RAPPORT MÉTHODE COMPUTATIONNELLE

RECUIT SIMULÉ :

LE VOYAGEUR

Maxime Yonnet SIIA1

I. Problème

Un voyageur de commerce doit visiter n villes données en passant par chaque ville exactement une fois. Il commence par une ville quelconque et termine en retournant à la ville de départ. Les distances entre les villes sont connues. Quel chemin faut-il choisir afin de minimiser la distance parcourue ?

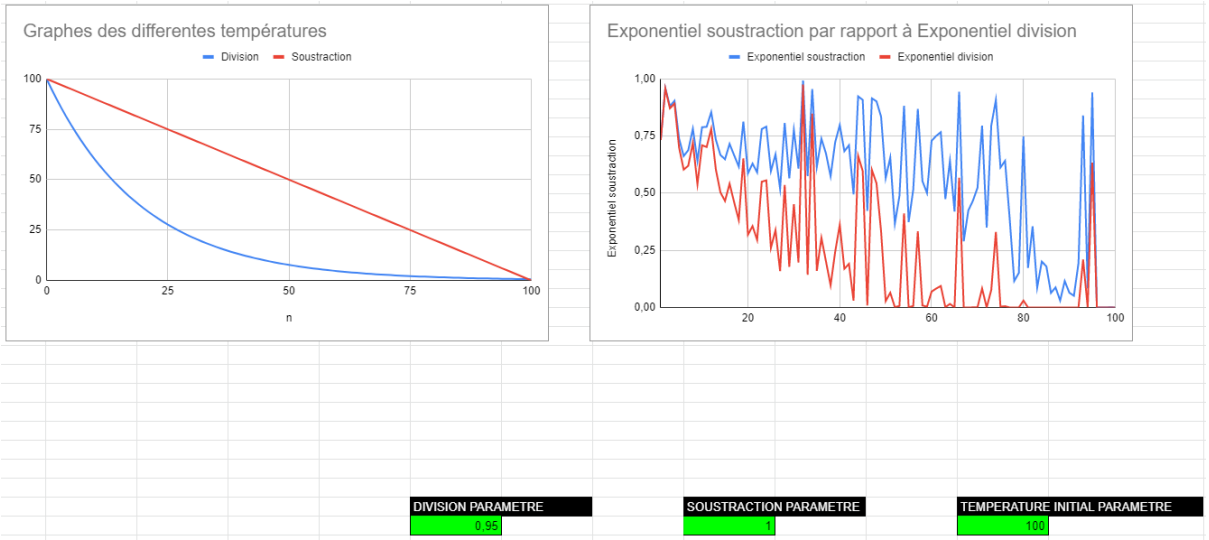
II. Solution mise en oeuvre

Ici nous avons fait le choix d'utiliser un algorithme d'approximation permettant de trouver une solution proche de la solution optimale mais avec l'avantage d'avoir le résultat en un temps raisonnable. De ce fait, ces algorithmes sont performants dans les cas où le nombre de données est très important et où l'on veut seulement un résultat approché. L'algorithme d'approximation choisi ici est un algorithme de recuit simulé, efficace pour contrer les minimums locaux afin d'arriver aux minimums globales.

Le principe ici est de simuler un matériel en fusion ayant une température T, plus T est élevé et plus le matériel a une probabilité élevée de se transformer et à chaque transformation, la température diminue jusqu'à atteindre une température ou toute transformation est impossible. Ici nous avons aussi une température T, une transformation est représenté par l'inversion de deux villes dans un chemin. A chaque transformation on diminue la température. On accepte automatiquement tous les changements dont cela fait baisser le coût de déplacement. Cependant si la transformation donne un coût plus élevé alors on consulte la température actuelle, puis selon si l'augmentation est élevée et si la température actuelle est faible, alors on définit une probabilité d'accepter ce changement. Lorsque la température minimum est rencontrée on affiche le meilleur chemin rencontré parmi tous ces changements.

III. Comparaison stratégies

Il y a ici plusieurs façons de faire varier le programme. La première est de faire varier la façon dont la température diminue, dans mon cas j'ai essayé de comparer deux stratégies. Une ou l'on fait diminuer de façon constante en retirant à T un entier et une autre façon ou l'on fait diminuer en multipliant la température actuelle par 0.95. N'arrivant pas vraiment à me rendre compte de la différence que cela pourrait apporter, j'ai décidé d'essayer de représenter la différence de façon graphique à l'aide de deux graphes, fait sur Google Sheets.



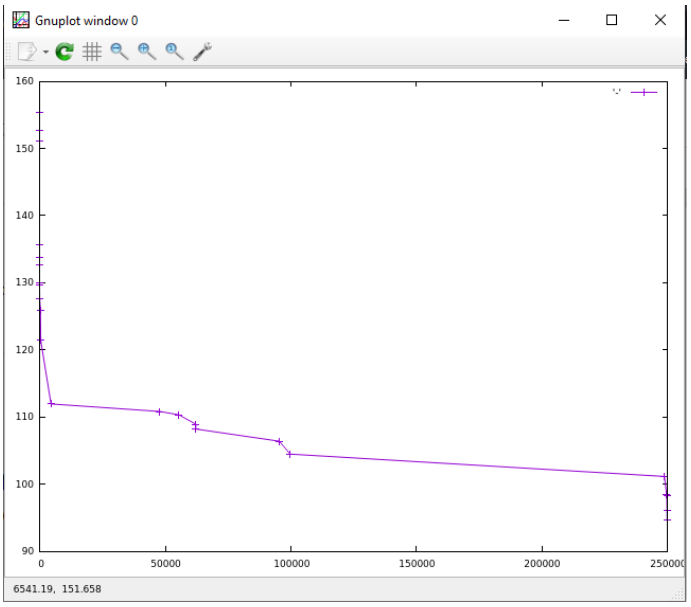
J'ai alors modélisé les 100 premières températures pour la fonction qui diminue la température par soustraction et pour la fonction qui diminue la température par multiplication. J'ai aussi ajouté un paramètre pour définir la température de départ. Afin de représenter la différence de coût entre deux valeurs, j'ai fait tourner mon programme en utilisant la multiplication et j'ai regardé toutes les différences de coûts. Comme on fait des inversions de villes de façon aléatoires alors les coûts sont eux aussi plus ou moins aléatoires et donc la différence l'est aussi. J'ai alors regardé la valeur la plus haute et la valeur la plus faible : 50, 0. J'ai alors créé pour chaque température un nombre aléatoire entre 0 et 50 pour créer la fonction qui permet d'afficher le graphe de droite et donc de savoir approximativement les chances de garder un coût moins bon. Bien sûr ces valeurs sont arbitraires mais je pense qu'elles permettent de se donner une idée de ce qu'il se passe.
Grâce à cela j'ai pu constater qu'avec une réduction par multiplication, il est impossible de passer en dessous de 0 ce qui est explicable car un nombre compris entre 0 et 1 multiplié par un autre tous deux différents de 0 ne donnera jamais 0. J'ai pu aussi faire varier les paramètres mais nous verrons ça dans la section V.

L'autre façon de faire varier le programme est de changer la façon dont on transforme le programme mais ayant déjà passé du temps à tester les différentes façon dans le précédent TP, j'ai décidé de garder celle qui m'était le plus efficace, c'est à dire celle ou l'on inverse deux villes dans un chemin.

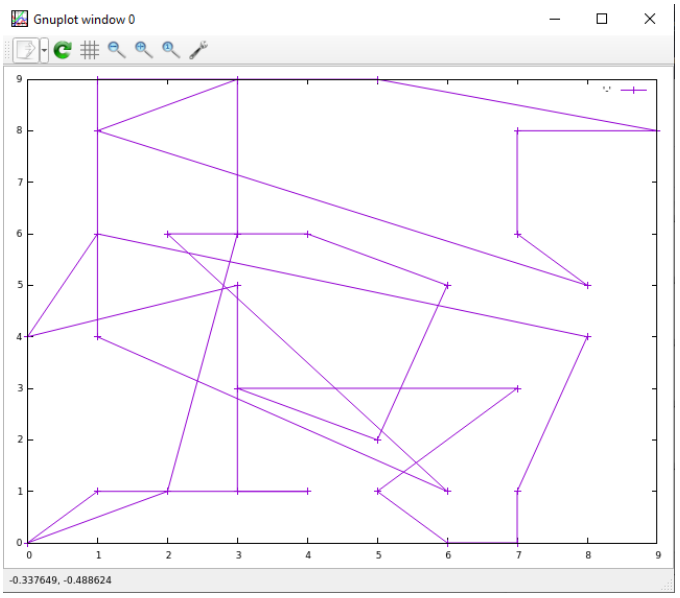
IV. Analyses résultats

Pour tester les programmes dans les meilleures conditions, j'ai mis les paramètres qui devraient donner les meilleurs résultats, c'est-à -dire ceux pour lesquels on fait tourner le programme pendant un long moment.

A) Soustraction



Coût du programme de soustraction



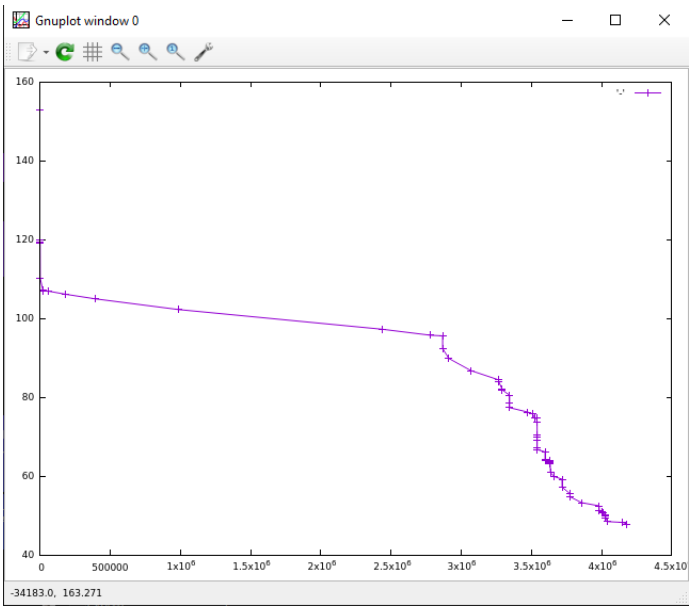
Représentation graphique du meilleur chemin trouvé

Pour le programme par soustraction, avec les paramètres suivants :

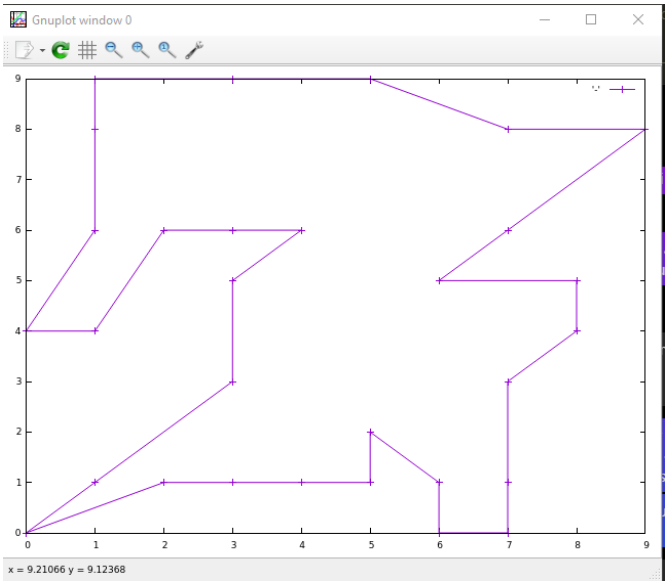
- Température initial 1000
- tf : 1
- amplitude : 3
- alpha : 1
- Max répétition : 250

Le résultat est de 96, ce qui n'est pas bon.

B) Multiplication



Graphe des coûts du programme par multiplication



Graphe de l'itinéraire avec le programme par multiplication

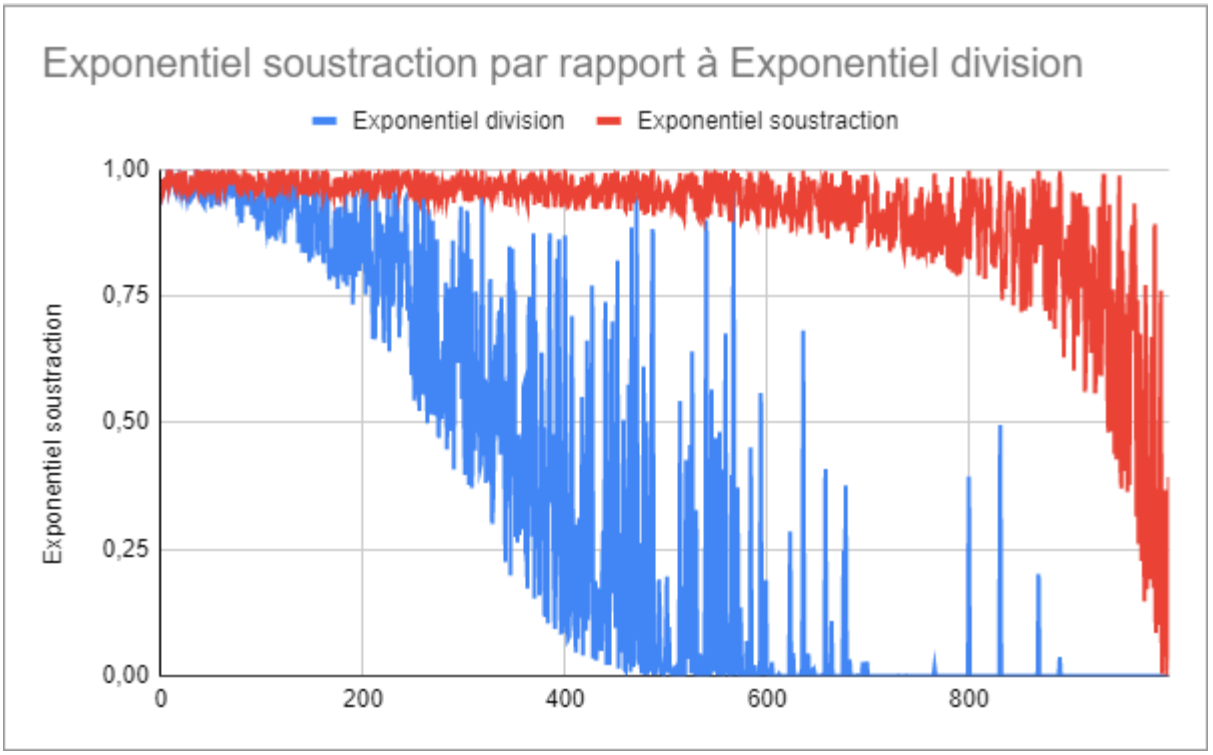
Pour le programme par multiplication avec les paramètres suivants :

- Température initial : 2000
- Température finale : 0.001
- amplitude : 15
- alpha : 0.99
- Max répétition : 5000

J'obtiens un résultat de 47 ce qui semble être un des meilleurs résultat possible au vu du graphe.

C) Explication

J'ai fait exprès de mettre des paramètres extrêmes pour être sûrs d'avoir la meilleur valeur dans les deux programmes. Je pense que la différence peut s'expliquer en regardant le graphe que j'ai fait sur le Google Sheet.



Graphe des probabilités d'acceptation.

Dans ce graphe, j'ai mis une température initiale à 1000 et une décroissance de 1 pour la soustraction et une multiplication par 0.99. Quelque soit les valeurs que je mets, je remarque que lors de la soustraction, la probabilité d'accepter une température moins bonne chute d'un coût alors que la multiplication diminue plus progressivement, ce qui permet au programme d'atteindre plus facilement les minimums locaux. Contrairement à la soustraction ou le programme accepte tout jusqu'à un point ou il n'accepte presque plus rien ce qui fait que le programme peut ne jamais rencontrer de minimum local et donc avoir un coût bien supérieur.

V. Analyse Paramètres

Dans ce programme il y a cinq paramètres : Température initiale, température finale, alpha, amplitude et max répétition. Ces cinq paramètres sont liés, c'est-à-dire qu'avec une différente combinaison de valeurs on peut obtenir un même résultat.

Pour ce qui est de la température initiale, j'ai remarqué que plus la température initiale est haute, plus il y a de grandes chances d'accepter des coûts moins élevés au cours du programme.

En ce qui concerne la température finale, plus elle est faible et plus la chance d'accepter un coût moins bon sur la fin est faible. A savoir qu'un programme qui utilise une multiplication ne peut pas avoir une température négative et que dès lors qu'un programme qui utilise une soustraction arrive à une température négative, il n'acceptera plus les coûts moins bon ce qui fait qu'il ira se coincer à un minimum local et aura de grande chance d'y rester coincer.

Ensuite alpha, il permet de choisir à quelle vitesse la température diminue, en sachant que pour le programme utilisant la multiplication, changer la température initiale ne fera pas changer la durée, il faut changer la température finale, l'alpha ou le nombre max de répétition. Contrairement à la soustraction où il suffit de changer n'importe lequel des paramètres excepté l'amplification pour changer la durée du programme.

Le nombre de répétitions max permet de définir combien d'essais on donne au programme pour atteindre un nouveau coût avant de changer la température.

Et enfin nous avons l'amplification qui permet de définir à quelle point on s'éloigne de la solution actuelle lorsqu'on la transforme. Il ne faut pas mettre une amplification trop élevée car on risquerait de changer de minimum local avant de l'avoir atteint, en sachant que lorsqu'on est coincé dans un minimum local c'est grâce à la température qu'on est censé réussir à en sortir et non grâce à l'amplification.

D'après les essais que j'ai fait, si l'on veut obtenir un résultat régulier il faut que le programme tourne assez longtemps ce qui n'est pas un problème car il est très rapide.

Il faut aussi savoir que plus le nombre de villes est important et plus les paramètres doivent être choisis judicieusement.

VI. Problème rencontrés

- Un des plus grands problèmes que j'ai eu est le fait que je suis tombé malade le jour où l'on avait deux tps mais je pense avoir réussi à rattraper toutes les informations qu'il me manquait.
- J'ai aussi eu un problème de pointeurs en ayant fait

```
cheminY = &cheminX
```

Au lieu de

```
 *cheminY = cheminX;
```