

TD3 - Flot maximum / coupe minimum

GRAPHES ET PROGRAMMATION DYNAMIQUE

M1 Informatique - Semestre 1 - Année 2022-2023

UNIVERSITÉ CÔTE D'AZUR

Christophe Crespelle

`christophe.crespelle@univ-cotedazur.fr`

Le sujet sera traité sur deux séances d'1h30. Il est recommandé que vous utilisiez les exercices que vous n'avez pas eu le temps de faire en séance comme entraînement en vue de l'examen final (ce qui ne veut pas dire que tous les exercices soient du niveau de ceux posés à l'examen, en particulier, les exercices nommés "pour aller plus loin" sont de difficulté supérieure). Les exercices les plus importants sont les 1, 2, 6 et 5 : commencez par ceux-là.

Exercice 1.

Max ou pas max ?

a. Le flot donné sur la Figure 1 - gauche est-il maximum ? Comment le prouver ?

Solution. Non, il ne l'est pas. Pour le prouver, il faut exhiber un chemin augmentant. Pour cela, la clé est de passer par le réseau résiduel (voir figure 1 bas-gauche). Dans le résiduel, on fait un parcours à partir de la source et on voit si on peut atteindre le puits. Dans ce cas oui, il y a donc un chemin augmentant. Un chemin augmentant possible (il peut y en avoir plusieurs) de s à t , avec flot de 1, est s, a, d, b, f, h, t .

b. Le flot donné sur la Figure 1 - droite est-il maximum ? Comment le prouver ?

Solution. Oui, il l'est. Pour le prouver, il faut exhiber une coupe saturée. Pour cela, la clé est la même : passer par le réseau résiduel (voir figure 1 bas-droit). Dans le résiduel, on fait un parcours à partir de la source et si on n'atteint pas le puits, l'ensemble des sommets qu'on peut atteindre définit une demi-coupe, qui est saturée. Dans ce cas, cela donne la demi-coupe $\{s, a, c, d, e\}$ qui est saturée : dans le résiduel, il n'y a que des arcs entrant de l'extérieur vers l'intérieur de cette demi-coupe. Dans le réseau de départ, cela se voit aussi mais moins facilement : tous les arcs qui sortent sont saturés et tous ceux qui entrent ont un flot nul. En général, il peut y avoir d'autres coupes saturées que celle définie par les sommets atteignables à partir de s , elles se voient aussi sur le résiduel.

Exercice 2.

Debit à credit

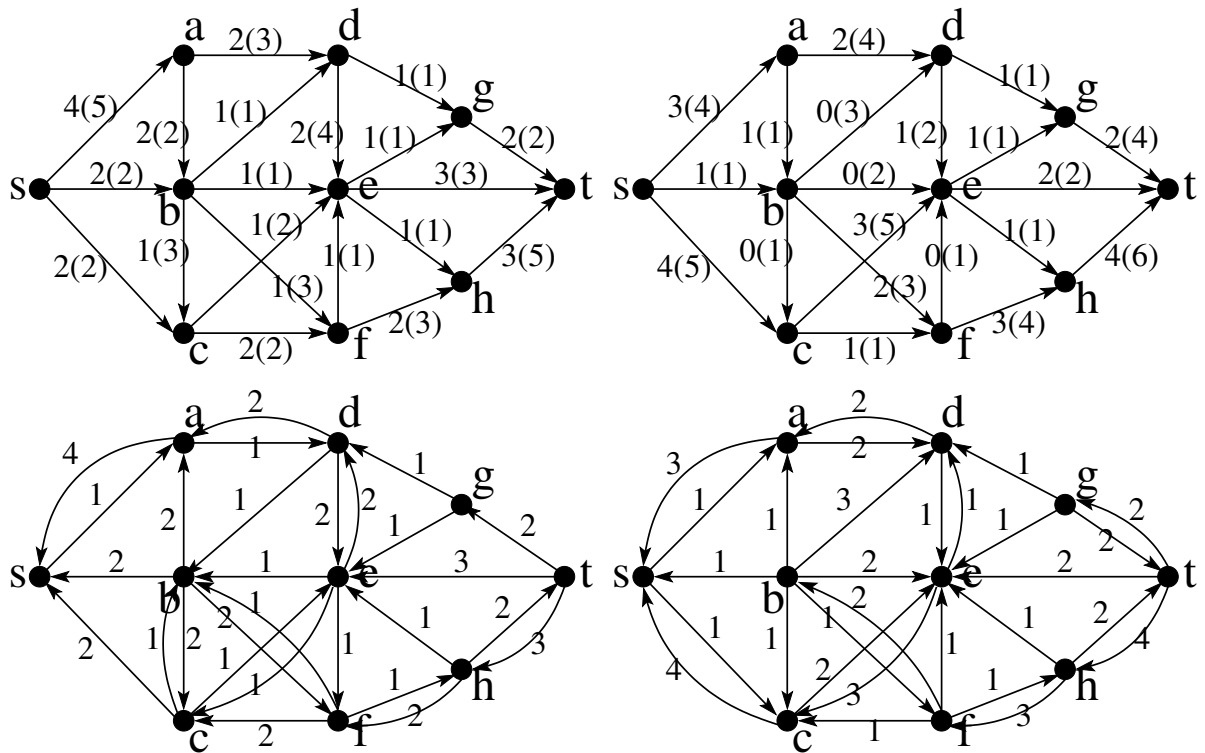


FIGURE 1 – Haut : Deux reseaux de flot, avec source s et puits t , chacun donne avec un flot. Les arcs sont etiquetes par deux nombres $a(b)$: celui entre parentheses, b , est la capacite de l'arc et celui qui precede les parentheses, a , est la valeur du flot a travers cet arc. Exemple : l'etiquette $2(3)$ signifie que la capacite de l'arc est 3 et que la valeur du flot sur cet arc est de 2. Bas : Les deux reseaux residuels correspondant.

Trois villes J, K, L sont alimentees en eau grace a quatre reserves A, B, C, D (nappes souterraines, chateaux d'eau, usines de traitement). Les reserves journalieres disponibles sont de 15 milliers de m^3 pour A, 10 pour B, de 15 pour C et de 15 pour D. Le reseau de distribution, comprenant aussi bien des aqueducs romains que des canalisations recentes, peut etre schematise par le graphe de la figure 2 (les debits maximaux sont indiques sur chaque arc en milliers de m^3 /jour).

Ces trois villes en pleine evolution desirent ameliorer leur reseau d'alimentation afin de satisfaire des besoins futurs plus importants. Une etude a ete faite et a permis de determiner les demandes journalieres maximales probables, a savoir pour la ville J : 15 milliers de m^3 , pour la ville K : 20 et 15 pour la ville L.

- a. Determiner la valeur du flot maximal pouvant passer dans le reseau actuel et donner la coupe minimale correspondante.

Solution.

Tout d'abord, il s'agit ici d'un probleme de flot a sources et puits multiples. Comme vu en cours, on se ramene au cas d'une source et d'un puits uniques en introduisant une nouvelle source s que l'on relie aux anciennes par des arcs sortant de s et un nouveau puits t que l'on relie aux anciens par des arcs entrant dans t . Les capacites de ces arcs sont donnees dans l'enonce : il s'agit pour les sources des capacites de "production"

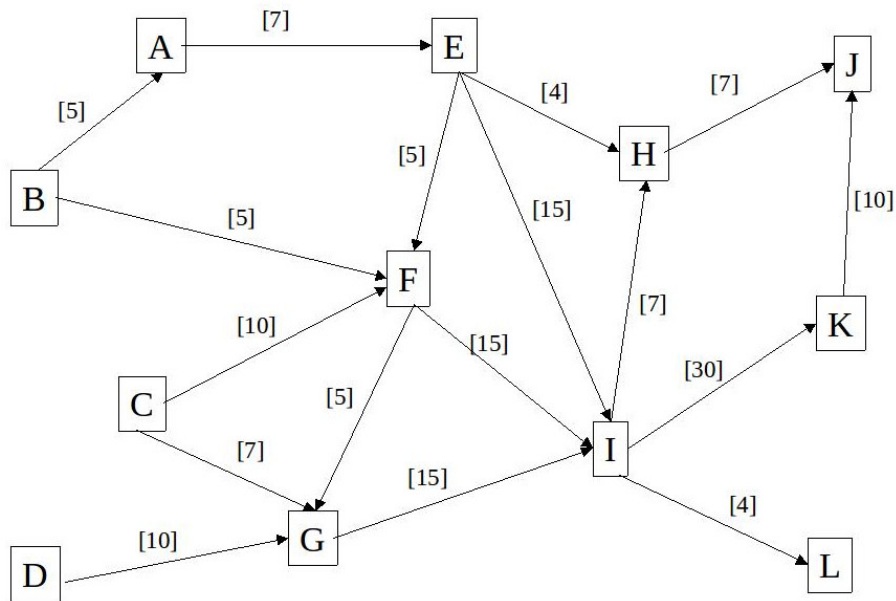


FIGURE 2 – Un réseau d'alimentation en eau.

du flot et pour les puits, des "demandes" en flot. Une fois ces modifications faites, on peut résoudre le problème d'optimisation sur le nouveau graphe formé, qui est alors un problème de flot à source et puits uniques.

L'important ici n'est pas tant la valeur maximale du flot que la méthode pour la trouver et pour être sûr qu'on a bien trouvé le maximum. On peut envisager trois types de méthodes. La troisième qui marche toujours en dernier ressort est d'appliquer l'algorithme de cours, les deux autres permettent parfois de trouver plus vite une solution optimale.

Première méthode : on construit un flot par essais/erreurs/corrections en prenant soin de bien respecter la contrainte de conservativité. Il n'y a aucune garantie que l'on trouve le maximum sauf... si on est capable d'exhiber une coupe saturée ! Si c'est le cas (et que le flot construit est bien conservatif et respecte les capacités des arêtes), c'est gagné. Sinon, on bascule vers une des deux autres méthodes.

Deuxième méthode : on trouve des chemins augmentants directement sur le graphe (sans construire le graphe résiduel). L'intérêt de cette approche est justement d'éviter la construction du graphe résiduel qui peut être fastidieuse. On cherche des chemins dans le graphe original selon lesquels on peut envoyer du flot de la source au puits. On réitère jusqu'à ce qu'on ne voit plus de telle possibilité. Cela ne veut pas dire qu'il n'y en a plus, juste qu'elles ne sont peut-être pas faciles à visualiser directement sur le graphe. Pour être sûr, la encore, une seule méthode : trouver une coupe saturée. Si on en voit une, c'est gagné, sinon on bascule sur la dernière méthode qui marche toujours, celle qui utilise le graphe résiduel.

Troisième méthode : l'algorithme de cours (Edmonds-Karp, ou Ford-Fulkerson si les capacités sont entières). Dans ce cas, ce qui fait qu'on trouve toujours c'est qu'on travaille sur le graphe résiduel, tel que défini en cours, c'est l'objet clé. On forme donc le résiduel du flot et on fait un parcours à partir de la source :

— si on n'atteint pas le puits : gagné ! Le flot est déjà max et maintenant on a une

- coupe saturee a exhiber : la premiere partie est s avec tous les sommets qu'on peut atteindre depuis s dans le reseau residuel, la deuxieme partie est le reste des sommets (qui contient t),
- si on atteint le puits, alors on a un chemin augmentant de la source vers le puit qu'on peut utiliser pour augmenter le flot (on prend un plus court chemin, comme dans EK, si on veut eviter les problemes de terminaison, mais ces problemes n'existent pas lorsque les capacites sont entieres). On actualise le residuel apres le changement du flot et on recommence : soit on trouve une coupe saturee, soit un chemin augmentant, et ainsi de suite.

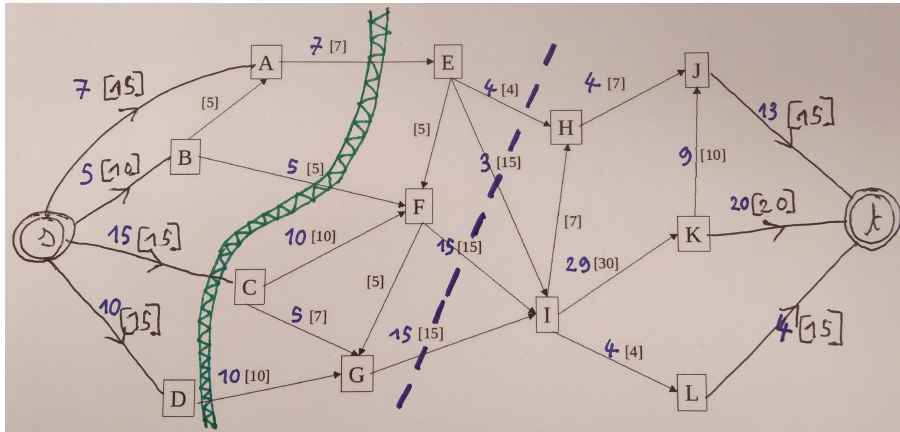


FIGURE 3 – Un flot max, en bleu, avec une coupe saturee, marquee par la separation verte.

Un flot maximum obtenu avec la premiere methode (sans les ratures dues aux essais/erreurs/corrections) est donne sur la Figure 3, avec une coupe saturee ($\{s, A, B, D\}, \{C, E, F, G, H, I, J, K, L, t\}$) qui atteste que le flot est bien maximum. Verifiez qu'il s'agit bien d'un flot, c'est a dire que les capacites des arcs sont respectees et que le flot est conservatif sur **chaque** sommet du graphe. Verifiez aussi que la coupe donnee est bien saturee et qu'elle separe s de t (s, t -coupe).

b. La valeur de ce flot est jugée nettement insuffisante, aussi le conseil intercommunal décide-t-il de refaire les canalisations (A,E) et (I,L). Déterminer les capacités à prévoir pour ces deux canalisations et la valeur du nouveau flot optimal.

Solution. Pour repondre a la question, il est bon de voir d'emblee de combien au plus on peut esperer augmenter le flot (qui est pour l'instant de 37)? On ne souhaite pas l'augmenter au dela de 50 puisque c'est la somme des demandes et qu'il y a, pour cette raison, une coupe ($V \setminus \{t\}, \{t\}$) de capacite 50 dans le graphe, qui ne sera pas modifiee par des changements de canalisations. Il y a meme une coupe assez facile a reperer, la coupe ($\{s, A, B, C, D, E, F, G\}, \{H, I, J, K, L, t\}$) (materialisee par les pointilles bleus sur la figure 3) qui est de capacite 49 et qui n'est traversee par aucune canalisation sur lesquelles des travaux sont envisages (AE et IL). On peut donc esperer augmenter le flot courant d'au plus 12. Cela exige alors d'augmenter la capacite de la coupe saturee qu'on a exhibee d'au moins 12. Faisons cela, c'est a dire augmentons la capacite de AE de 7 a 19, et voyons si on peut augmenter d'autant le flot en jouant aussi sur IL. On

peut bien faire arriver 12 de flot supplémentaire en E en envoyant du flot supplémentaire depuis s : 8 selon l'arc sA et 4 selon le chemin s,B,A, par exemple. Ce flot se retrouve alors en E ou il peut traverser la coupe limitante de capacité 49, par l'arc EI, pour arriver en I. On a alors un flot supplémentaire entrant sur I de 12. Pour augmenter la capacité de IL le moins possible, essayons d'acheminer ce flot supplémentaire de I vers t sans augmenter IL. Sans changer la capacité de IL, on peut envoyer au plus 2 de flot supplémentaire jusqu'à t car le flot traversant la coupe $(V \setminus \{L, t\}, \{L, t\})$ est de 37 alors que la capacité de la coupe est de 39. Et on peut bien en effet acheminer ce flot supplémentaire de 2 de I à t en passant par exemple par le chemin I,H,J,t. Il nous reste donc un flot supplémentaire de 10 en I, que l'on peut tout acheminer vers t par le chemin I,L,t si on augmente la capacité de IL de 10. L'augmentation maximale qu'on peut atteindre pour le flot est donc de 12 (et le flot total de 49) et les augmentations minimales des capacités de AE et IL qu'il faut faire pour y parvenir sont de respectivement 12 et 10.

c. Devant l'importance des travaux, le conseil intercommunal décide de ne pas refaire les deux canalisations en même temps. Dans quel ordre doit-on entreprendre leur réparation de façon à augmenter, après chaque tranche de travaux, la valeur du flot optimal passant dans le réseau ?

Solution. Si on ne commence pas par AE, la coupe dessinée en vert reste saturée et il n'y a pas d'augmentation du flot. Il faut donc commencer par AE.

d. Quelles sont, après chaque tranche de travaux, les valeurs des flots optimaux ?

Solution. Lorsqu'on augmente la capacité de AE de 12 sans modifier celle de IL, on augmente la valeur du flot max de 2, comme explique précédemment. On ne peut pas faire mieux comme en atteste la coupe $(V \setminus \{L, t\}, \{L, t\})$ pour laquelle le flot qui la traverse est seulement inférieur de 2 à sa capacité.

e. Le conseil intercommunal décide finalement d'augmenter les capacités de AE et IL de respectivement 13 et 11. Il souhaite maintenant modifier encore la capacité d'une seule arête pour permettre de satisfaire pleinement la demande des trois villes. Quelles sont toutes les arêtes qui permettraient de réaliser cet objectif ? De combien faut-il augmenter la capacité de l'arête choisie ?

Solution. Il faut forcément modifier la capacité d'une arête de la coupe en pointillés bleus, car elle est saturée après les travaux sur AE et IL et l'augmentation de 12 du flot qui en résulte. N'importe laquelle des quatre arêtes traversantes convient car pour chacune de ces arêtes si on augmente leur capacité de 1, on peut acheminer un flot supplémentaire de 1 de E à I : si on augmente la capacité de EH, en passant par E,H,I (comme il y a un flot de 2 de I à H, qu'on diminue de 1), si on augmente la capacité de EI, directement par EI, si on augmente la capacité de FI, en passant par E,F,I, et si on augmente la capacité de GI, en passant par E,F,G,I.

Exercice 3.

Pour mieux comprendre.

a. Donnez un exemple de réseau et un flot ayant au moins une coupe saturée et au moins une coupe non saturée.

Solution. C'est le cas general pour un flot maximum, toutes les coupes ne sont pas forcement saturees. Un exemple simple a trois sommets et trois arcs, avec le formalisme utilise sur la figure 1 :

s 1(1) t

s 1(1) u

u 1(2) t.

La coupe $(\{s\}, \{u, t\})$ est saturee, la coupe $(\{s, u\}, \{t\})$ ne l'est pas.

b. Donnez un exemple de reseau et un flot dans lequel toutes les coupes sont saturees.

Solution. Cela peut arriver en effet. Un exemple tres simple est celui qui suit, mais on peut en imaginer des plus sophistiques :

— le reseau : un chemin de s a t avec tous les arcs ayant capacite 1

— le flot : flot 1 sur chaque arc.

c. Peut-on toujours augmenter la valeur du flot maximum en augmentant la capacite d'une seule arete ?

Solution. Non, pas toujours. Il suffit qu'il y ait au moins deux coupes saturees qui ne partagent pas d'arc traversant pour que ce ne soit pas possible : en augmentant la valeur d'un seul arc, une des deux coupes reste saturee, donc le flot est encore maximum et n'augmente pas. Exemple simple :

s 1(1) u

s 1(1) v

u 1(1) t.

v 1(1) t.

Remarquez que tous les arcs sont incident soit a s soit a t . Si on augmente la capacite d'un arc incident a s , la coupe $(\{s, u, v\}, \{t\})$ reste saturee, donc le flot maximum est inchangé. Et si on augmente plutot la capacite d'un arc incident a t , la coupe $(\{s\}, \{u, v, t\})$ reste saturee, donc le flot maximum est inchangé, dans tous les cas.

d. Peut-on toujours reduire la valeur du flot maximum en reduisant la capacite d'une seule arete ?

Solution. Oui, il suffit de prendre un arc sortant qui traverse une coupe saturee et de baisser sa capacite. Cela baisse la capacite de la coupe, par definition, et comme le flot maximum est toujours au plus egal a la capacite de n'importe quelle coupe (voir cours) alors la valeur du flot maximum a diminue.

Exercice 4.

Coupes minimum

a. Donnez un exemple de reseau G dans lequel la coupe minimum est aussi petite qu'on veut devant le degre minimum de G .

Indication. Remarquez que dans cette question on parle de coupe quelconque, pas de s, t -coupe.

Solution. Commencez par remarquer que la coupe minimum est toujours au moins egale au degre minimum, car on peut toujours definir une coupe qui isole le sommet de degre minimum. Pour repondre a la question, il faut construire un graphe dans lequel

le degré minimum est grand et la coupe minimum est petite. Un exemple simple est celui de deux cliques (ensemble de sommets tous reliés deux à deux par une arête) à n sommets avec une arête entre les deux cliques. Le degré minimum est $n - 1$, c'est celui des sommets dans la clique, et la coupe minimum est de 1, c'est celle qui met chaque clique d'un côté de la coupe.

b. Donnez un exemple de graphe G et de couple (s, t) pour lequel la s, t -coupe minimum est aussi grande qu'on veut devant la coupe minimum de G .

Solution. Ici, on cherche à faire la distinction entre une coupe du graphe, qui est une bipartition quelconque de ses sommets, et une s, t -coupe qui est une bipartition avec contrainte : s et t ne doivent pas se trouver du même côté de la bipartition (on dit que la coupe *separe* s et t). On peut alors reprendre l'exemple précédent pour répondre à la question. On a dit que la coupe minimum du graphe est 1. Par contre, si on choisit s et t dans la même clique K , toute coupe qui les separe va avoir au moins $n_s \cdot n_t$ arêtes qui la traverse, où n_s est le nombre de sommets de K qui se trouvent du côté de s dans la coupe (y compris s) et n_t est le nombre de sommets de K qui se trouvent du côté de t dans la coupe (y compris t). On a donc $n_s + n_t = n$, avec n fixe. Or les mathématiques nous apprennent que la valeur minimum de $n_s \cdot n_t$ dans ce cas est obtenue pour $n_s = 1$ ou $n_t = 1$. Ainsi, la valeur de la s, t -coupe minimum est au moins $n - 1$, alors que la valeur de la coupe minimum dans tout le graphe est 1.

Exercice 5.

Pour que le courant passe, encore faut-il être au courant

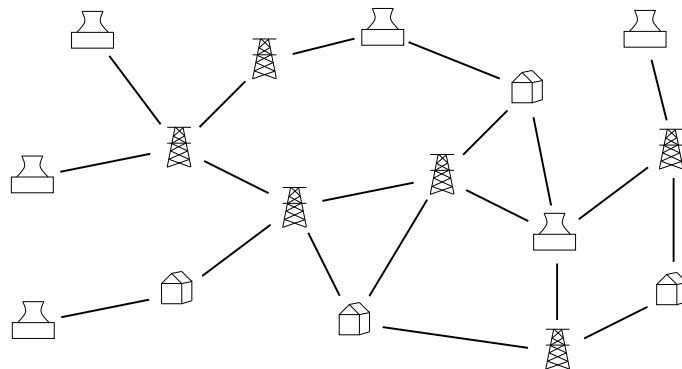


FIGURE 4 – Un réseau électrique. Les usines électriques sont représentées par des pylones haute tension, les sites industriels sont représentés par des cheminées et les autres points d'interconnexion du réseau par des bâtiments.

Un réseau électrique relie un certain nombre d'usines de production électrique (représentées par des pylones haute tension sur la Figure 4) à un certain nombre de sites industriels à alimenter (représentés par des cheminées sur la Figure 4). Le tout est modélisé par un graphe non orienté où usines et sites sont positionnés sur différents sommets, les autres sommets représentant des points d'interconnexion du réseau (représentés par des bâtiments sur la Figure 4). L'électricité peut transiter par tous les sommets et chaque usine

a le pouvoir d'alimenter tous les sites. Donner un algorithme qui calcule le nombre minimum de liens (donc ici d'arêtes) tel que, s'ils sont rompus, plus aucun site n'est alimenté. Analysez la complexité de votre algorithme.

Indication. Modéliser le problème comme un problème de coupe minimum dans un graphe bien choisi que vous formerez, puis utilisez les algorithmes de flot vu en cours pour résoudre ce problème.

Solution. Il s'agit d'un problème qui ressemble à celui de la s, t -coupe minimum mais avec plusieurs sources (les usines électriques) et plusieurs puits (les sites industriels à alimenter). En effet, déconnecter toutes les usines de tous les sites en retirant des arêtes du graphe revient à trouver une coupe qui sépare les usines des sites et à retirer les arêtes qui la traversent. On cherche donc une telle coupe qui ait un nombre minimum d'arêtes qui la traversent. S'il n'y avait qu'une usine et qu'un site, ce serait exactement le problème de la s, t -coupe minimum. Pour prendre en compte la multiplicité des sources et des puits, on peut faire comme en cours pour les flots à sources et puits multiples. C'est à dire qu'on va changer le graphe pour se ramener au cas d'une source et d'un puits uniques. L'idée est d'ajouter un nouveau sommet qui sera la source et qui est relié par une arête à chaque usine, et un autre nouveau sommet qui sera le puits et qui est relié par une arête à chaque site industriel. Il reste ensuite à définir des poids sur chacune des arêtes du nouveau graphe ainsi formé. Toutes les arêtes du réseau original sont de coût de déconnexion identique et reçoivent donc le même poids, disons 1. Le poids que l'on va définir pour les arêtes qu'on a ajoutées entre la source s et les usines et entre le puits t et les sites industriels doit garantir que la s, t -coupe minimum du nouveau graphe formé ne contienne aucune des arêtes incidentes à la nouvelle source et au nouveau puits, de sorte que cette coupe soit la coupe minimum qui sépare les usines des sites dans le graphe original. Pour cela, il suffit de prendre pour les nouvelles arêtes un poids assez grand pour garantir qu'aucune d'entre elles ne traversera la s, t -coupe minimum dans le nouveau graphe. Par exemple on peut leur attribuer un poids de $m + 1$ (ou un poids infini, mais ça pose la question de comment le représenter en machine), où m est le nombre d'arêtes dans le graphe original. Ainsi, la coupe minimum du nouveau graphe n'est traversée que par des arêtes du graphe original car si une seule nouvelle arête la traverse, son poids est d'au moins $m + 1$ alors qu'il existe des coupes de poids au plus m : mettre s et toutes les usines d'un côté, t et tous les sites de l'autre, et les autres sommets du graphe de n'importe quel côté.

Comme aucune nouvelle arête ne traverse la s, t -coupe minimum, toutes les usines sont du côté de s et tous les sites du côté de t , ce qui est ce que l'on cherche. La s, t -coupe minimum est alors exactement la coupe minimum du graphe original qui sépare toutes les usines de tous les sites.

L'algorithme est le suivant :

1. former le nouveau graphe \tilde{G} en ajoutant une source s reliée à toutes les usines avec un poids $m + 1$ et un puits relié à tous les sites avec un poids $m + 1$
2. lancer l'algorithme d'Edmonds-Karp (ou n'importe quel autre algo de flot, ici Ford-Fulkerson marche aussi car les poids sont entiers) pour trouver la valeur du flot maximum dans \tilde{G} , qui est aussi la valeur de la coupe minimum par le théorème du cours

La première étape demande de recopier le graphe en y ajoutant les deux nouveaux sommets et se fait en $O(n+m)$ et l'algorithme d'EK a une complexité de $O(nm^2)$, qui est donc aussi la complexité totale de l'algorithme présentée ici.

Question subsidiaire : comment augmenter l'algorithme pour obtenir non seulement la valeur de la coupe minimum mais aussi une telle coupe ?

Exercice 6.

Gestion de crise

A la suite d'une catastrophe naturelle, les secours ont identifié en différents lieux n individus blessés qui doivent rapidement être évacués vers les hôpitaux de la région. Il y a p hôpitaux en tout, et pour chaque blessé, en fonction de son état et de sa localisation, les options d'évacuations sont réduites à un sous-ensemble fixé d'hôpitaux.

Vous devez organiser l'évacuation en affectant chaque blessé à un hôpital parmi ses options, sans surcharger les services de soins : chaque hôpital h , $1 \leq h \leq p$, peut admettre au plus n_h individus.

Connaissant pour chaque individu la liste des hôpitaux où l'évacuation est possible, donnez un algorithme efficace pour déterminer si tout le monde va pouvoir être évacué dans les conditions souhaitées. Analysez sa complexité.

Indication. Modéliser le problème comme un problème de flot dans un graphe bien choisi, puis utilisez l'algorithme de flot vu en cours.

Indication. Utilisez une formulation du problème du flot un peu différente, qui requiert que les valeurs du flot sur chaque arc soit des entiers. Montrez que l'algorithme d'Edmonds-Karp permet de résoudre cette variante du problème de flot dans le cas où les capacités sont entières.

Solution. Comme à l'exercice précédent, on peut se ramener à utiliser un algo de flot max/coupe min en modélisant la situation de façon adéquate. Pour modéliser ce genre de problème d'offre et de demande, on peut utiliser un graphe biparti, c'est une approche classique à retenir. D'un côté les blessés, de l'autre les hôpitaux. On se ramène à un problème de flot en construisant un graphe G dans lequel on ajoute une source unique reliée à chaque blessé par une arête et un puits relié à chaque hôpital. Les arêtes entre la source et les blessés reçoivent une capacité de 1 car elles correspondent chacune à un blessé. Les arêtes entre les blessés et les hôpitaux ont aussi une capacité de 1 car elles correspondent à l'affectation d'un blessé à un hôpital. Les arêtes entre les puits et les hôpitaux ont chacune une capacité qui correspond à la capacité d'accueil de l'hôpital en question. On cherche alors à calculer un flot entre s et t qui soit à valeurs entières (c'est à dire que la valeur du flot sur chaque arc est un entier) et de valeur maximum possible avec cette condition.

Pourquoi est-ce que cela répond à la question ? La valeur maximale d'un flot entier est exactement le nombre maximum de blessés que peuvent prendre en charge les hôpitaux avec les contraintes matérielles posées. Pour s'assurer de cela il faut voir d'une part qu'une affectation valide (respectant les contraintes) de b blessés vers les hôpitaux définit un flot de valeurs b dans le graphe G et d'autre part, que réciproquement, un flot de valeurs b dans le graphe G donne une affectation valide de b blessés dans les hôpitaux.

Commençons par considérer une affectation valide de b blessés dans les hôpitaux. On peut construire un flot de valeur b comme suit. Il suffit de mettre le flot à 1 sur les arêtes entre la source et chaque blessé pris en charge, le flot à 1 sur l'arête entre chaque blessé pris en charge et l'hôpital qui le prend en charge et le flot entre l'hôpital h et le puits à la valeur b_h , où b_h est le nombre de blessés pris en charge par l'hôpital h . Comme la capacité de cette dernière arête a été définie précisément comme le nombre maximum de blessés que peut accueillir l'hôpital h , ce flot (qui est un entier) respecte les capacités des arêtes et est donc bien un flot entier valide.

Reciproquement, soit un flot à valeur entière de valeur b . Comme le flot est à valeur entière, chaque blessé reçoit de la source un flot de 1 ou 0. Par conservation du flot, ceux qui reçoivent un flot de 1 envoient aussi un flot de 1 à exactement un hôpital. On affecte chacun des b blessés qui reçoivent un flot de 1 de la source à l'unique hôpital auquel ils envoient un flot de 1. Cela donne une affectation valide car chaque hôpital h reçoit un flot de 1 de la part de b_h blessés, avec b_h qui est au plus la capacité de l'arête entre l'hôpital h et le puits (car le flot est conservatif, respecte les capacités des arêtes, et la capacité de l'arête entre l'hôpital h et le puits a été fixée exactement à la capacité d'accueil de l'hôpital h).

Ainsi, il y a une bijection entre les flots à valeur entières dans G et les affectations valides des blessés : une affectation qui prend en charge le maximum possible de blessés est donc un flot à valeur entières de valeur maximum dans G . En cours, nous n'avons pas vu d'algorithme qui détermine le flot maximum à valeurs entières... mais en fait si ! L'algorithme d'Edmonds-Karp ou de Ford-Fulkerson par exemple. À chaque étape, ces algorithmes modifient le flot sur chaque arc en faisant des additions et soustractions de la valeur courante du flot et des capacités. Comme les capacités sont ici entières et que le flot initial dans l'algorithme est à valeur entières (il s'agit du flot nul), le flot sur chaque arc à chaque étape de l'algorithme est à valeurs entières. Ainsi, le flot maximum retourne à la fin est à valeur entière. Cela montre en fait plus généralement que si les capacités sont entières, il existe toujours un flot maximum qui est à valeur entière. Dans notre cas, on applique donc l'algo d'EK ou de FF pour trouver un flot maximum et on sait que le flot renvoyé sera à valeur entières : c'est donc un flot entier maximum. Remarquez que l'on peut appliquer FF lorsque les capacités sont entières, car il n'y a alors pas de problème de terminaison.

L'algorithme est donc le suivant :

1. on forme le graphe G en mettant une arête entre chaque blessé et ses hôpitaux d'affectation possibles et on ajoute la source et le puits avec les arêtes les reliant aux blessés et aux hôpitaux, affectées de la bonne capacité (voir ci-dessus). Si on note n le nombre total de blessés plus d'hôpitaux et m le nombre de possibilités d'affectation entre tous les blessés et tous les hôpitaux, on obtient un graphe de taille $O(n + m)$ et cela prend un temps $O(n + m)$ de le construire.
2. on applique l'algo d'EK au graphe G ainsi formé, cela prend $O(nm^2)$
3. on obtient ainsi le nombre maximum de blessés qu'on peut affecter aux hôpitaux en respectant les contraintes données (c'est la valeur du flot maximum retournée par EK)
4. si ce nombre est n on répond oui (il existe une affectation possible de tous les blessés) sinon on répond non.

La complexite totale est donc $O(n + m) + O(nm^2) = O(nm^2)$. Et si on veut une affectation des blesses, on en obtient directement une en prenant les aretes entre blesses et hopitaux dont le flot est a 1 dans le resultat retourne par EK. cela prend un temps $O(n + m)$ et n'augmente donc pas la complexite totale de l'algorithme.

Exercice 7.

Sauve qui peut!... mais sans pousser.

Soit G une grille de $n \times n$ points et P un sous-ensemble de p points parmi les n^2 . Le problème de l'évacuation consiste à déterminer s'il est possible de relier chacun des p points au bord de la grille par des chemins sans sommets communs.

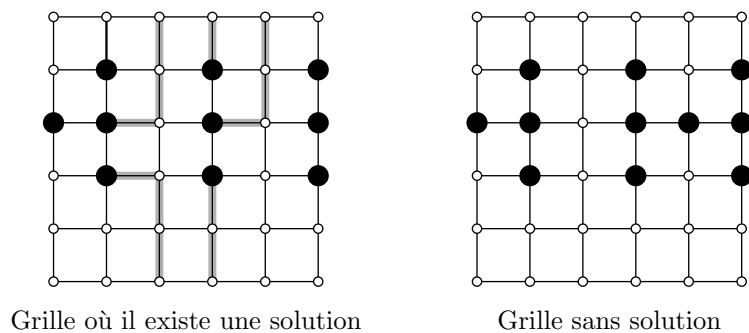


FIGURE 5 – Une grille.

a. On considère l'extension du problème du flot maximum où à la fois les arcs et les sommets ont une capacité maximum (le flot total transitant par un sommet doit être inférieur à sa capacité). Comment se ramener à un problème où seuls les arcs ont des capacités?

Solution. L'idée est de former un graphe modifié \tilde{G} en remplaçant chaque sommet u de G par deux sommets u_{in} et u_{out} dans \tilde{G} , avec un arc de u_{in} à u_{out} . Tous les arcs entrant sur u dans G deviennent alors entrant sur u_{in} dans \tilde{G} et tous les arcs sortant de u deviennent alors sortant de u_{out} . La capacité du sommet u est reportée sur l'arc (u_{in}, u_{out}) et les sommets u_{in} et u_{out} eux memes n'ont pas de capacité, seulement les arcs de \tilde{G} en ont, s_{in} est la source dans \tilde{G} et t_{out} le puits. On peut montrer (faites le) que le flot maximum entre s_{in} et t_{out} dans \tilde{G} est egal au flot maximum entre s et t dans G . De plus, il est aise de transformer un flot dans \tilde{G} en un flot dans G et vice-versa.

b. Comment résoudre le problème initial?

Solution. La resolution de cette question reprends plusieurs des idées que nous avons vu dans les exercices precedents. On modelise d'abord le probleme comme un probleme de flot entier avec capacités sur les arcs et les sommets. Tous les sommets du bord de la grille sont des puits et tous les points donnes sont des sources. Tous les arcs ont capacité 1 (ils pourraient avoir une capacité plus grande, voire infinie, ça conviendrait aussi) et tous les sommets ont capacité 1 (ca c'est la clef). On veut trouver un flot entier (c'est a dire 0 ou 1 sur chaque arc ou sommet) de valeur maximum et savoir si cette valeur est au moins p , le nombre de points donnes. Si oui, on peut evacuer

les p points donne en respectant les contraintes, sinon on ne peut pas. On se ramene a un probleme a source et puits uniques en faisant la transformation usuelle et en attribuant une capacite de p a la nouvelle source et au nouveau puits. On fait ensuite la transformation decrite a la question precedente pour se ramener au cas ou on a des capacites seulement sur les arcs. On cherche ensuite un flot entier maximum avec l'algo d'Edmonds-Karp. Le graphe de la grille a $O(n^2)$ sommets et $O(n^2)$ aretes. Toutes les transformations necessaires peuvent etre faites en temps $O(n^2)$ et la complexite d'EK sur le graphe forme s'exprime comme $O(n^6)$, ce qui contrairement aux apparences est plus efficace que le cas general car ici le nombre de sommets est $\theta(n^2)$, la complexite est donc seulement cubique en le nombre de sommets (ce n'est pas specifique a la grille, c'est vrai pour tout graphe dont le nombre d'aretes est lineaire en le nombre de sommets).

Exercice 8.

Couplage maximum dans les graphes bipartis

Un couplage d'un graphe G est un ensemble M d'aretes de G tel que tout sommet de G est incident a au plus une arete de M . Un couplage maximum est un couplage dont le nombre d'aretes $|M|$ est maximum parmi tous les couplages de G .

Un graphe biparti est un graphe dont l'ensemble des sommets est partitionne en deux parties (L, R) et qui ne contient aucune arete entre deux sommets de L , ni aucune arete entre deux sommets de R . Autrement dit, toutes les aretes du graphe biparti ont une extremite dans L et une dans R .

a. Expliquer comment modéliser le problème du calcul d'un couplage de cardinal maximum dans un biparti sous forme d'un problème de flot maximum.

Indication. Comme a l'exercice 6, vous pourrez utiliser un probleme de flot entier plutot qu'un flot classique, c'est a dire que les capacites sont des entiers et que l'on requiert que la valeur du flot sur chaque arc soit aussi un entier.

Solution. Pas de correction. Tout est dit dans les indications et on a fait ca plusieurs fois dans les corrections des exercices precedents.

b. En déduire un algorithme polynomial pour calculer un couplage max dans un biparti. Précisez sa complexité.

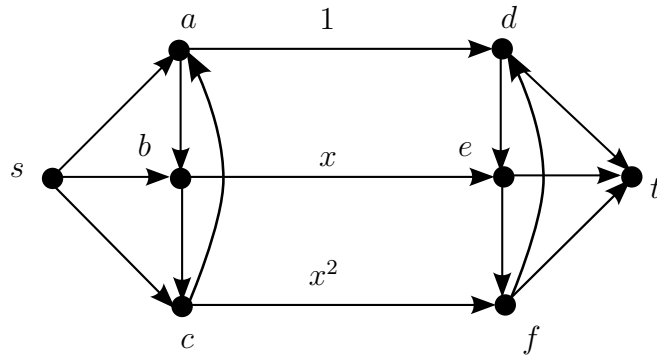
Indication. Comme a l'exercice 6, montrez que les algorithmes d'Edmonds-Karp et de Ford-Fulkerson vus en cours permettent aussi de resoudre le probleme du flot entier maximum.

Solution. Pas de correction. Tout est dit dans les indications et on a fait ca plusieurs fois dans les corrections des exercices precedents.

Exercice 9. *Pour aller plus loin : non-terminaison de la méthode de Ford-Fulkerson dans \mathbb{R}*

Pour un réseau de flot à capacités dans \mathbb{Q}_+ , la méthode de Ford-Fulkerson pour trouver un flot maximum termine en temps fini. On va montrer que ce n'est pas toujours le cas si les capacités sont dans \mathbb{R}_+ .

Soit $x = \frac{\sqrt{5}-1}{2}$, solution de l'équation $x^2 + x - 1 = 0$, et le réseau suivant (les arcs où rien n'est indiqué ont pour capacité $+\infty$) :



a. Exhibez une suite infinie de chemins améliorants de s à t pour l'exemple ci-dessus.

Solution. Voir wikipedia **en anglais** : cherchez "Ford-Fulkerson wiki".

b. En général, si la méthode de Ford-Fulkerson ne termine pas, le flot trouvé tend-il quand même vers un flot maximum ?

Solution. Wikipedia **en anglais** répond aussi à la question : non. Sur l'exemple précédent, la valeur du flot converge en croissant mais ne tend pas vers la valeur maximum!!!