DM de Graphes

Cédric Bevilacqua – Camille Billouard

Question 1 - Qu’est-ce qu’un flot maximal ?

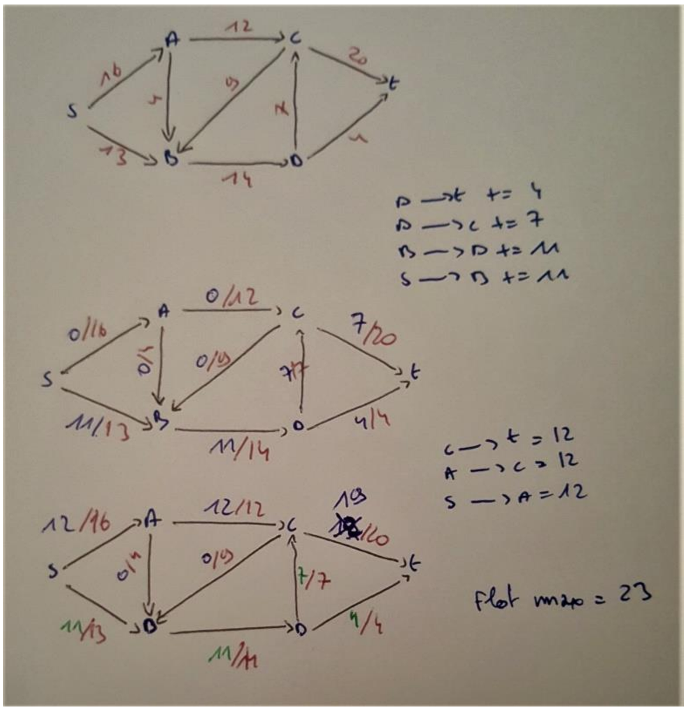
Le problème de flot maximal consiste à transporter la quantité maximale possible d’une origine (source) à une destination (puits) donnée, sans dépasser les capacités des arcs. Le flot représente la quantité d’unité que l’on transporte de la source s au puits t.

On souhaite maximiser la quantité d’unité acheminée. On essaie donc d’obtenir un flot maximum, sachant que les arc (ou canaux) du réseau sont caractérisés chacun par une capacité que l’on ne peut dépasser.

Exemples :

- Les canalisations entre les châteaux d’eau et les villes ont des débits limités.
- Réseau routier d’un département où chaque route a un débit de circulation limité.
- Réseau internet où chaque backbone dispose d’un débit limité.

Question 2 - Calculer `a la main, sans essayer d’appliquer un algorithme, un flot maximal pour l’exemple 42 du polycopié.



Question 3 - Quelles sont les informations à stocker pour modéliser un réseau ?

La modélisation d’un réseau de flot est représentée sous la forme d’un graphe pondéré orienté qui implique de connaître :

- La **source** et le sommet de départ
- Le **puits** ou le sommet d’arrivé.
- L’ensemble des arcs ou **arêtes pondérées** par leur capacité.
- L’ensemble des **sommets** qui vont interconnecter les arcs entre eux.

Question 4 - Qu’est-ce qu’un chemin augmentant (on dit aussi chaîne améliorante ou chemin améliorant) ?

Étant donné un réseau de transport G = (S, A) et un flux f, un chemin améliorant p est un chemin de la source à un puits dans le réseau résiduel. C’est un chemin partant de la source jusqu’au puit et empruntant des arêtes du réseau de transport.
Selon la définition du réseau résiduel, chaque arc (u, v) d'un chemin améliorant possède un flux positif supplémentaire de u à v tout en restant soumis à la contrainte de capacité sur cet arc.
La capacité de chaque arc à une capacité strictement supérieur à zéro et qui n’est pas encore saturé. Pour le calcul du flot maximal l’algorithme choisit cherche un chemin augmentant dans le graphe résiduel. Il sature ce chemin s’il existe, sinon il retourne le flot maximum.

Question 5 - Qu’est-ce qu’un réseau résiduel ? A quoi cela sert-il ?

Le graphique résiduel d'un réseau de flux est un graphe qui indique les flux supplémentaires possibles. S'il existe un chemin de la source au puits dans le graphique résiduel, il est alors possible d'ajouter un flux. Chaque arc d'un graphe résiduel a une valeur appelée capacité résiduelle qui est égale à la capacité initiale de l'arc moins le débit actuel. La capacité résiduelle est essentiellement la capacité actuelle de l'arc.

Question 6 - Quels sont les algorithmes de calcul de flot disponibles dans NetworkX ?

Voici les principaux algorithmes de calcul de flot disponibles dans NetworkX :

- Edmonds−KarpEdmonds−Karp : O(nm2) O(nm2)
- Shortest Augmenting PathShortest Augmenting Path : O(n2m) O(n2m)
- Preflow−PushPreflow−Push : O(n2m−−√) O(n2m)
- DinitzDinitz : O(n2m) O(n2m)
- Boycov−KolmogorovBoycov−Kolmogorov : O(n2m||C||) O(n2m|C|)

Chacun de ces algorithmes retourne un graphe résiduel à l’issu de son exécution.

Question 7 - Quelles modifications doivent-être faites dans le réseau lorsqu’on augmente la capacité d’un arc ?

Lors de l’augmentation d’un arc nous devons d’une part mettre à jour la capacité de l’arc du graphe de « base » et celle du graphe résiduel pour permettre par la suite de calculer le flot max à partir du graphe résiduel.

Question 8 - Détailler l’algorithme que vous adoptez pour l’approche fine du recalcul du flot maximal lorsqu’on augmente la capacite d’un arc.

Avant de détailler notre approche pour le recalcul du flot maximal après l’augmentation de la capacité d’un arc, nous allons faire un bref rappel sur l’algorithme de Edmonds Karps qui a servi d’inspiration pour notre approche plus “fine”.

Cet algorithme prend en entrée un Graphe et le sommet d’entrée (la source) et le sommet de sortie (le puits).
A l’initialisation le graphe résiduel est égal au graphe passé en paramètre et tous ses flots sont mis à 0.
Ensuite on effectue un parcours en largeur du graphe résiduel et tant qu’un chemin améliorant est possible on augmente le flot le long de ce chemin d’une quantité égale à la capacité résiduelle du chemin parmi les plus courts chemins de la source au puits. Une fois les chemins saturés on renvoie le flot max. C’est algorithme se fait en $O(|S|.|A|^2)$.

Le problème posé, lorsque la valeur d’un arc est augmentée, nous renvoie aux dernières itérations de la boucle, “tant qu’un chemin améliorant est possible”, sur le graphe résiduel. Or, au moment du premier calcul du flot max, nous avons conservé le graphe résiduel et la valeur du flot max. Il ne sera donc pas nécessaire de mettre à zéro tous les flots et de repasser par les chemins déjà saturés, mais de se concentrer directement sur les chemins non saturés. Il suffit d’exécuter l’algorithme de Ford-Fulkerson à partir de ce graphe résiduel. Si nous trouvons un chemin augmentant p, on retourne l’ancien flot max plus le flot trouvé. Si nous ne trouvons pas de chemin d'augmentation, on retourne le flot max précédent.

De plus, nous ne recalculons le flot maximal seulement si l’arc est saturé car il n’a aucune chance d’augmenter s’il n’est pas déjà saturé, dans le cas contraire nous mettons seulement les capacités à jour des deux graphes (base et résiduel).

Question 9 - Proposer une analyse de la complexité théorique de l'approche proposée.

Comme nous ne recalculons le flot maximal seulement si l'arc est saturé, s'il ne l'est pas, la complexité dans le meilleur des cas sera atteinte en O(|A|).

Ensuite, il ne sera question que d'arc « saturé » et nous allons distinguer deux cas :

(a) Augmentation de la capacité d'une seule unité
(b) Augmentation de la capacité de plus d'unité

(a) - A partir du précédent flot max, il suffit de faire *k* nouvelles itérations de l'algorithme de Ford-Fulkerson avec le graphe modifié. Le graphe de flux résiduel augmente la capacité de l'arc (u, v) de *k*. On fais une recherche sur le graphe, en O(|V | + |E|), pour voir s'il y a un chemin dans le graphe de flux résiduel le long duquel le flux peut augmenter. S'il y en a un, il doit y avoir un flot de taille un. Et comme chaque itération augmente le débit max d'au moins 1, le temps de fonctionnement total sera O(k|A|), pour k égale au nombre d'unité de capacité en plus et A le nombre d'arête.

S'il n'y a pas de flux dans le graphique de flux résiduel, l'ancien flux max est toujours le flux maximum.

Pour nous convaincre que la nouvelle approche était plus « fine » nous avons fait une série de test :

Nous avons, pendant 10 000 itérations, aléatoirement choisit d'augmenté la capacité d'arc appartenant au graphe de *k* unités avec l'approche où le flot est recalculé à partir d'un nouveau graph résiduel (A) et avec notre approche (B) qui recalcul à partir du graphe résiduel.

(A) `Temps = 2.247214412689209`
(B) `Temps = 0.19431426525115966`

Nous constatons un gain significatif dix fois plus rapide.

Q 10 - Quelles modifications doivent-être faites dans le réseau lorsqu'on réduit la capacite d'un arc ?

Lors de l'augmentation d'un arc nous devons d'une part mettre à jour la capacité de l'arc du graphe de « base » et celle du graphe résiduel pour permettre par la suite de calculer le flot max à partir du graphe résiduel.

Q 11 - Détailler l’algorithme que vous adoptez pour l’approche fine du recalcul du flot maximal lorsqu’on diminue la capacité d’un arc.

Notre algorithme augmente d’une unité seulement. Donc dans le cas d’une augmentation $k > 1$, alors nous faisons k itérations.

Nous allons distinguer deux cas :

(a) L’arc choisit n’est pas saturé f (u, v) < c (u, v))
(b) L’arc est saturé f (u, v) = c (u, v))

(a)- Si le flot de l’arc du graphe avant la mise à jour n’est pas saturé alors il n’y a pas de chance qu’il augmente, donc nous retournons l’ancien flot max.

(b)- Dans le cas où l’arc est saturé :
Pour connaître le chemin de concerné par la modification on trouve le chemin p1 dans le graphe résiduel de u à s. Et on trouve le chemin p2 dans le graphe résiduel de t à v.
Ensuite, on prend le chemin p2,(v, u), p1 dans le graphe résiduel, et décrémente d’une unité de flux de t à s le long de ce chemin. On soustrait au flot max précédent d’une unité.
Enfin, on effectue une recherche dans le graphe pour trouver un chemin d'augmentation.
Si nous trouvons un chemin augmentant, nous retournons le nouveau flot max.

Le temps de fonctionnement de l'algorithme est $O(|V| + |E|)$ k fois donc une complexité totale de $k*O(|V| + |E|)$.

Pour nous convaincre que la nouvelle approche était plus « fine » nous avons fait une série de test :

Nous avons, pendant 10 000 itérations, aléatoirement choisit d’augmenté la capacité d’arc appartenant au graphe de k unités avec l’approche où le flot est recalculé à partir d’un nouveau graph résiduel (A) et avec notre approche (B) qui recalcul à partir du graphe résiduel.

(A) `Temps = 2.6872247457504272`
(B) `Temps = 0.25634539127349854`

Nous constatons un gain significatif dix fois plus rapide.

Q 12 - Discuter des limites de l’approche que vous proposez

L’approche proposée pour le recalcul du flot max après la diminution de la capacité d’un arc, trouva sa limite à la fin de son exécution. Car nous rappelons à nouveau la méthode calcul du flot à partir du graphe résiduel. Dans le cas ou la mise à jour des capacités des arcs aurait entrainé des erreurs de recalcul du flot, étant donné que nous diminuons le flot de manière égale à la diminution réalisée sur le chemin concerné.