TD Algo2 – session 10 – Flot maximum

28 mai 2025

Objectifs d'apprentissage :
— analyser les propriétés formelles liées au flot dans les réseaux de transport ;
— modéliser des problèmes en problème de flot maximum (ou les y réduire) ;
— concevoir des algorithmes relatifs à ce problème ;

Les 6 premiers exercices sont essentiels. Plus d'exercices sur `https://algs4.cs.princeton.edu/64maxflow/`.

Exercice 1 : Séparation d'arc

Montrer que séparer un arc en deux dans un réseau de transport produit un réseau équivalent. Plus formellement, supposons qu'un réseau de transport G contienne l'arc (u, v). On définit un nouveau réseau de transport G' avec un nouveau sommet x et on y remplace (u, v) par deux arcs (u, x) et (x, v) avec $c(u, x) = c(x, v) = c(u, v)$. Argumenter que le flot maximum de G' a la même valeur que celui de G.

On peut procéder par une preuve par l'absurde, mais il est plus direct ici de montrer que pour tout flot, f', de G', on peut construire un flot, f, de même valeur dans G, et inversement.
Pour le sens $f' \to f$, on peut construire le même flot pour G en remplaçant $f(u, v)$ par $f'(u, x)$ sans violer la contrainte de capacité car $f(u, v) = f'(u, x) \leq c(u, x) = c(u, v)$.
Pour le sens $f \to f'$, on peut construire le même flot pour G' en remplaçant $f(u, v)$ par $f'(u, x)$ sans violer la contrainte de capacité car $f'(u, x) = f'(x, v) = f(u, v) \leq c(u, v) = c(u, x) = c(x, v)$.

Exercice 2 : Somme des flots

Étant donné un réseau de transport $G = (V, E)$, soient f_1 et f_2 deux fonctions de $V \times V$ vers $\mathbb{R}$. La somme des flots $f_1 + f_2$ est la fonction de $V \times V$ vers $\mathbb{R}$ définie par $(f_1 + f_2)(u, v) = f_1(u, v) + f_2(u, v)$ pour tout $u, v \in V$. Si f_1 et f_2 sont des flots de G, laquelle des deux propriétés de flot la somme de flots $f_1 + f_2$ doit-elle satisfaire et laquelle pourrait-elle violer ?

— Contrainte de capacité : pas de garantie.
— Conservation du flot : $\sum_{u \in V} f_1(u, x) = \sum_{u \in V} f_1(x, u)$ et $\sum_{u \in V} f_2(u, x) = \sum_{u \in V} f_2(x, u)$. Donc $\sum_{u \in V} f_1(u, x) + f_2(u, x) = \sum_{u \in V} f_1(x, u) f_2(x, u)$.

Exercice 3 : Produit scalaire des flots

Soit f un flot dans un réseau et soit α un nombre réel. Le produit scalaire des flots αf est une fonction de $V \times V$ vers $\mathbb{R}$ définie par $(\alpha f)(u, v) = \alpha f(u, v)$. Montrer que, si f_1 et f_2 sont deux flots, alors $\alpha f_1 + (1 - \alpha) f_2$ est aussi un flot pour tout α dans l'intervalle $0 \leq \alpha \leq 1$.

Comme f_1 et f_2 sont des flots, ils vérifient la contrainte de capacité et $O \leq \alpha f_1(u, v) + (1 - \alpha) f_2(u, v) \leq \alpha c(u, v) + (1 - \alpha) c(u, v) = c(u, v)$. Le nouveau flot vérifie la contrainte de capacité. D'autre part, f_1 et f_2 vérifient la contrainte de conservation de flot. Donc pour tout sommet u sauf la source et le puits :

$$\sum_{v \in V} \alpha f_1(v, u) + (1 - \alpha) f_2(v, u) = \alpha \sum_{v \in V} f_1(v, u) + (1 - \alpha) \sum_{v \in V} f_2(v, u)$$

$$\sum_{v\in V} \alpha f_1(v,u) + (1-\alpha) f_2(v,u) = \alpha \sum_{v\in V} f_1(u,v) + (1-\alpha) \sum_{v\in V} f_2(u,v)$$
$$\sum_{v\in V} \alpha f_1(v,u) + (1-\alpha) f_2(v,u) = \sum_{v\in V} \alpha f_1(u,v) + (1-\alpha) f_2(u,v)$$

Exercice 4 : Capacité sur les sommets

Supposons qu'en plus de la contrainte de capacité sur les arcs, il y ait également une contrainte de capacité sur les sommets. Chaque sommet v aurait une limite $l(v)$ sur la quantité maximum de flot qui peut traverser v. Montrer comment transformer un réseau de transport $G = (V, E)$ avec des capacités sur les sommets en un réseau de transport équivalent $G' = (V', E')$ sans capacité sur les sommets et tel que le flot maximum de G' est le même que celui de G. Combien de sommets et d'arcs possède G' ?

On dédoublerait chaque sommet v en deux sommets v_1 et v_2. Tous les arcs entrants sur v pointeraient vers v_1 et tous les arcs sortants de v partiraient de v_2. Il resterait à rajouter un arc entre v_1 et v_2 de capacité $c(v_1, v_2) = l(v)$.
Il y aurait alors $2|V|$ sommets et $|E| + |V|$ arcs.

Exercice 5 : Distanciation

Le professeur Adam a deux enfants qui, hélas, se détestent. Ils se haïssent à tel point que chacun refuse d'aller à l'école en empruntant un trottoir sur lequel l'autre a marché le jour même. Les enfants ne font pas de difficultés si leurs chemins se croisent à un coin de rue. Heureusement, la maison du professeur et l'école sont situées à des coins de rue ; à part cela, le professeur ne sait pas s'il pourra envoyer ses deux enfants à la même école. Le professeur a un plan de la ville. Montrer comment formuler en tant que problème de flot maximum le problème consistant à savoir si les deux enfants pourront aller à la même école.

On modéliserait le plan de la ville en un graphe où les sommets représenterait les intersection et les arcs les rues. Comme une rue a 2 trottoirs, on aurait 2 arcs anti-parallèles entre chaque paire de sommets. La capacité de chaque arc serait $c(u, v) = 1$ pour indiquer le fait qu'il a déjà été emprunté une fois et ne peut plus l'être. La maison est la source et l'école le puits. Si le réseau de transport a un flot maximum supérieur ou égale à 2, alors il existe 2 chemins permettant d'aller de la maison à l'école sans passer 2 fois par la même rue.

Exercice 6 : Coupe minimum

Supposons que l'on souhaite trouver, parmi toutes les coupes minimums dans un réseau de transport G, celle qui contiennent le plus petit nombre d'arcs. Montrer comment modifier les capacités de G pour créer un nouveau réseau de transport G' tel que la coupe minimum soit une coupe minimum avec le plus petit nombre d'arcs dans G.

On conserve les sommets et les arcs avec les capacités $c'(u, v) = |V| \times c(u, v) + 1$. On multiplie la capacité suffisamment pour que la coupe minimum reste valide. Exemple avec 6 sommets s, v_1, v_2, v_3, v_4, t : $c(s, v_1) = c(s, v_2) = c(s, v_3) = 1$, $c(v_1, v_4) = c(v_2, v_4) = c(v_3, v_4) = 10$ et $c(v_4, t) = 3$.

Exercice 7 : Connectivité

La connectivité d'arête d'un graphe non orienté est le nombre minimal k d'arêtes qu'il faut supprimer pour que le graphe ne soit plus connexe. Par exemple, la connectivité d'arête d'un arbre est 1 et celle d'un cycle de sommets est 2. Montrer comment déterminer la connectivité d'arête d'un graphe non orienté $G = (V, E)$ en exécutant un algorithme de flot maximum sur au plus $|V|$ réseaux de transport, comportant chacun $O(V)$ sommets et $O(E)$ arcs.

Il faut créer une version orienté du graphe pour obtenir un réseau de transport dont toutes les capacités sont 1. On sélectionne ensuite un sommet en tant que puits, puis on connecte la source à l'ensemble des autres sommets. On cherche ensuite la valeur minimale de chaque flot maximum obtenu en considérant chaque sommet en tant que puits.

Exercice 8 : Réentrant

Soit un réseau de transport G, où G contient des arcs entrants dans la source s. Soit f un flot dans G avec $|f| \geq 0$ dans lequel l'un des arcs (v, s) entrant dans la source a $f(v, s) = 1$. Prouver qu'il existe un autre flot f' avec $f'(v, s) = 0$ tel que $|f| = |f'|$. Donner un algorithme en temps $O(E)$ pour calculer f', étant donné f.

Since every vertex lies on some path starting from s there must exist a cycle which contains the edge (v, s). Use a depth first search to find such a cycle with no edges of zero flow. Such a cycle must exist since f satisfies conservation of flow. Since the graph is connected this takes $O(E)$. Then decrement the flow 6 of every edge on the cycle by 1. This preserves the value of the flow so it is still maximal. It won't violate the capacity constraint because f > 0 on every edge of the cycle prior to decrementing. Finally, flow conservation isn't violated because we decrement both an incoming and outgoing edge for each vertex on the cycle by the same amount.

Exercice 9 : Chemins améliorants

Montrer qu'on peut toujours trouver un flot maximum dans un réseau $G = (V, E)$ à l'aide d'une séquence d'au plus $|E|$ chemins améliorants. (Conseil : Déterminer les chemins après avoir trouvé le flot maximum.)

Suppose we already have a maximum flow ff. Consider a new graph GG where we set the capacity of edge (u,v) to f(u,v). Run Ford-Fulkerson, with the modification that we remove an edge if its flow reaches its capacity. In other words, if f(u,v)=c(u,v)f(u,v)=c(u,v) then there should be no reverse edge appearing in residual network. This will still produce correct output in our case because we never exceed the actual maximum flow through an edge, so it is never advantageous to cancel flow. The augmenting paths chosen in this modified version of Ford-Fulkerson are precisely the ones we want. There are at most $|E|$ because every augmenting path produces at least one edge whose flow is equal to its capacity, which we set to be the actual flow for the edge in a maximum flow, and our modification prevents us from ever destroying this progress.

Exercice 10 : Flots fixés

Supposons que chaque source s_i d'un problème à sources et puits multiples produise au plus p_i unités de flot, de sorte que $f(s_i, S) = p_i$. Supposons par ailleurs que chaque puits t_j consomme au plus q_j unités de flot, de sorte que $f(V, t_j) = q_j$, où $\sum_i p_i = \sum_j q_j$. Montrer comment convertir le problème consistant à trouver un flot f qui respecte ces contraintes supplémentaires en un problème consistant à trouver un flot maximum dans un réseau de transport à source et puits uniques.

$c(s, s_i) = p_i, c(t_j, t) = q_i, c(t_j, t) = q_j$.

Exercice 11 : Majorant

Soit $G = (V, E)$ un graphe biparti ayant la partition de sommets $S = L \cup R$, et soit G' son réseau de transport correspondant. Donner un bon majorant de la longueur d'un chemin améliorant quelconque trouvé dans G' pendant l'exécution de FORD-FULKERSON.

Exercice 12 : Intégralité

Démontrer le théorème de l'intégralité par récurrence sur le nombre d'itérations.

$\square$