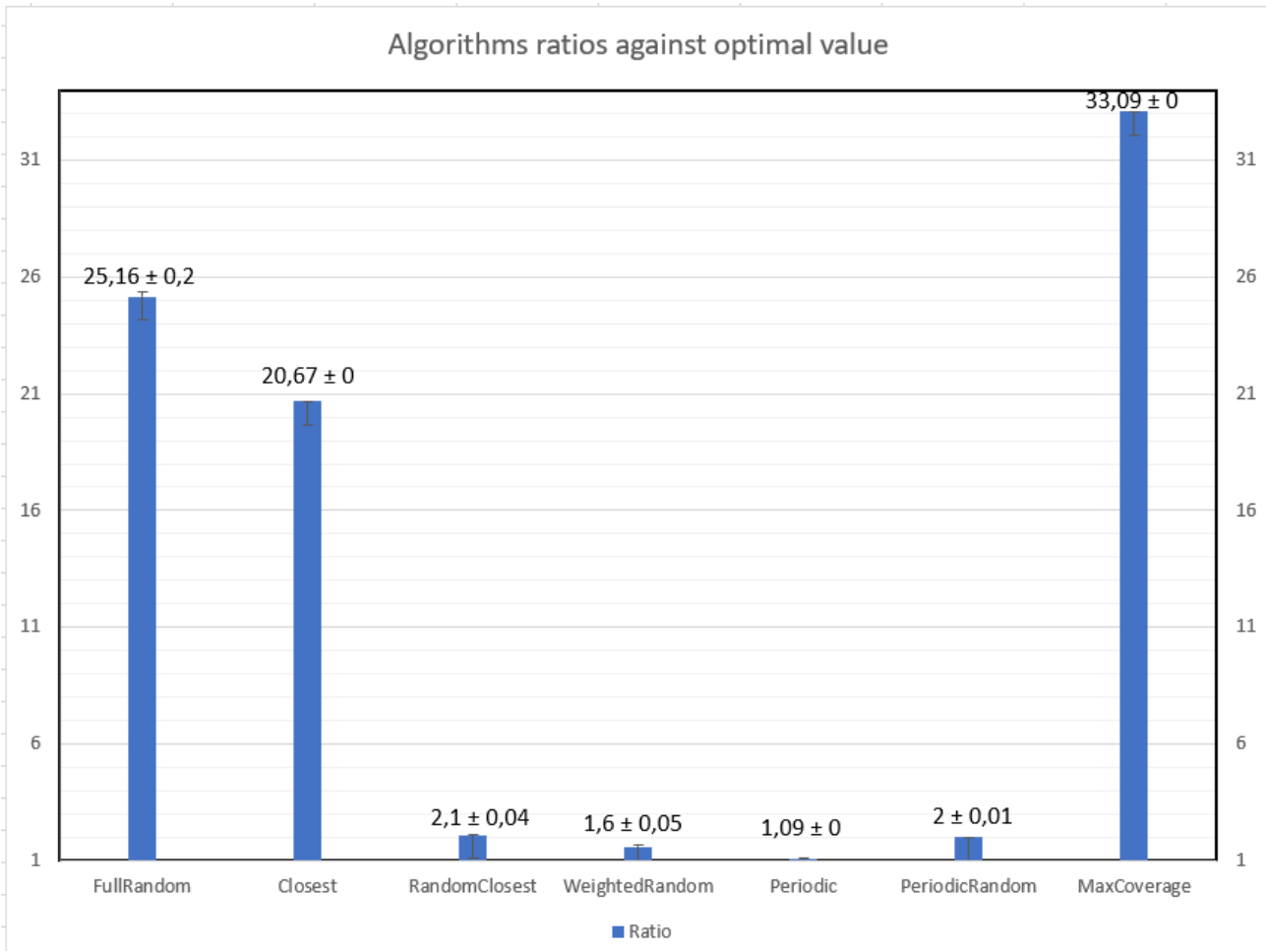
7 Algorithme de couverture maximum

MaxCoverage

Nous avons voulu expérimenter une nouvelle méthode basée sur l'idée de répartir au maximum les techniciens sur la carte, au prix d'une forte complexité: $O(N^3)$. Malheureusement, cette expérience n'a pas été concluante et nous avons obtenu un ratio de 6,33.

8 Conclusion

Nous avons pu obtenir un algorithme très proche de l'optimal (ratio de 1.09 pour Periodic) cependant il est spécifique aux instances sur lesquelles nous travaillions et ne serait surement pas aussi performant dans le cas général. En revanche, l'algorithme WeightedRandom est plutôt performant (ratio de 1.66) et cela sans être adapté aux instances. Pour finir, nous avons compilé les résultats des différents algorithmes présentés ici.


Algorithms ratios against optimal value
31
26
21
16
11
6
1
25,16 ± 0,2
20,67 ± 0
2,1 ± 0,04
1,6 ± 0,05
1,09 ± 0
2 ± 0,01
33,09 ± 0
FullRandom
Closest
RandomClosest
WeightedRandom
Periodic
PeriodicRandom
MaxCoverage
Ratio