DM11

Dans ce problème, les programmes sont écrits en langage OCaml.

Le calcul de *flots* dans des graphes permet de modéliser et de répondre à une large classe de problèmes, dont l'interprétation correspond à la circulation de flux physiques sur un réseau : distribution électrique, acheminement de paquets de données sur Internet, etc. Il s'agit en général d'acheminer la plus grande quantité possible de *matière* (courant, données, ...) entre une source s et une destination t. Les liens permettant d'acheminer les flux ont une capacité limitée et il n'y a ni perte, ni création de matière lors de l'acheminement : pour chaque noeud intermédiaire du réseau, le flux entrant (ce qui arrive) doit être égal au flux sortant (ce qui repart).

Ce sujet étudie une modélisation de ces problèmes de *flots* par des graphes, dits réseaux de transport. Un algorithme de recherche d'un *flot maximal* y est détaillé et relié à d'autres notions sur les graphes, telles que les *coupes* ou les *couplages*.

Définitions et propriétés

Dans toute la suite, nous considérons un graphe orienté pondéré $G = (V_G, E_G)$, où V_G est l'ensemble des sommets de G et E_G l'ensemble des arcs de G. $|V|$ désigne le cardinal de V_G, c'est-à-dire le nombre de ses sommets.

G est un **réseau de transport** si :

- il existe dans V_G deux sommets particuliers, la **source** s_G et le **puits** t_G ;
- il existe une fonction $c : V_G \times V_G \to \mathbb{R}_+ \cup \{\infty\}$, appelé **capacité** et évaluant les arcs $(u, v) \in E_G$ de G.

Dans cette définition, ∞ est un élément absorbant pour tous les opérateurs arithmétiques, sauf l'inversion.

Un **flot** de G est une fonction $x : V_G \times V_G \to \mathbb{R}$ qui, à chaque arc (u, v) de E_G, *associe* une **quantité de flot** $x(u, v)$.

On définit pour x une fonction **bilan** $\mathcal{B} : V_G \to \mathbb{R}$ telle que :

$$\forall y \in V_G \quad \mathcal{B}(y) = \sum_{\substack{(u,v)\in E_G \\ u=y}} x(u,v) - \sum_{\substack{(v,u)\in E_G \\ u=y}} x(v,u)$$

Le flot x est alors de valeur v si :

$$\begin{cases} & \mathcal{B}(s_G) = v \\ & \mathcal{B}(t_G) = -v \\ \forall u \in V_G \backslash \{s_G, t_G\} & \mathcal{B}(u) = 0 \end{cases} \tag{1}$$

et :

$$\forall (u, v) \in E_G \quad 0 \leqslant x(u, v) \leqslant c(u, v) \tag{2}$$

Question 1. Donner une interprétation des équations (1) et (2).

Le problème du *flot maximum* consiste à maximiser v en respectant les contraintes imposées par (1) et (2). Dans la suite du problème, on suppose que :

- G ne contient aucun circuit orienté de s_G vers t_G composé uniquement d'arcs de capacité infinie ;
- G est un graphe symétrique, ce qui est possible car nous autorisons les arcs à avoir une capacité nulle. Seuls les arcs dont la capacité est strictement positive seront représentés.

La valeur maximum d'un flot peut être reliée avec une autre caractéristique du réseau de transport G.

Soit $S \subset V_G$ et $T = V_G \setminus S$ tels que $s_G \in S$ et $t_G \in T$. Les arcs dont l'origine est dans S et l'extrémité terminale dans T forment une s, t-**coupe** de G. La capacité de la coupe est notée :

$$c(S, T) = \sum_{\substack{(u,v)\in E_G \\ u\in S, v\in T}} c(u, v)$$

Une **coupe minimale** est une coupe de capacité minimale parmi toutes les s, t-coupes de G.

Question 2. Soit x un flot de valeur v dans G et (S, T) une s, t-coupe de G. Montrer que :

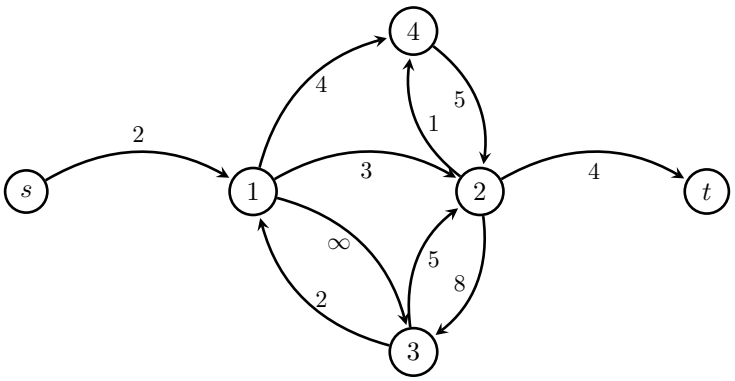
$$v \leqslant c(S, T)$$

Soit G un réseau de transport et x un flot sur G. Si la valeur du flot x est égale à la capacité d'une coupe minimale de G alors le **flot** est **maximal**. À l'inverse, si une coupe de G a une capacité égale à la valeur d'un flot maximal alors cette coupe est minimale.

Cette relation sera utilisée pour stopper dans de bonnes conditions un algorithme itératif de maximisation de flot.

Modélisation

Dans la suite, un graphe est représenté par sa matrice d'adjacence A, $A_{u,v}$ représentant la capacité de l'arc (u, v). Le réseau de transport G suivant est utilisé pour répondre aux questions de ce paragraphe.



Question 3. Définir deux types `reseau_transport` et `flot` permettant de modéliser un réseau de transport et un flot. En particulier, préciser comment indiquer que deux sommets ne sont pas reliés dans le graphe et comment modéliser une capacité infinie. Déclarer alors le réseau de transport G.

Question 4. Écrire une fonction `flot_admissible : reseau_transport -> flot -> bool` qui teste qu'un flot vérifie les équations (1) et (2) pour le réseau de transport passé en paramètre.

Algorithme de Ford et Fulkerson

Algorithme 1 : schéma général de l'algorithme de Ford et Fulkerson

Entrée : un réseau de transport $G = (V_G, E_G)$
Sortie : un flot x

pour $(u, v) \in E_G$ **faire**
  $x(u, v) \leftarrow 0$
tant que *il existe un chemin améliorant* P *de* s_G *à* t_G **faire**
  calculer α, augmentation maximum sur le chemin P
  $x(u, v) \leftarrow x(u, v) + \alpha$ pour tous les arcs (u, v) pris dans le sens direct
  $x(u, v) \leftarrow x(u, v) - \alpha$ pour tous les arcs (u, v) pris dans le sens contraire
renvoyer x

Soit G un réseau de transport et x un flot. Un **chemin** d'origine u et d'extrémité v dans un graphe $G = (V_G, E_G)$ est défini par une suite finie d'arcs consécutifs reliant u à v.

Plusieurs chemins peuvent exister entre deux sommets de G.

Question 5. Définir une fonction récursive `chemin(g,u,v)` qui renvoie la liste des sommets successifs dans le graphe `g` pour passer de `u` à `v`. Dans le cas où plusieurs chemins existent, la fonction renvoie un seul de ces chemins.

Un **chemin** P de s_G à t_G dans G est dit **améliorant** de valeur α lorsqu'il construit un flot x' à partir de x tel que :

- si l'arc $(u, v) \in P$ est pris dans le sens direct, son flot peut être augmenté de α unités :

$$x'(u, v) = x(u, v) + \alpha \qquad \text{et} \qquad x'(u, v) \leqslant c(u, v)$$

- si l'arc $(u, v) \in P$ est pris dans le sens contraire, son flot peut être diminué de α unités :

$$x'(u, v) = x(u, v) - \alpha \qquad \text{et} \qquad x'(u, v) \geqslant 0$$

Question 6. Montrer que x' est toujours un flot de G. Donner la valeur de x' en fonction de x.

Question 7. Soit P un chemin améliorant. Pour un arc $(u, v) \in P$, donner la capacité maximum d'augmentation α, suivant qu'il soit parcouru dans le sens direct ou contraire, pour que x' reste un flot du réseau de transport G. En déduire, pour P donné, l'augmentation maximale possible de P pour que x' reste un flot de G. L'algorithme de Ford et Fulkerson décrit dans l'algorithme 1 reprend l'idée du chemin améliorant pourconstruire un flot maximum dans un réseau de transport.

Question 8. Démontrer que l'algorithme de Ford et Fulkerson s'arrête effectivement.

Graphes des résidus et algorithme d'étiquetage

Tel qu'il est décrit dans l'algorithme 1, l'algorithme n'est pas exploitable, en raison de sa complexité élevée.

Question 9. Si $|V|$ est le nombre de sommets du réseau de transport et $|E|$ le nombre d'arcs, évaluer la complexité de l'algorithme de Ford et Fulkerson, dans le cas où les capacités sont entières et bornées par C, nombre positif fini.

L'algorithme 1 présente un principe général qui a donné lieu à de nombreuses variantes en vue de diminuer la complexité. Les questions suivantes traitent l'une de ces variantes. La notion *graphe des résidus* permet de construire une variante efficace de l'algorithme de Ford et Fulkerson.

Soit $G = (V_G, E_G)$ un réseau de transport, muni d'une fonction capacité c. Soit x un flot dans G. Le **graphe des résidus** de G, noté $G[x]$, possède les mêmes sommets et arcs que G. Chaque arc (u, v) est valué par un résidu $r(u, v)$, donné par :

$$\forall (u,v) \in E_G \quad \begin{cases} r(u,v) = c(u,v) - x(u,v) + x(v,u) \\ r(v,u) = c(v,u) - x(v,u) + x(u,v) \end{cases}$$

Question 10. Définir un type `residu` permettant de modeliser un graphe des résidus.

Question 11. Écrire une fonction `graphe_residu : reseau_transport -> flot -> residu` qui, à partir d'un réseau de transport G et d'un flot x, renvoie un graphe des résidus de type `residu` spécifiant le graphe des résidus entre le réseau de transport G et le flot x. Cette fonction fera impérativement appel à une fonction récursive.

Utilisons alors $G[x]$ pour construire un algorithme dérivant de Ford et Fulkerson, appelé *algorithme d'étiquetage* (algorithme 2), étudié dans la suite de ce problème. Dans cet algorithme, la constante *NULL* désigne l'absence de valeur associée à la variable correspondante.

Algorithme 2 : algorithme d'étiquetage

Entrée : un réseau de transport $G = (V_G, E_G)$
Sortie : un flot x

```
pour (u, v) ∈ E_G faire
    x(u, v) ← 0
Marque[t] ← Vrai
tant que Marque[t] faire
    pour u ∈ V_G faire
        Marque[u] ← Faux
        Pred[u] ← NULL
    Marque[s] ← Vrai
    S ← {s}
// ===== Phase 1
tant que S ≠ ∅ et Non(Marque[t]) faire
    choisir u ∈ S
    S ← S \ {u}
    pour (u, v) ∈ G[x] faire
        si r(u, v) > 0 et Non(Marque[v]) alors
            Pred[v] ← u
            Marque[v] ← Vrai
            S ← S ∪ {v}
// ===== Phase 2
si Marque[t] alors
    P ← {(u, v) | Pred[v] = u
    α ← min(r(u, v), (u, v) ∈ P)
    MiseAJour((u, v), P, x, α)
renvoyer x
```

Question 12. Montrer que l'on a toujours $r(u, v) \geqslant 0$.

Question 13. À l'aide d'un exemple simple d'un réseau de transport à deux sommets, montrer qu'il est possible d'avoir $r(u, v) > c(u, v)$. Il n'est donc pas toujours possible d'ajouter le résidu au flot de l'arc correspondant.

Question 14. Soit un chemin améliorant P de α dans $G[x]$, empruntant l'arc (u, v), de capacité $c(u, v)$ et de flot $x(u, v)$ tels que $x(u, v) \leqslant c(u, v)$ mais $x(u, v) + \alpha \geqslant c(u, v)$. Proposer une stratégie d'amélioration de P permettant d'augmenter

la valeur du flot x de la quantité α, tout en conservant les contraintes (2).
En déduire une interprétation du résidu. À quelle étape de l'algorithme 2 correspond cette stratégie ?

On s'intéresse maintenant à la phase 2 de l'algorithme 2.

Question 15. Si la condition `Marque[t]` est vérifiée, que cela implique-t-il sur la recherche des chemins dans le réseau de transport ? Que représentent alors P et α ?

L'algorithme 2 poursuit son exécution tant qu'un chemin améliorant de s_G à t_G dans $G[x]$ existe. À l'arrêt, notons S l'ensemble des sommets marqués et $T = V_G \setminus S$.

Question 16. Montrer que (S, T) est une s,t-coupe de G et que pour tout couple $(u, v) \in S \times T$, $r(u, v) = 0$. En déduire que :

(i) pour tout arc (u, v) de S vers T, $x(u, v) = c(u, v)$;

(ii) pour tout arc (u, v) de T vers S, $x(u, v) = 0$.

Question 17. Soit x un flot de valeur v dans G. En utilisant la question 2, montrer alors que :

$$v = c(S, T)$$

La valeur d'un flot maximum de s_G à t_G est donc égale à la plus petite capacité d'une s,t-coupe.

Application : problème de couplage

Soit $G = (V_G, E_G)$ un graphe non-orienté et (A, B) une partition de V_G.
Un **couplage** de G est un ensemble d'arêtes $E'_G \subset E_G$ tel que les arêtes de E'_G soien deux à deux non adjacentes.

En considérant une partition (A, B) de V_G, un couplage de G peut aussi être vu comme un ensemble d'arêtes $E'_G \subset E_G$ tel que toute arête de E'_G relie un sommet dans A à un sommet dans B.

Dans toute la suite, on recherche un couplage de G qui contient un maximum d'arêtes (*couplage maximum*).

Question 18. Montrer que ce problème peut se modéliser comme un problème de flot. Définir le réseau de transport correspondant et le flot associé.

Question 19. Proposer une implémentation récursive de la recherche d'un couplage maximum.