

### Exercice 1

Le but de cet exercice est de traiter le problème des flots canalisés, c'est à dire où sur chaque arc, le flot doit être compris entre une valeur minimum ( $\geq 0$ ) et une valeur maximum.

Les réseaux munis de capacités minimum et maximum (on parlera de réseaux à capacité bornée) n'admettent pas tous un flot réalisable. C'est le cas de l'exemple de la figure 1 où l'on souhaite un flot maximum entre  $s$  et  $t$ . Les capacités minimum et maximum sont données entre parenthèses. Le flot sur l'arc entre  $s$  et  $A$  devra ainsi avoir une valeur comprise entre 2 et 3. Celui sur l'arc entre  $A$  et  $t$  devra avoir une valeur comprise entre 0 et 1.



FIGURE 1 – Réseau à capacité bornée sans flot réalisable

Pour trouver un flot maximum sur un réseau à capacité bornée, il faut donc d'abord trouver un flot réalisable, puis améliorer ce flot en utilisant par exemple l'algorithme de Ford et Fulkerson, rappelé en fin d'exercice. Cette version de l'algorithme tient compte de l'existence d'une capacité minimum autre que 0. Les modifications par rapport à un réseau où les capacités minimum sont toutes nulles sont :

- dans la procédure `FlotMaximum`,
  - l'ajout du paramètre `b` donnant les valeurs de capacité minimum,
  - l'ajout du paramètre `fi` donnant une valeur de flot initiale réalisable,
  - les lignes 1 et 2 réalisant cette initialisation ;
- dans la procédure `Marquage`,
  - l'ajout du paramètre `b` donnant les valeurs de capacité minimum,
  - la modification des lignes 9 et 12 prenant en compte ces valeurs minimum.

**Recherche d'un flot réalisable** Soit  $R = (G, s, t, b, c)$  un réseau à capacité bornée.  $G$  désigne le graphe sous-jacent,  $s$  et  $t$  respectivement la source et la destination du flot,  $b$  et  $c$  respectivement les valeurs minimum et maximum du flot sur chaque arc. On construit à partir de  $R$  un réseau associé  $R' = (G', s', t', c')$  munis uniquement d'une capacité maximum  $c'$ , la valeur minimum du flot sur chaque arc étant 0. Pour tout sommet  $v$ ,  $E^-(v)$  et  $E^+(v)$  désigneront respectivement l'ensemble des arcs entrant et sortant de  $v$ . Par exemple, sur le graphe de la figure 1,  $E^-(A) = \{(s, A)\}$  et  $E^+(A) = \{(A, t)\}$ .

### Construction de $R'$

1.  $V(G') = V(G) \cup \{s', t'\}$ .
2. Ajouter dans  $G'$  tous les arcs de  $R$  avec comme nouvelle capacité  $c'(e) = c(e) - b(e)$ .
3. Pour tout sommet  $v \in V(G) \setminus \{t\}$ , ajouter dans  $G'$  un arc  $e = (v, t')$  de capacité  $c'(e) = \sum_{f \in E^+(v)} b(f)$ .

4. Pour tout sommet  $v \in V(G) \setminus \{s\}$ , ajouter dans  $G'$  un arc  $e = (s', v)$  de capacité  $c'(e) = \sum_{f \in E^-(v)} b(f)$ .
5. Ajouter deux arcs  $(s, t)$  et  $(t, s)$  dans  $G'$ , de capacité  $c'(s, t) = c'(t, s) = \infty$ .

Le réseau associé au réseau de la figure 1 est donné par la figure 2.

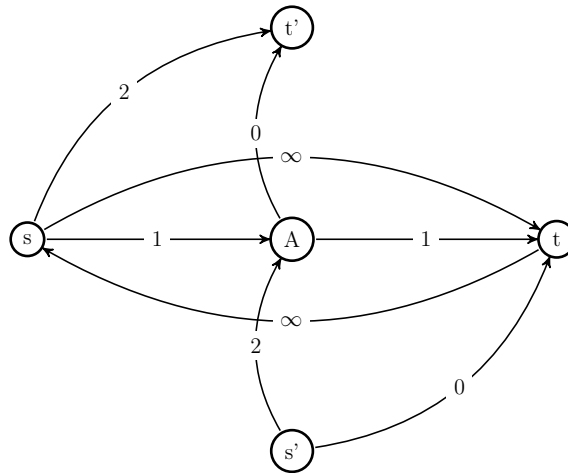


FIGURE 2 – Réseau associé au réseau de la figure 1

- 1) Dessiner, le plus clairement possible, le réseau associé au réseau à capacité bornée de la figure 3.

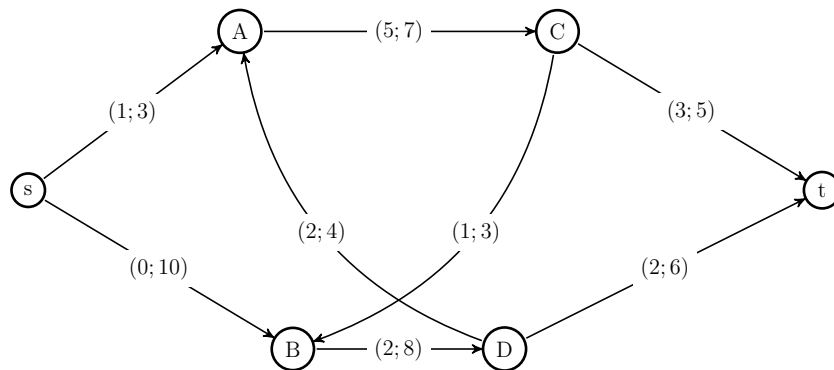


FIGURE 3 – Réseau à capacité bornée

- 2) En utilisant l'algorithme de Ford-Fulkerson rappelé ci-dessous, trouver le flot maximum entre les sommets  $s'$  et  $t'$  du réseau associé  $R'$  obtenu à la question 3.1. Le flot initial sera le flot nul (= 0 sur tous les arcs). On vérifiera que le flot maximum trouvé sature bien tous les arcs sortants de  $s'$  et tous les arcs entrants de  $t'$ .
- 3) Soit  $f'$  le flot maximum de  $R'$  trouvé à la question précédente (3.2). On définit un flot  $f_i$  sur  $R$  de la façon suivante : pour tout arc  $e$  de  $G$ ,  $f_i(e) = f'(e) + b(e)$ . Donnez les valeurs de  $f_i$  et vérifiez que le flot  $f_i$  est bien réalisable pour  $R$ .
- 4) En utilisant comme flot initial le flot réalisable  $f_i$  trouvé à la question précédente (3.3), donner un flot maximum pour le réseau de la figure 3.

On veut maintenant prouver qu'un réseau quelconque  $R$  à capacité bornée possède un flot réalisable  $f$  si et seulement si le réseau associé  $R'$  (tel que défini dans l'entête) possède un flot maximum  $f'$  qui sature tous les arcs sortant de  $s'$ .

5) Montrer que si  $R$  possède un flot réalisable, alors  $f'$  sature tous les arcs sortant de  $s'$  (condition nécessaire).

Pour montrer que la condition de saturation est suffisante, on va construire à partir de  $f'$  un flot réalisable pour le réseau  $R$ . Pour cela, on définit  $f_i$  le flot sur  $R$  défini par : pour tout arc  $e$  de  $G$ ,  $f_i(e) = f'(e) + b(e)$ .

6) Montrer que pour tout arc  $e$  de  $G$ , on a bien  $b(e) \leq f_i(e) \leq c(e)$ .

7) Montrer que pour tout sommet  $v$  de  $G$ , on a bien  $\sum_{e \in E^-(v)} f_i(e) = \sum_{e \in E^+(v)} f_i(e)$

**Rappel :**

Les paramètres  $b$  et  $c$  désignent les capacités minimum et maximum du graphe  $G$ ,  $s$  la source et  $t$  la destination du flot  $f$  calculé.  $f_i$  désigne la valeur initiale du flot (c'est à dire un flot réalisable).  $I(e)$  et  $T(e)$  désignent respectivement le sommet initial et le sommet terminal d'un arc  $e$ .

```

0  FlotMaximum(G,b,c,s,t,fi)
1  pour tout e de E(G) faire
2    f[e] <- fi(e)
3  repeter
4    Marquage(G,b,c,f,s,t)
5    si t appartient a Y
6      alors v <- t
7          C+ <- {(t,s)}
8          C- <- {}
9    tant que v different de s faire
10     e <- A[v]
11     si v = T(e)
12       alors C+ <- C+ U {e}
13         v <- I(e)
14     sinon C- <- C- U {e}
15         v <- T(e)
16     pour tout e de C+ faire
17       f(e) <- f(e) + delta[t]
18     pour tout e de C- faire
19       f(e) <- f(e) - delta[t]
20  tant que t appartient a Y

```

```

0  Marquage(G,b,c,f,s,t)
1  Y <- {s}
2  delta[s] <- infini
3  Max <- faux
4  tant que Y ne contient pas t et Max = faux faire
5      si il existe e = (u,v) avec Y contenant u mais pas v et f(e) < c(e)
6          alors Y <- Y U {v}
7              A[v] <- e
8              delta[v] <- min(delta[u], c(e)-f(e))
9  sinon si il existe e = (u,v) avec Y contenant v mais pas u et f(e) > b(e)
10     alors Y <- Y U {u}
11         A[u] <- e
12         delta[u] <- min(delta[u], f(e)-b(e))
13     sinon Max <- vrai

```