Projet en outil logiciel optimisation (problème de génération de coupes)

Fait par : BOUNEJMATE Sarah & YEBLI Pala-Beh
Sous l'encadrement de Mr : TAHIRI Issam

Spé 3 - MIMSE.

Question 1 : Pour avoir la solution pour un C très grand, il suffit d'avoir un graphe connecté. C'est à dire il faut avoir un arbre couvrant.

Question 2 : Variables de décision :

x_{ij} : Nombre de câbles entre i et j.
y_{ij^k} : flot venant de k et passant de i vers j.(orienté)

$$Min\ \Sigma_{i=0..13}\, \Sigma_{j=i+1..14}\ x_{ij}$$
$$\Sigma_j\, y_{ij^k} - \Sigma_j y_{ji^k} = 14 \qquad \forall\ j$$
$$\Sigma_j\, y_{ij^k} - \Sigma_j\, y_{ji^k} = -1 \qquad \forall\ i \neq k \qquad \forall\ j$$
$$\Sigma_k\, y_{ij^k} - \Sigma_k\, y_{ji^k} \leq C\, x_{ij} \qquad \forall\ i \qquad \forall\ j$$

Question 3 : Algorithme combinatoire pour résoudre le problème

```
Entrée : une liste de sommets & de demandes & de voisinage

Sortie : un compteur ( nombre de câbles utilisés)

  list <- 0
  cmpt <-0
  sommetcourant <- N1
  source <- N1
       Tant que demande i non satisfaite faire :
               Appliquer un algorithme de + court chemin de source
vers i et envoyer du flot sur
              chaque arc (i,j) entre source et puit Incrémenter
cmpt de 1 fin tant que ajouter sommetcourant
              à la liste
              sommetcourant <- sommetvoisin
       fin tant que
retourner cmpt
```

Question 4 : Non ce n'est pas possible, il faut que le nombre des câbles soit au moins égal à, sinon l'arbre ne serait plus couvrant et on violera les demandes. Il faut donc que :

$$\Sigma_{i=0..13} \Sigma_{j=i+1..14} \; x_{ij} \geq 14 \,.$$

Question 5 : Une contrainte améliorante serait :

$$\Sigma_{i=0..13} x_{ij} \geq 28/C \,.$$

Question 8 :

Avec aucune des inégalités valides on observe ces résultats :

Capacité	Optimum	Temps d'exécution
1	548	0.02
5	113	0.32
10	58	0.22
15	40	0.39
20	32	0.24
40	17	0.13
60	15	0.21
100	14	0.23

Avec la première inégalité valide ($\Sigma_{i=0..13} \Sigma_{j=i+1..14} \; x_{ij} \square 14$) on observe ces résultats :

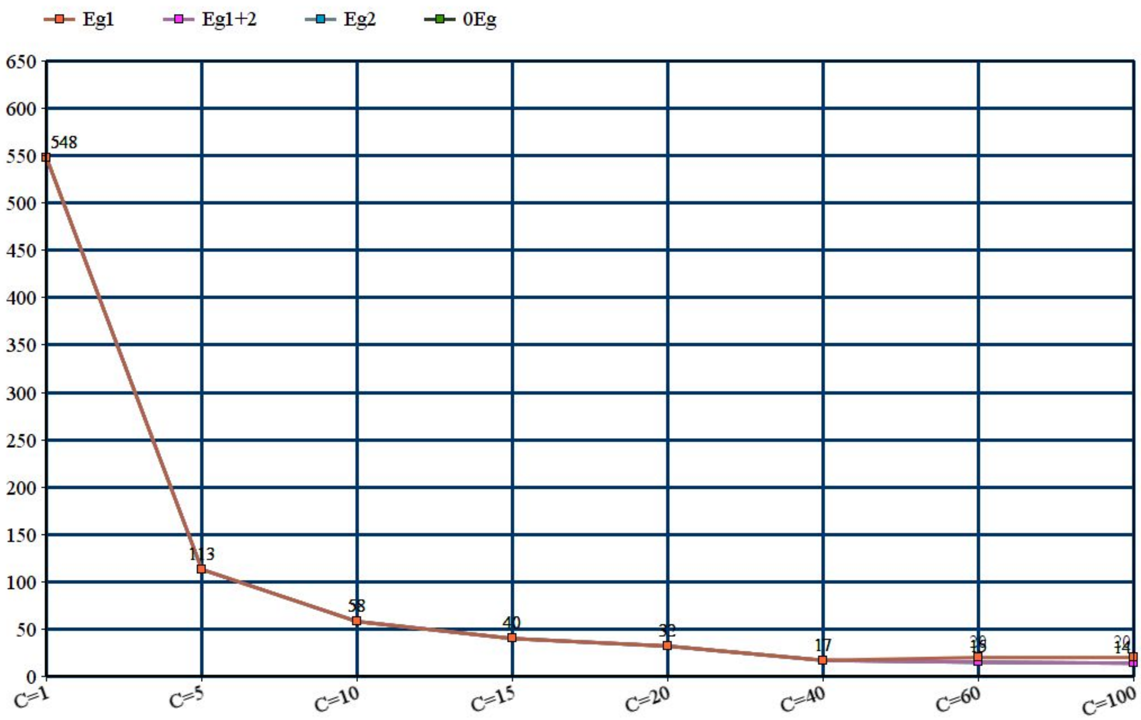
Capacité	Optimum	Temps d'exécution
1	548	0.02
5	113	0.6
10	58	0.13
15	40	0.23
20	32	0.14
40	17	0.08
60	20	0.04
100	20	0.03

Avec la deuxième inégalité valide ($\Sigma_{i=0..13} x_{ij} \square 28/C$) on observe les résultats suivants :

Capacité	Optimum	Temps d'exécution
1	548	0.02
5	113	0.61
10	58	0.17
15	40	0.30
20	32	0.11
40	17	0.14
60	15	0.22
100	14	0.39

Pour les deux inégalités ajoutées à la fois on observe les résultats :

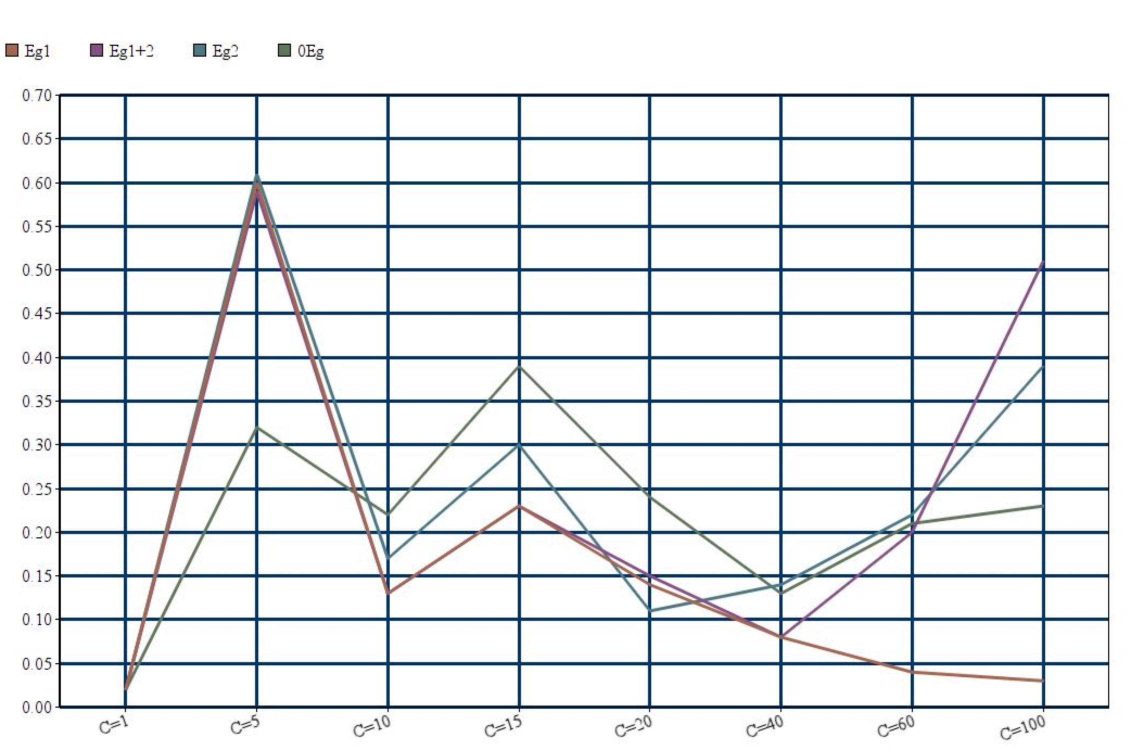
Capacité	Optimum	Temps d'exécution
1	548	0.02
5	113	0.59
10	58	0.13
15	40	0.23
20	32	0.15
40	17	0.08
60	15	0.20
100	14	0.51



Graphe représentant la valeur de l'optimum pour chaque capacité

Comme première remarque on constate que plus la capacité augmente plus l'optimum diminue, chose qui est normale car plus les câbles ont de grandes capacités moins de câbles on aura à installer.

On peut voir que la première inégalité elle réduit beaucoup les temps de calculs et comme elle fait des cuts lors du branchement dès qu'on est à un noeud qui ne satisfait pas la contrainte. Mais son optimum n'est pas très avantageux (ce n'est pas la valeur la plus optimale qu'on peut obtenir c-a-d 14).
Quand aux autres inégalités, les valeurs de l'optimum restent inchangées.



Graphe représentant les temps d'exécution de l'optimum pour chaque capacité

Comme nous pouvons le remarquer, les temps d'exécution évoluent de manière aléatoire selon qu'on ait appliquer ou pas un callback.

Dans le cas où on a aucune inégalité, la courbe évolue de façon aléatoire.
Lorsqu'on applique la première inégalité valide, on remarque que plus la capacité augmente, moins le temps d'exécution est important.
Pour la deuxième inégalité valide, le temps d'exécution ne fait qu'augmenter à partir d'une capacité seuil.
Lorsqu'on applique les deux inégalités, la courbe se confond avec celle où on applique la première inégalité, jusqu'à un seuil ou il croît de façon exponentielle.