VIỆN QUỐC TẾ PHÁP NGỮ
Institut Francophone International

Rapport de TP

SUJET : Implémentation de l'algorithme de Ford-Fulkerson sur le Problème de Flot-Max

COURS DE RECHERCHE OPPERATIONNELLE

TRESENTATION : BARRY Mamadou Dian / kaoudian@hotmail.fr
KAFANDO Rodrique / rdiuskaf@gmail.com

Promotion 21, System Intelligent et MultiMedia

Introduction

La recherche opérationnelle peut être définie comme l'ensemble des méthodes et techniques rationnelles orientées vers la recherche du meilleur choix; la façon d'opérer en vue d'aboutir au résultat visé ou au meilleur résultat possible. L'objet de cette discipline est de fournir des bases rationnelles à la prise de décisions, habituellement dans le but de contrôler ou d'optimiser (améliorer l'efficacité, diminuer les coûts, etc.). Elle fait partie des «aides à la décision» (WIKIPEDIA).

L'algorithme de Ford-Fulkerson utilisé en recherche opérationnelle permet de résoudre des problèmes d'optimisation pouvant être représentés sous forme de graphe comportant une source et un puits. Notre travail consistera tout d'abord à expliquer sous quelles hypothèses l'algorithme et notre programme sont en concordance, Ensuite, nous donnerons les effets de ces hypothèses, et enfin nous verrons dans quelles mesure notre programme afin de pouvoir vérifier ces hypothèses ou encore mieux de pouvoir traiter le problème dans un cas générale.

Problématique :

L'algorithme de Ford-Fulkerson, est un algorithme pour le problème du flot maximum, c'est-à-dire un problème d'optimisation classique dans le domaine de la recherche opérationnelle. Ce problème d'optimisation peut être représenté par un graphe ayant une entrée (source) et une sortie (puits). Le flot représente la capacité d'un chemin (arc) de l'entrée vers la sortie, c'est d'ailleurs la raison de l'utilisation de cet algorithme dans les problèmes de réseaux. Les domaines d'applications de cet algorithme sont multiples : **problèmes informatiques, routiers, ferroviaires** pour ne citer que ceux-ci.

Soit le graphe définit par **G=(S, P)**, Ford-Fulkerson est basé sur les graphes orientés dont chaque arrête est valuée par une capacité, un sommet source et sommet Puits.

Voir Fig.grahe

Le problème à résoudre revient à répondre à la question de savoir :

Quel est le flot maximum qu'il est possible de faire passer dans ce réseau depuis la source vers le puits ?

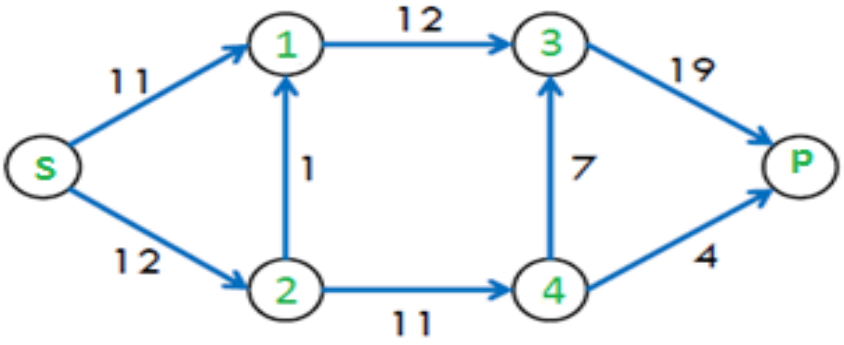

Fig. graphe

Principe :

Le principe de fonctionnement de cet algorithme est principalement basé sur la recherche de flot maximum dans un réseau.

Ici On dispose d'une source, d'un puits et le flot doit s'écouler de la source vers le puits et il doit être maximum en fonction de la capacité des arcs.

- **Déterminer le Flot augmentant**

Le principe de l'algorithme est le suivant:

Tant Que

Il existe un chemin augmentant de s vers p **Faire** mettre à jour le graphe résiduel chercher un nouveau chemin augmentant

FinTantQue ;

- **Mise à jour la capacité résiduelle des arcs**

Soit B un chemin augmentant Soit Bo la capacité résiduelle de ce chemin.

Pour

Tous les arcs (si, sj) appartenant au chemin **Faire** mettre à jour la capacité résiduelle de l'arc

Fin Pour

- **Recherche des voisins de la source non visités dont les arcs ne sont pas saturés**

Tant Que

Il reste des voisins non visités ET chemin pas trouvé ; **Faire**

Choisir un voisin pour poursuivre le chemin augmentant

Si ce voisin est le puits **Alors**

Le chemin construit est le chemin augmentant ainsi le flot valider mettre à jour le graphe résiduel retour au départ pour la recherche d'un nouveau chemin,

finSi

FinTantQue

- **Afficher le flot max**

Le flot maximum est affiché lorsqu'il n'existe plus de chemin augmentant de la source vers le puits. Cela signifie que nous avons marqué tous les sommets à partir desquels aucun nouveau chemin augmentant n'est possible.

Mode de fonctionnement du Programme

Notre algorithme fonctionne à base de fonctions recevant des paramètres à partir desquelles il s'exécute et affiche des résultats. Nous expliquons quelques-unes de ces fonctions en exemple.

- **min** : nous utilisons cette fonction qui compare et retourne le minimum entre deux nœuds,
- **enqueue** : il parcourt les nœuds et colore en gris chaque nœud déjà parcouru, cela permet de savoir qu'un nœud est visité au moins une fois.
- **dequeue= ()**: permet de marquer tous les nœuds tête de chaque arrête en noir.
- **bfs (start, target):** permet de parcourir le graphe, de vérifier les capacités de chaque arrête.
- **max_flow (source, target)** : initialise le flot max, recherche les chemins augmentant tout en incrémentant le flot le long de ces chemins.

- Détermine l’arrête par laquelle une augmentation du flux est possible et incrémente la valeur du flot. Elle répète cette itération jusqu’à finir les chemins augmentant possible, ainsi il retourne la valeur du flot max.
- **read_input_file ()**: cette fonction permet de lire la matrice contenu dans le fichier « fichier.txt »

Apres l’expérimentation de notre programme et les tests effectués sur ce programme nous pouvons affirmer qu’il concorde avec le fonctionnement logique de Ford-Fulkerson. Cependant, il reste à améliorer car en ce qui concerne l'insertion du fichier matrice nous n’avons pas pu faire appel à une matrice carré.

Ce programme peut bien être améliore car des limites y figures notamment le format du fichier. Nous avons entre autre le respect de l’ordre de saisie de la matrice. Ces améliorations permettront la résolution des problèmes à grandes échelles.

Difficultés

Principalement le problème majeur que nous avons rencontrés réside au niveau de l’intégration du fichier contenant les matrices carré dans le code afin que le programme l’appel automatiquement.

Conclusion

Conformément a l'algorithme de ford-fulkerson, notre programme respect l'ensemble des itérations permettant d'obtenir de meilleur résultats.
Ce travail nous a permis de mieux comprendre le fonctionnement de l’algorithme de Ford-Fulkerson.

Retenons que cet algorithme est très populaire de par sa capacité de résoudre les problèmes les plus pertinents et dans tous les domaines.