ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

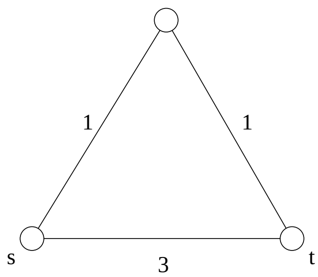
Section d'Informatique et de Systèmes de Communication

Corrigé de la série 10 30/11/2009

1. Dijkstra et Moore-Bellman-Ford

a) Sous la condition que la constante soit positive, oui. En effet, la longueur de chaque chemin sera multipliée par la même constante, ce qui fait que le chemin le plus court reste le même.

b) Non, pas en général. Considérons un exemple simple:



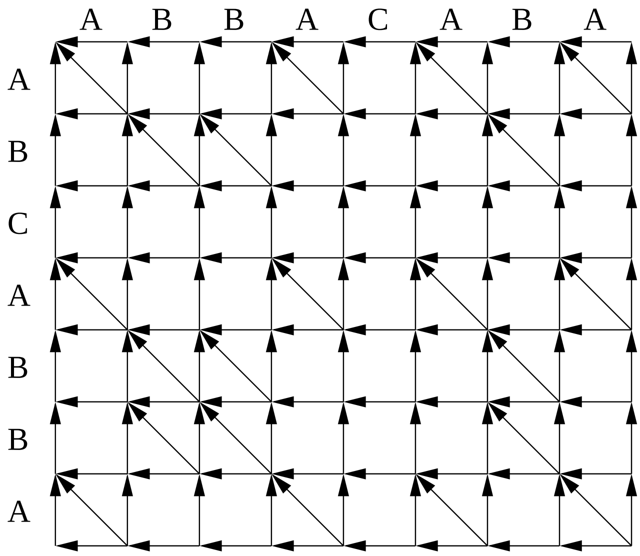
Le plus court chemin de s à t passe évidemment par le troisième sommet dans ce cas. Si on ajoute 100 à chacun des poids, ça ne reste pas vrai.

2. Dijkstra et Programmation Dynamique

a) On utilise le graphe du corrigé de la série 6, exercice 2. On attribue des poids *huge* aux arêtes verticales et $huge - v_i$ aux arêtes diagonales de la i-ème ligne. (*huge* est choisi de sorte que tous les poids soient positifs.)

Il est alors facile de voir qu'un plus court chemin correspond bien à une solution optimale et que son poids sera $n \cdot huge - v_{\text{opt}}$.

b) On fait un quadrillage d'arêtes de poids un, et on ajoute des diagonales de poids zéro là où on a une correspondance dans les suites. Dessin-exemple:



Le plus court chemin d'en bas à droite jusqu'en haut à gauche est celui qui passe par le nombre maximal de diagonales, ce qui nous donne aussi une sous-suite optimale.

c) Pour le point a): Si on a un poids maximal pour le knapsack, par niveau on a à peu près $2m$ arêtes. On a n niveaux, donc à peu pres $2mn = O(mn)$ arêtes, ce qui correspond à la complexité de l'algo DP.

Pour le b): Par nœud du cadrillage, on a au plus 3 arêtes entrantes, donc si on a des sous-suites de longueur m et n, on trouve un nombre d'arêtes $O(mn)$, ce qui correspond à la complexité du premier algo LCS.

Remarquons que ces correspondances ne sont pas des hasards: chaque arête dans le graphe correspond à une décision que l'on prend dans l'algo DP correspondant.

d) Dans ce cas il n'est pas aussi simple de réduire le problème à une application de l'algorithme de Dijkstra. La différence structurelle est que dans ce cas, pour trouver une solution optimale à un sous-problème, on utilise les solutions de *plusieurs* sous-problèmes optimaux.

Plus concrètement, considérons le parenthèsement optimal de $M_i \cdots M_j$. La solution optimale est de la forme $(M_i \cdots M_k) \cdot (M_{k+1} \cdots M_j)$ pour un k, ou $M_i \cdots M_k$ et aussi $M_{k+1} \cdots M_j$ sont des solutions optimales (donc on en réutilise deux!).

Par opposition, si on considère par exemple le sac à dos pour n objets et poids m, on utilise le fait que la meilleure solution est de la forme

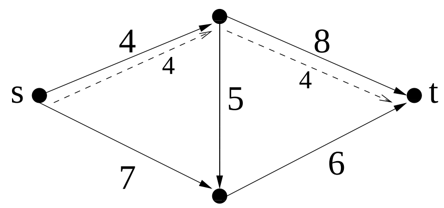
$$S_n(m) = \begin{cases} \text{soit} & \{n\} \cup S_{n-1}(m - c_n), \\ \text{soit} & S_{n-1}(m), \end{cases}$$

donc la solution optimale S_n n'utilise qu'une seule solution optimale de S_{n-1}.

3. Flux

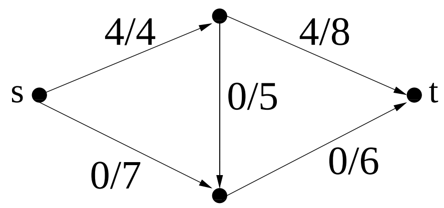
a) Non, ce n'est pas un flux puisqu'il y a un sommet intérieur pour lequel le bilan est non-nul: Dans le sommet en haut ce qui rentre est 2 alors qu'il sort 5 de ce sommet.

b) On utilise l'algorithme maxflow-mincut du cours. On commence par trouver un chemin quelconque de s à t:

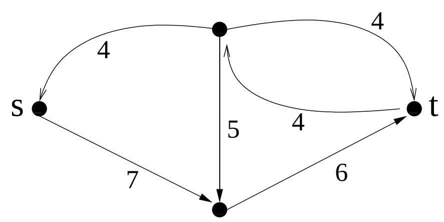


La valeur de ce chemin est 4 car on ne peut pas faire passer plus de 4 dans la première arête.

On obtient donc le flux suivant:



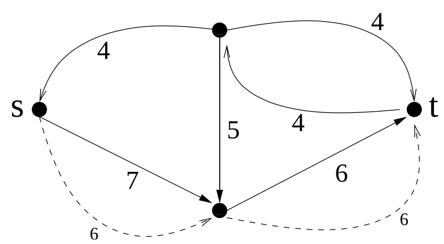
On construit à présent le graphe résiduel correspondant. On rappelle que le graphe résiduel indique comment peut être modifié un flux:



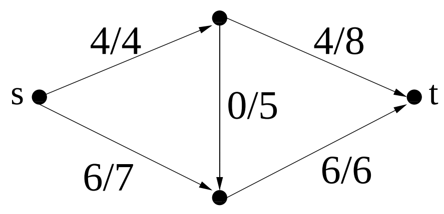
En effet, la première arête quittant s (celle de capacité 4) à une valeur égale à sa capacité, sa valeur ne peut donc pas être augmentée (donc pas de flèche dans le même sens). Par contre, la valeur peut être diminuée à 0 (donc diminuée de 4), on a donc une flèche de capacité 4 dans le sens inverse.

De même, la première arête arrivant vers t (celle de capacité 8) peut soit voir sa valeur augmentée de 4 (pour arriver à sa capacité) ou soit diminuée de 4 (pour revenir à 0), on a donc une flèche de capacité 4 dans chaque sens. etc.

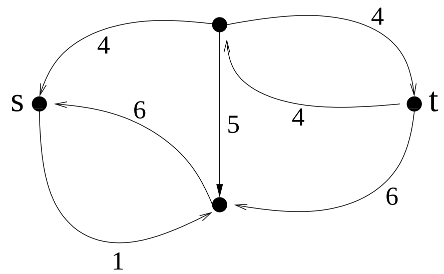
On choisit un chemin de s à t dans ce graphe résiduel:



On obtient donc le flux suivant:



On construit maintenant le graphe résiduel correspondant pour voir si ce flux peut encore être amélioré:



On voit qu'il n'y a aucun chemin de s à t, notre flux est donc maximal.

Sa valeur est égale à

$$|f| = \sum_{e \in E(s, V \setminus \{s\})} f(e) - \sum_{e \in E(V \setminus \{s\}, s)} f(e).$$

Donc dans ce cas

$$(4 + 6) \quad - \quad 0 = 10$$

c) On voit dans la question précédante que l'arrête de capacité 5 n'est pas utilisée dans le flux maximal, donc si elle est enlevée le flux maximal restera le même.

d) Dans ce réseau très simple, si la capacité de la première arête quittant s est augmentée de 4 à 10, alors le flux pourra être augmenté de 10 à 14.

De manière générale, il faut choisir une arête adéquate sur le min cut du graphe, ce qui permet de considérablement simplifier le problème pour les réseaux plus compliqués.

4. Attribution optimale

a) Le problème ressemble à un problème de bipartite matching, et donc on veut appliquer l'algorithme max-flow-min-cut pour le résoudre. La seule subtilité est celle des chambres doubles. On peut la résoudre en travaillant plutôt sur des "lits" qu'on met dans les chambres. Dans les chambres simples on met un lit simple, dans les chambres doubles on en met deux: un pour les informaticiens et un pour les physiciens.

b) On obtient le graphe suivant: