

Conception et Pratique de l'Algorithmique

Binh-Minh Bui-Xuan

**Rapport du Projet final :
Arbre de Steiner dans un graphe**

Réalisé par : ABBES Billel

Année Universitaire : 2019/ 2020



Partie 1 : Arbre de Steiner dans un graphe sans budget

Dans la fonction Steiner de la classe DefaultTeam il y'a deux fonctions :

1. *calculSteinerSanspointFermat(points, edgeThreshold, hitPoints)*
2. *calculSteinerAvecpointFermat(points, edgeThreshold, hitPoints)*

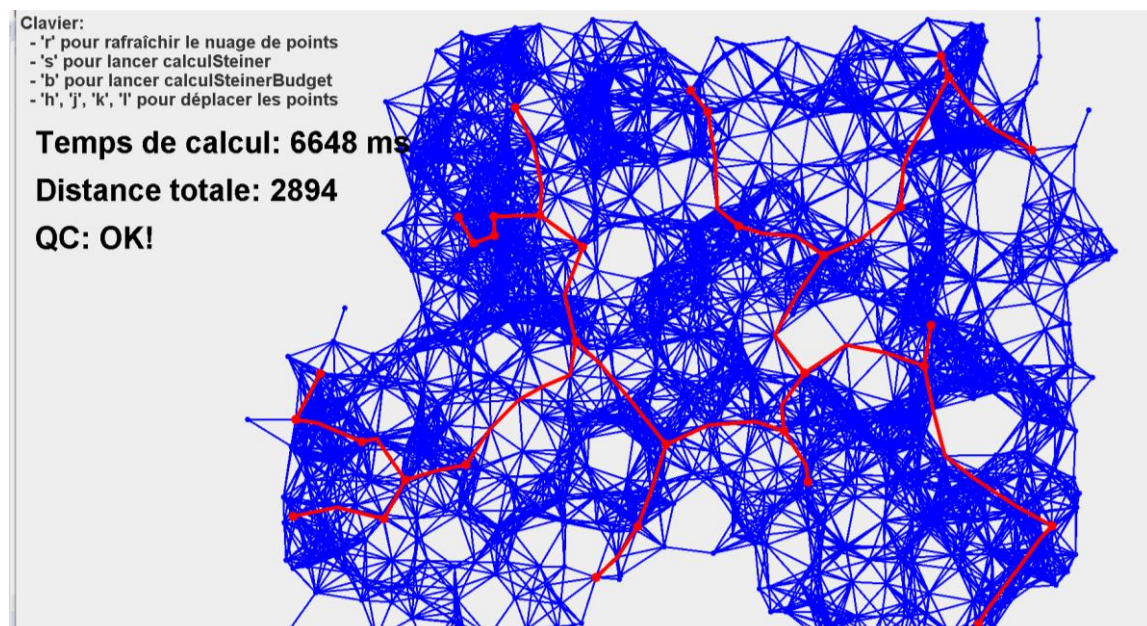
la deuxième fonction, celle du point Fermat est commenté, il va donc falloir la décommenter et commenter la première fonction pour tester.

la deuxième fonction calcule pour $i=500$ le point Fermat.

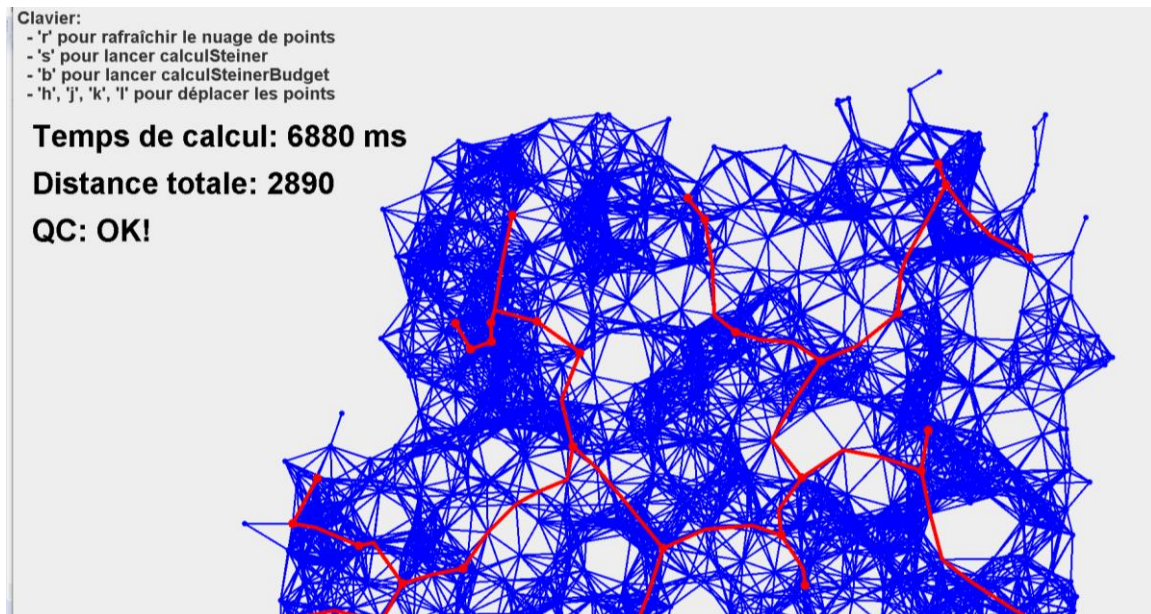
Pour calculer le point de Fermat j'ai rajouté une variable **eps** que j'ai initialisé à 3, pour trouver le point un point dans points et dont la distance de I ne dépasse pas $\text{racine}(3^2+3^2)$

Voici deux résultats du même graphe :

En calculant le point Fermat



En calcule du point Fermat :



On peut remarquer qu'on a gagné 4 unités.

- Dans certain graphe, quand il y a un point qui fait partie de *hitPoints* et qui est isolé (la distance avec le point le plus proche est supérieure à *threshol*d) Le programme rend une erreur car la fonction qui calcule le plus court chemin créer une boucle infinie en tentant de trouver un chemin qui relie le point isolé avec un autre.

Partie 2 : Arbre de Steiner dans un graphe avec budget

Algorithme 1 :

Enlever l'arête la plus grande, ensuite recalculer Steiner du nouvel ensemble des points. Répéter tant que la distance globale du graphe est supérieure au budget.

Cet algorithme n'est pas efficace car dans certain cas il vaut mieux commencer par enlever des arêtes de longueur petite pour accéder à des longues arêtes.

Algorithme 2 :

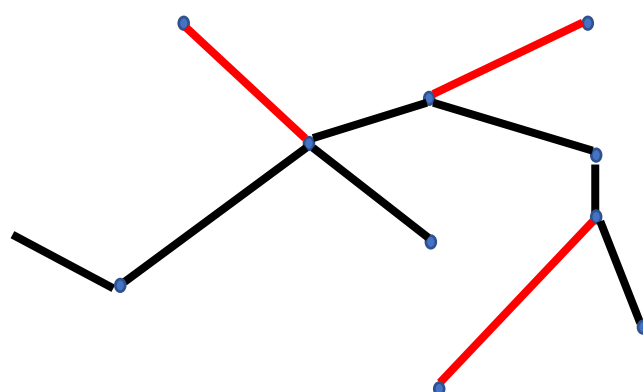
Cet algorithme est basé sur la programmation dynamique, il consiste à calculer le meilleur sous arbre à enlever de l'arbre pour chaque i allant de 1 à $(n-1)$, i présente le nombre de points à enlever.

A l'itération 1 on enlève la plus grande arête, à l'itération 2 on compare entre enlever les 2 arêtes les plus grandes et enlever le sous-arbre de taille 2 le plus grand...etc.

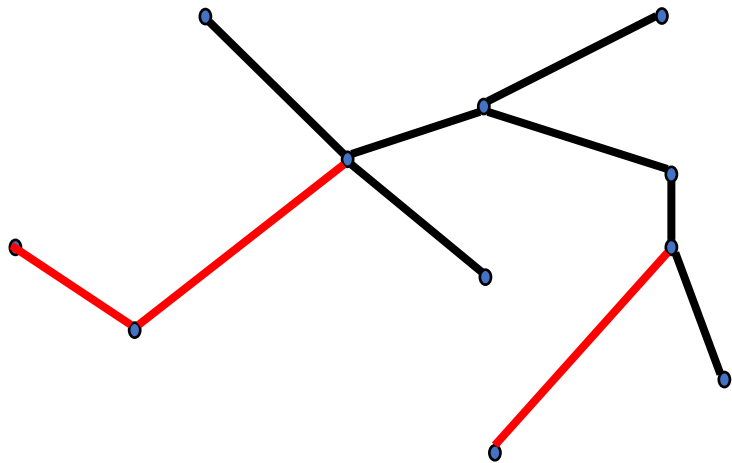
Pour chaque i on calcule toutes les combinaisons dont la somme est égale à i , par exemple pour 3 les combinaisons sont :

$(1 - 1 - 1) (2 - 1) (1 - 2) (3)$.

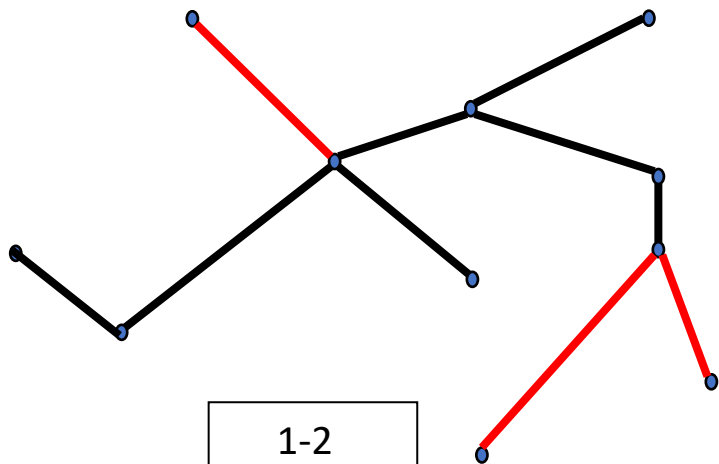
Ensuite on compare la distance de chaque arbre après avoir enlevé des sous arbres de taille x comme la combinaison précédemment indique.



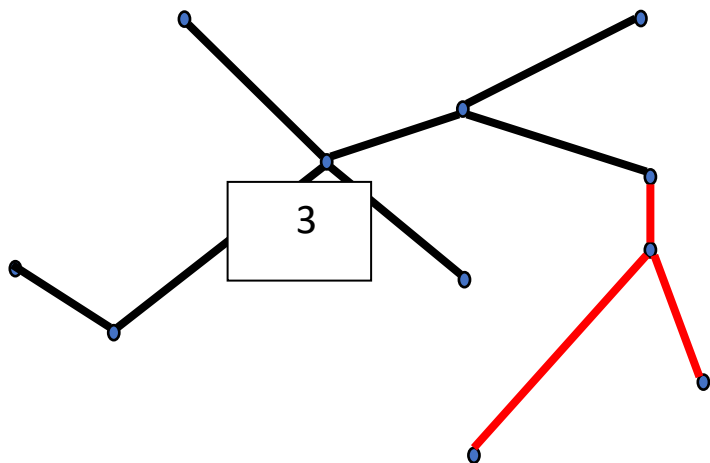
1-1-1



2-1



1-2



3

Pour les 4 arbres on calcule la distance des arêtes noirs, ensuite on les compare entre eux pour déterminer l'ensemble des meilleurs sous arbres à enlevé quand on veut enlever 3 points.

Ce calcule de $i = 3$, sera utile pour calculer la même chose pour $i = 4$.

1-1-1	1-1-1-1
2-1	2-1-1
1-2	1-2-1
3	3-1
	1-1-2
	2-2
	1-3
	4

On peut voir que la moitié des combinaisons de 4 sont les mêmes combinaisons de 3 en ajoutant 1 à droite, cela peut être traduit par :

La solution pour $i = 3$ ensuite enlever la plus grande arête.

L'avantage de cet algorithme c'est qu'il donne des bons résultats, il semble même être optimal vu qu'il calcule toutes les possibilités, mais vu que les combinaisons à calculer sont de complexité exponentielle, car ça se redouble pour chaque itération, à partir de 15 cela commence à devenir lent voire impossible. Cependant, jusqu'à 15, cela reste calculable avec un ordinateur de configuration moyenne.

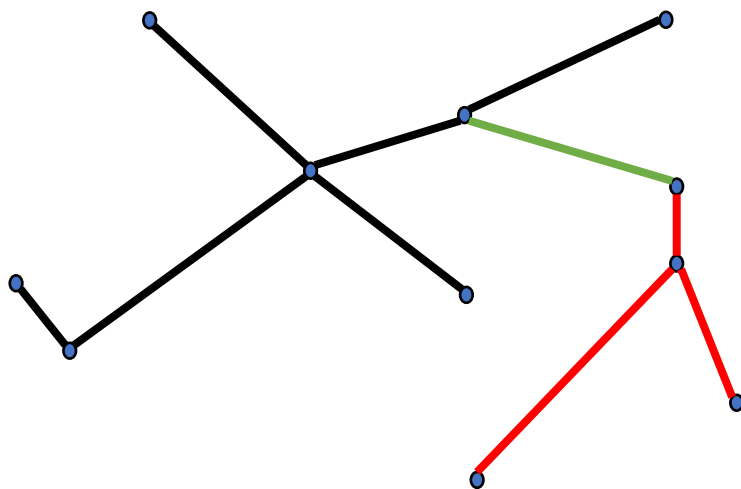
En outre, pour un nombre de points à enlevés i , si on prend l'arbre résultant A après avoir enlevé la liste des sous arbres

rendue par cet algorithme pour i , et si on applique à nouveau Steiner sur l'ensemble de point de l'arbre A , il est possible d'avoir un nouveau arbre de taille inférieure de la taille de A .

Après plusieurs analyses, il semble qu'il vaut mieux de commencer par enlever des gros sous arbres, cela pour viser les arêtes longues proches de la racine, et pour rapprocher du budget, mais pas beaucoup, puis enlever arête par arête en calculer Steiner pour chaque nouvel ensemble de points,

- J'ai rajouté un nouvel Arbre qui hérite de Tree2D,

Un arbre Tree2Dr :



Cela est dû pour la raison suivante : Si on enlève le sous arbre rouge, l'arête verte sera automatiquement supprimée mais on a intérêt à garder l'information qui sera utile prochainement afin de faire des comparaisons, fusionner des sous Arbre, pour calculer la distance gagner ...etc