

RAPPORT PROJET OPTIMISATION ET RECHERCHE OPÉRATIONNELLE

A propos de ce projet j'ai pu implémenter la méthode des potentiels puis l'algorithme du simplexe.

Méthode des potentiels

Par rapport à la méthode des potentiels, nous avons un fichier d'entrée qui traite un problème avec 9 tâches, chaque tâche étant affectée avec sa durée, ses antécédents et son nombre d'antécédents. Ainsi j'ai commencé par implémenter une fonction pour lire le fichier d'entrée donc la fonction `read_data()`. Pour cela j'ai défini une structure de données avec les paramètres cités ci-dessus. Ainsi on stocke les données du fichier d'entrée dans les variables de la structure de données. Puis on passe à l'implémentation des fonctions pour calculer les dates au plus tôt et au plus tard, la marge totale et la marge libre. Pour cela on applique juste les formules vues en cours. Cependant mon implémentation est beaucoup basée sur l'exemple que j'essaie de traiter, de ce point de vue mon code n'est pas infaillible et ne résout pas tous les problèmes de ce genre en l'occurrence il ne prend en compte que le type de fichier d'entrée que j'ai utilisé.

Méthode du simplexe

Par rapport à la méthode du simplexe, nous avons également un fichier d'entrée qui traite un problème de maximisation. Dans le fichier d'entrée, on a les contraintes et pour chaque contrainte on a des variables x, y et les bornes de chaque équation ainsi que la fonction objective qui n'a pas de borne. Comme précédemment j'ai fait une fonction `read_data()` pour lire le fichier d'entrée. Pour cela j'ai fait une structure de données pour stocker les données dans les variables de la structure. Puis j'ai fait une implémentation de la fonction de maximisation du problème de simplexe. Cette implémentation est une méthode itérative qui se réitère tant qu'on a des valeurs strictement positives dans la fonction objective.

Dans mon algo je commence par former un tableau avec une contrainte à chaque ligne et la fonction objective dans la dernière ligne.

Cette implémentation prend en compte que le type de fichier décrit plus haut. On peut le rendre plus scalable de ce point de vue. Pour les détails d'implémentation, tout est commenté dans le code (voir code).

Bug identifié :

A la fin j'arrive pas à avoir toutes les valeurs de la fonction objective qui deviennent négatives i.e l'algorithme n'est pas réellement fini puisque j'ai encore une valeur positive dans la fonction objective donc potentiellement un max. C'est sur cette ligne de code que je dois encore bosser

```
if((func_objective[s->nb_iter] > 0) ){
```

```
    maximisation(s);
```

```
    printf("\n");
```

```
}
```

J'ai essayé de faire une boucle `for` et aussi une boucle `while` pour corriger le fait que j'ai encore des valeurs positives sur la fonction objective mais j'ai des erreurs de segmentation sûrement un problème de mémoire cpu.