29 août 2018
Licence Parcours Info
Omar Abdelkafi
François Lemaire
Arnaud Liefooghe
Sylvain Salvati
Marie-Émilie Voge
Léopold Weinberg

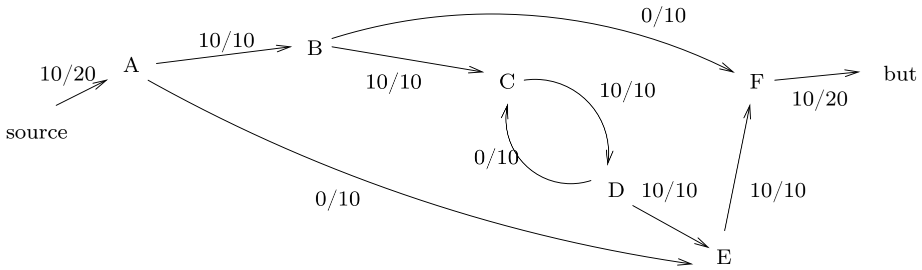
ARO

TD 10

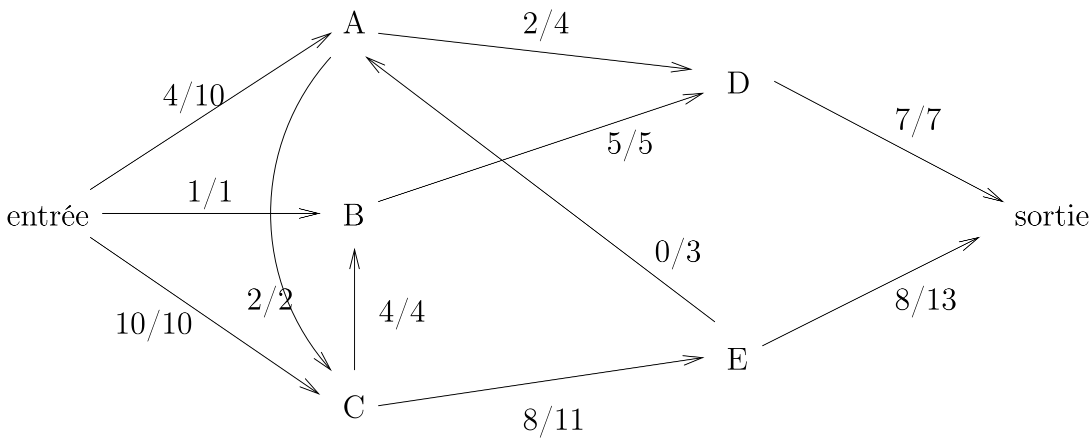
1 Flots de valeur maximale

1.1 Exercices d'application

Q 1. Prouver que ce flot n'est pas maximal. L'augmenter. Combien y a–t–il de flots de valeur maximale ?



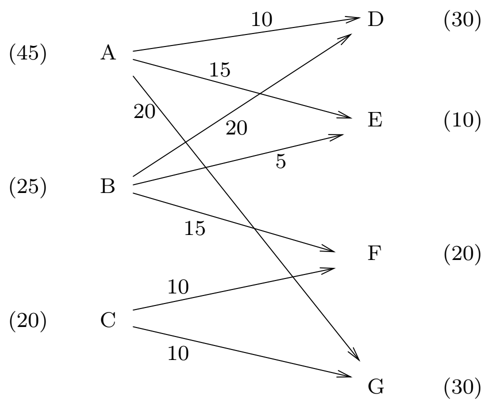
Q 2. Appliquer l'algorithme de Ford–Fulkerson au flot suivant (donner à chaque itération une chaîne améliorante, mais ne pas détailler le calcul de la chaîne). À la fin de l'exécution de l'algorithme, montrer que le flot est bien maximal en utilisant un théorème vu en cours.



Q 3. Trois châteaux d'eau A, B et C alimentent quatre villages D, E, F et G. Sur les arcs du graphe suivant on a reporté les capacités des canalisations existantes. On a reporté sur les châteaux d'eau, entre parenthèses, les capacités maximales de débit. On a reporté sur les villages, entre parenthèses, les débits souhaités. On cherche la meilleure alimentation possible.

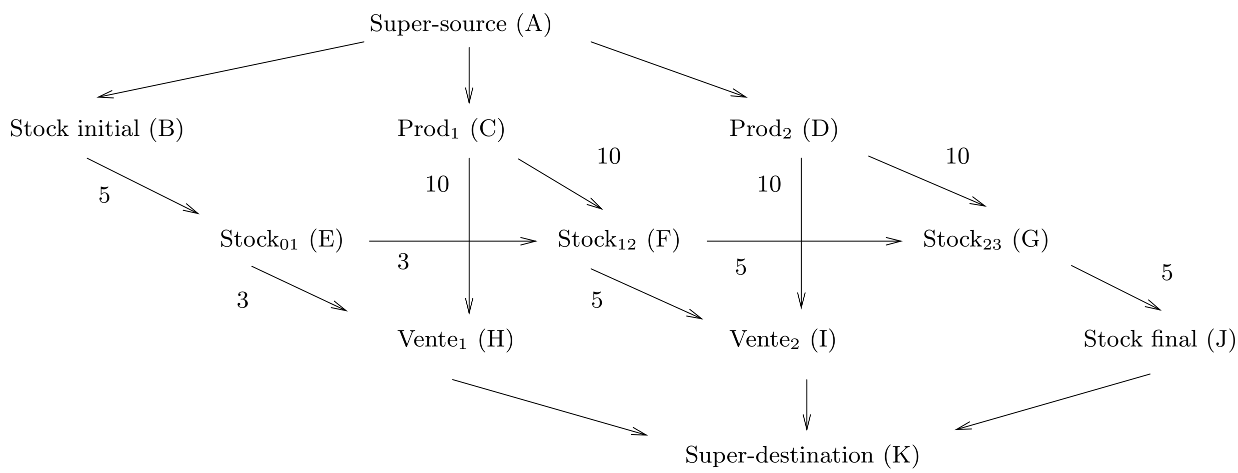
Appliquer l'algorithme de Ford–Fulkerson. Quelle est la valeur maximale du flot ? Est–il possible de satisfaire les souhaits des villages ?

On augmente après coup la capacité de l'arc (A, D) de 5. Reprendre l'exécution de l'algorithme de Ford–Fulkerson à partir du flot précédemment obtenu.



1.2 Flot et gestion de stock

On s'intéresse à l'activité d'une entreprise sur deux mois, désignés par les numéros 1 et 2. Chaque mois, l'entreprise fabrique entre 0 et 10 produits et les vend. Les ventes sont comprises entre 0 et 10 aussi. L'entreprise peut aussi stocker des produits d'un mois sur l'autre. Le stock initial et le stock final sont compris entre 0 et 5. En raison d'un aménagement des entrepôts, la capacité des stocks est réduite pendant le mois 1. Le réseau de transport (incomplet) suivant modélise cette activité. Il comporte trois sources et trois destinations.



Q 4. Déduire de l'énoncé les capacités à donner aux arcs partant de la super–source et à ceux arrivant sur la super–destination.

Q 5. Appliquer l'algorithme d'Edmonds–Karp sur le réseau ainsi obtenu. À chaque itération, déterminer la chaîne améliorante la plus courte possible. Au cas où deux chaînes améliorantes de même longueur seraient disponibles, choisir celle qui passe le plus à gauche possible dans le réseau.

À chaque itération, surligner la chaîne avant augmentation des flux. Préciser si la chaîne comporte un arc indirect. À la fin de l'algorithme, indiquer la coupe exhibée par l'algorithme. Que peut–on dire de la capacité de cette coupe ?

1.3 Réduction de problèmes

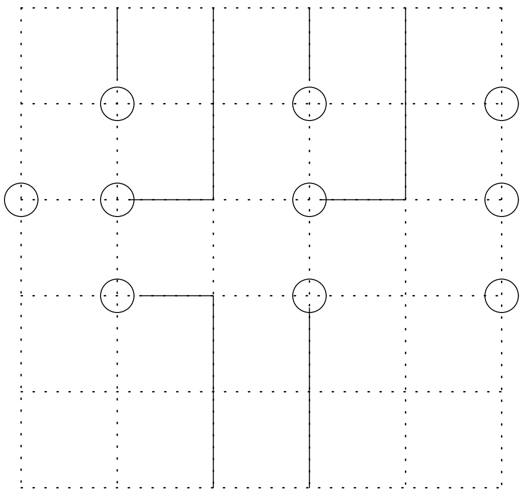
Dans les deux exercices qui suivent, le problème consiste à chercher à réduire en un problème de flot, un problème qui n'est pas posé comme tel.

On ne veut surtout pas modifier les algorithmes étudiés en cours.

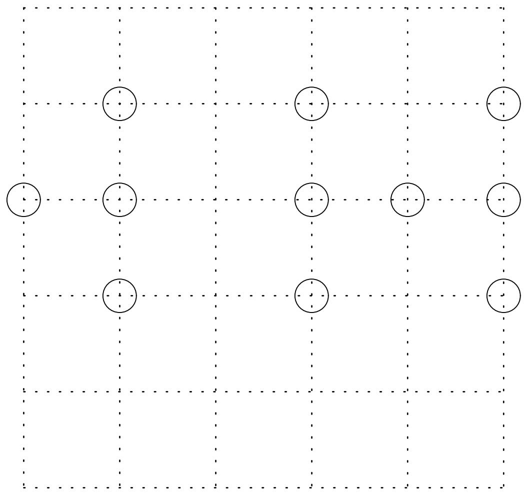
Ce type d'exercices est très important : dans votre future vie professionnelle, il est peu probable qu'on vous demande de programmer un algorithme de flot maximal. Savoir reconnaître un problème donné comme un problème de flot, de façon à pouvoir lui appliquer un algorithme classique, disponible dans une bibliothèque logicielle, semble beaucoup plus utile.

1.3.1 Le problème de l'évacuation

Les cercles représentent les personnes et les traits pleins des trajets d'évacuation



Évacuation possible



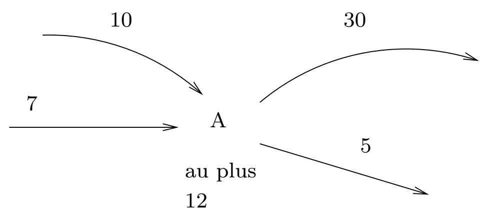
Évacuation impossible

Une *grille* $n \times n$ est un graphe non orienté constitué de n lignes et n colonnes, numérotées de 1 à n. Tous les sommets d'une grille ont exactement quatre voisins à l'exception des sommets situés sur le bord de la grille, qui en ont moins.

On considère p personnes réparties sur p sommets distincts de la grille. En cas d'alerte, on veut que les p personnes évacuent la grille en suivant des trajets qui ne se coupent pas (on ne se préoccupe pas des longueurs des trajets). En d'autres mots, on dit que l'évacuation est possible s'il existe p chaînes reliant chacun des p sommets à un sommet quelconque du bord telles que tout sommet de la grille n'appartienne qu'à une chaîne au plus.

Q 6. (question préliminaire). Il est parfois utile de résoudre le problème du flot maximal pour des réseaux de transport dans lesquels les sommets aussi ont des capacités (pour chaque sommet, il y a une limite sur la somme des flux entrants). Indiquer en quelques phrases comment réduire le problème du flot maximal pour ce type de réseaux en le problème étudié en cours.

À titre d'illustration, on peut considérer l'exemple suivant.



Q 7. Décrire en quelques phrases un algorithme basé sur l'algorithme de Ford–Fulkerson qui résolve le problème de l'évacuation. Bien préciser le réseau de transport auquel appliquer Ford–Fulkerson et justifier. On peut supposer résolu le problème de la question précédente.

1.3.2 Flot maximal à coût minimal

Lorsque plusieurs flots de valeur maximale sont possibles, il est intéressant d'en déterminer un dont le coût est minimal. Dans ce type de problème, on considère des réseaux de transport dont les arcs ont deux valeurs : une capacité et un coût.

Q 8. Cinq personnes doivent emménager dans cinq bureaux a, b, c, d et e. Le tableau suivant donne les préférences de chacune des personnes

	a	b	c	d	e
Albert	1	2	3	4	5
Bertrand	1	4	2	5	3
Charles	3	2	1	5	4
Dimitri	1	2	3	5	4
Eric	2	1	4	3	5

Le problème : affecter un bureau à chaque personne de la façon la plus satisfaisante possible peut se réduire en un problème de calcul de flot maximal de coût minimal. Comment ?

Q 9. Déterminer un flot maximal à coût minimal peut se résoudre par programmes linéaires. Voici un début de modèle AMPL pour l'exemple ci–dessus. Compléter ce modèle.

```
set PERSONNES;
set BUREAUX;

param capacite {PERSONNES, BUREAUX} >= 0;
param cout {PERSONNES, BUREAUX} >= 0;

var flux {p in PERSONNES, b in BUREAUX} >= 0, <= capacite [p, b];
```