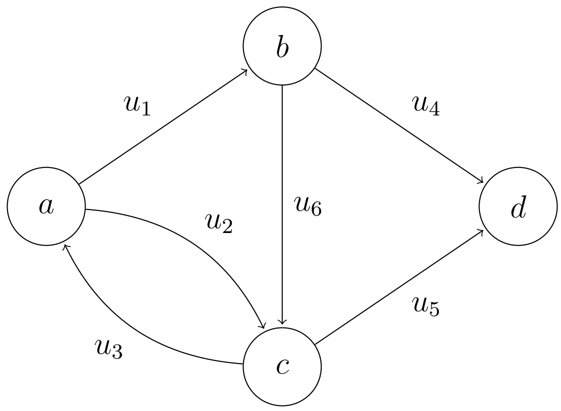
Programmation Avancée - TD1

Elana Courtines

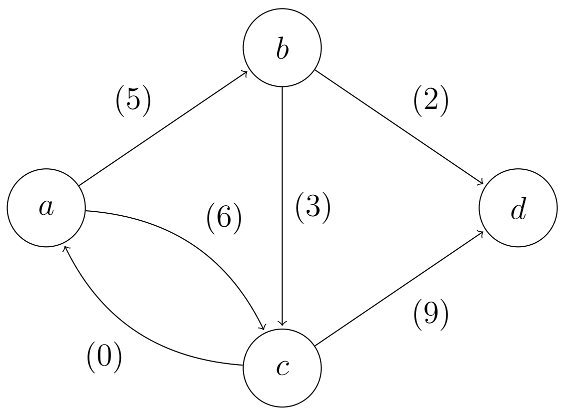
2022-09-12

1 Loi de Kirchhoff et algorithme de Ford-Fulkerson

Exercice 1 :



Question 1.1 : (un seul parce que flemme)



Question 1.2 :

Rappelons la loi de Kirchhoff :

$\forall x$ *un noeud tq.* $x \neq s$ *et* $x \neq t$

Alors : $\sum_{y \in \omega^-(x)} \varphi(y,x) = \sum_{y \in \omega^+(x)} \varphi(y,x)$

Soit $A \subseteq X$,

- $(x,y) \in \omega^-(A)$ *ssi* $x \notin A$ *et* $y \in A$
- $(x,y) \in \omega^+(A)$ *ssi* $x \in A$ *et* $y \notin A$

$\forall x \in A\ :\ \sum_{y \in \omega^-(x)} \varphi(y,x) = \sum_{y \in \omega^+(x)} \varphi(y,x)$

$\rightarrow (F)\ \sum_{x \in A} \sum_{y \in \omega^-(x)} \varphi(y,x) = \sum_{x \in A} \sum_{y \in \omega^+(x)} \varphi(y,x)$

Ce résultat n'exclut pas les arcs internes (c'est à dire qui sont au sein de A), le prochaine étape consiste donc à prouver qu'on peut les exclure.

Prenons l'arc $(x_1, x_2) \in U$ tq. $x_1, x_2 \in A$

et $x_1 \in \omega^-(x2)$

et $x_2 \in \omega^+(x1)$

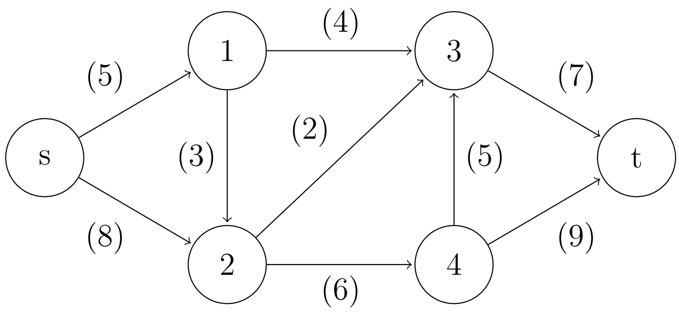
$\varphi(x,y)$ apparaît alors des deux côtés $\rightarrow$ on peut donc simplifier F.

Quand on a simplifié par tous les termes internes à A, il reste

- les arcs de $\omega^-(A)$ à gauche ;
- les arcs de $\omega^+(A)$ à droite.

Nous avons donc bien la loi de Kirchhoff généralisée. CQFD. $\square$

Exercice 2 :



Définition capacité d'une coupe : capacité de tous les arcs sortants par la coupe.

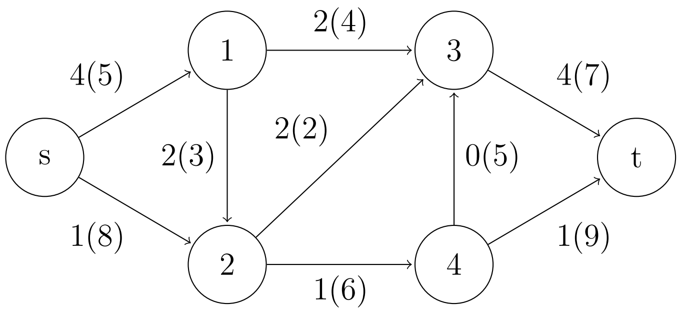
Question 2.1 : (un seul parce que flemme)

$A = (s,1,2,3) \rightarrow capa(A) = 7 + 6 = 13$

Question 2.2 :

$A' = (s,1,2) \rightarrow capa(A') = 6 + 4 + 2 = 12$ est a priori minimal (pas de preuve)

Question 2.3 :



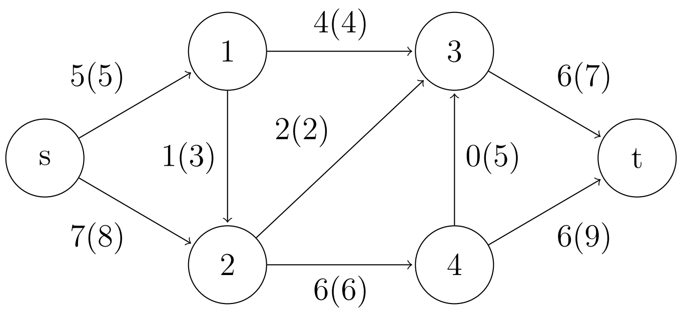
Un flot est compatible si la loi de K est vérifiée en tout $x \neq s, t$:
- aucune des capacités n'est dépassée ;
- loi de K en 1 : 4 - 2+2 ok ;
- loi de K en 2 : 1+2 - 2+1 ok ;
- loi de K en 3 : 2+2 - 4 ok ;
- loi de K en 4 : 1 - 1 ok ;

Et $v(\varphi_0) = 5$.

Question 2.5 :

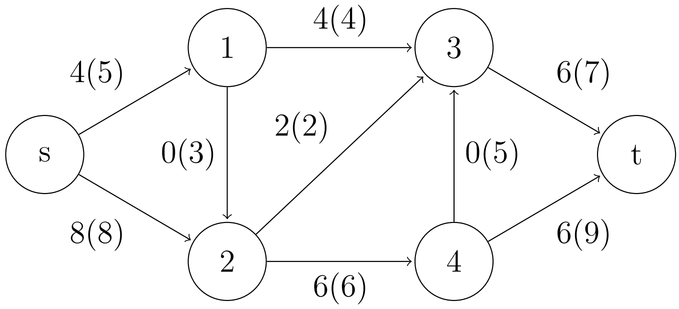
marquage	φ	s1	s2	12	13	23	24	3t	43	4t
	φ_0	4	1	2	2	2	1	4	0	1
s,2,4,t $\rightarrow$ +5	φ_1	4	6	2	2	2	6	4	0	6
s,1,2,t $\rightarrow$ +1	φ_2	5	6	2	3	2	6	5	0	6
s,(2,1),3,t $\rightarrow$ +1	φ_3	5	7	1	4	2	6	6	0	6

Table 1: Version en démarrant à partir du graph 2.5



marquage	φ	s1	s2	12	13	23	24	3t	43	4t
	φ_0	0	0	0	0	0	0	0	0	0
s,1,3,t $\rightarrow$ +4	φ_1	4	0	0	4	0	0	4	0	0
s,2,3,t $\rightarrow$ +2	φ_2	4	2	0	4	2	0	6	0	0
s,2,4,t $\rightarrow$ +6	φ_3	4	8	0	4	2	6	6	0	6

Table 2: Version en démarrant d'un graphe vide

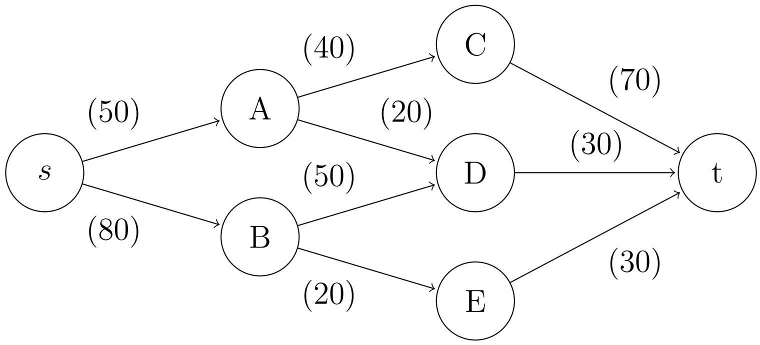


2 Modélisation

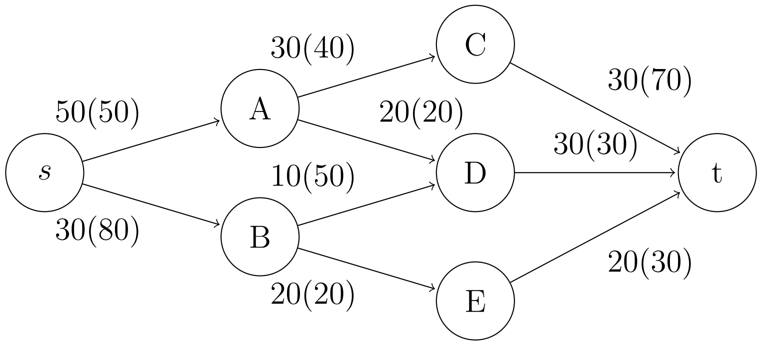
Exercice 3 :

Question 3.1 :

Schématisation de la situation :



Question 3.2 :



Un flot est compatible si la loi de K est vérifiée en tout $x \neq s, t$:
- aucune capacité n'est dépassée ;
- loi K en A : 50 - 20+30 ok ;
- loi K en B : 30 - 10+20 ok ;
- loi K en C : 30 - 30 ok ;
- loi K en D : 20+10 - 30 ok ;
- loi K en E : 20 - 20 ok ;

Et $v(\varphi_0) = 80$.

Question 3.3 :

	sA	sB	AC	AD	BD	BE	Ct	Dt	Et
φ_0	0	0	0	0	0	0	0	0	0
φ_1	40	0	40	0	0	0	40	0	0
φ_2	40	30	40	0	30	0	40	30	0
φ_3	40	50	40	0	30	20	40	30	20

Un flot de valeur 90 est atteint ici, ce qui correspond à la valeur de la coupe {s,A,B,D}, l'algorithme est donc terminé. Si cette justification est suffisante en soi, plus peut être demandé en examen : terminer l'algorithme complètement pour déterminer qu'il n'y a plus aucune chaîne augmentante.

Programmation Avancée - TD2

Elana Courtines

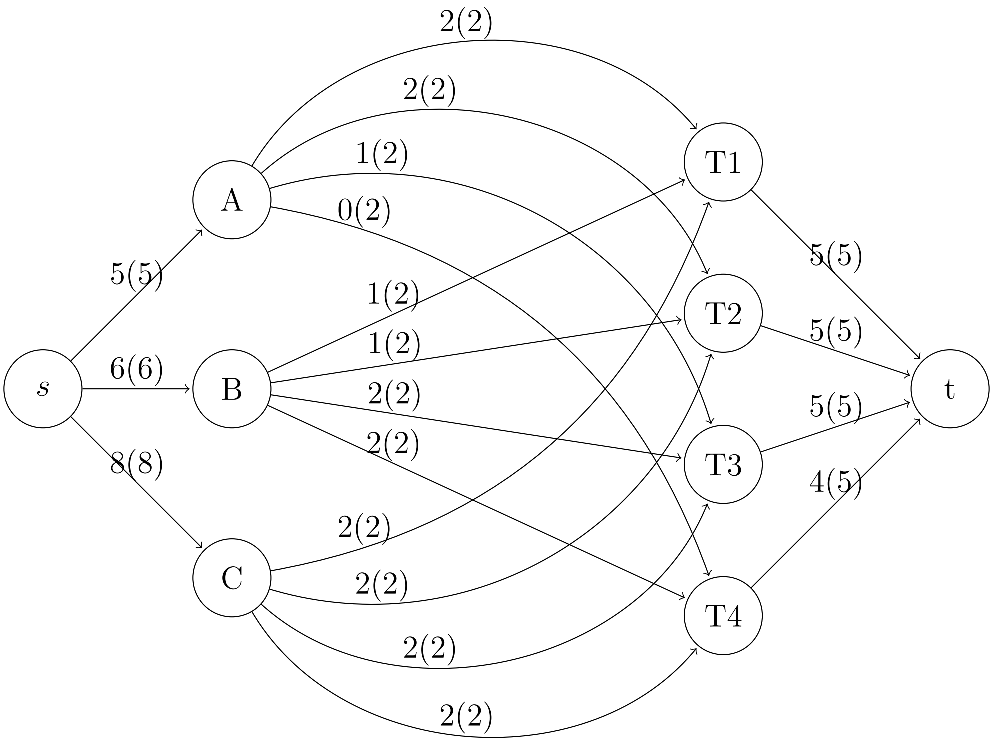
2022-09-19

Exercice 4 :

Trois familles A,B et C souhaitent diner ensemble. Mais pour développer leurs affinités, elles aimeraient qu'à chaque table, il y ait au plus deux membres de chaque famille. Supposons que la famille A possède 5 membres, la famille B 6, et la famille C 8, La salle de réception comporte 4 tables (T1, T2, T3 et T4) de 5 places chacune. Les familles souhaitent qu'un maximum de personnes puissent manger.

Montrer que ce problème peut se formuler comme un problème de recherche de flot maximum dans un réseau que vous dessinerez (vous préciserez les capacités des arcs du réseau sur le graphe, vous expliquerez pourquoi trouver un plan de table correspond à un flot maximum de ce réseau avec ces capacités précises).

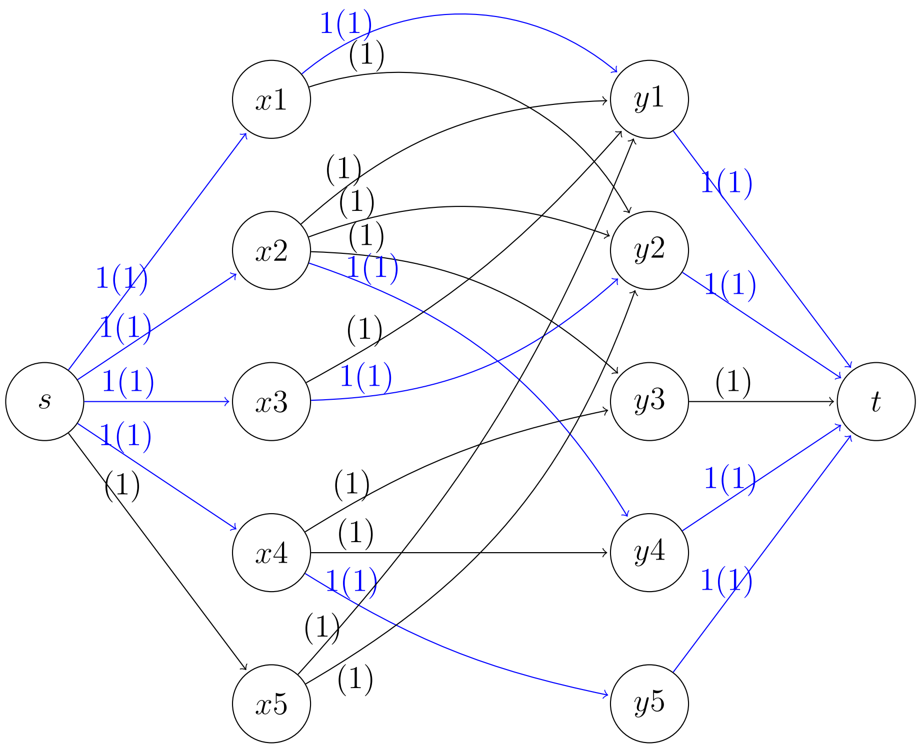
Exemple d'un flot ($\varphi = 19$) pour le problème posé :



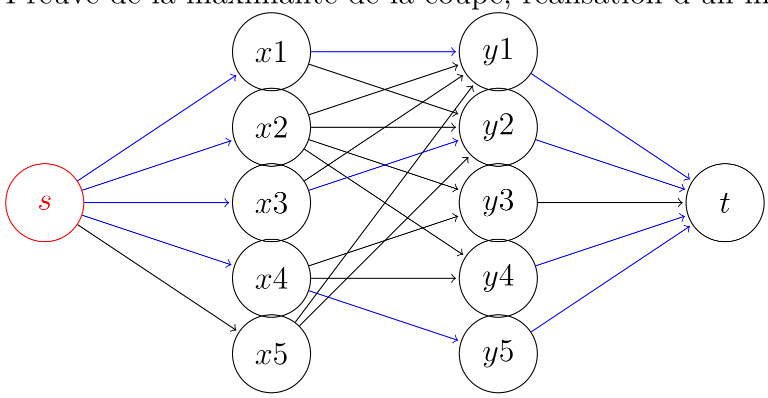
La valeur de la coupe {s} valant $5 + 6 + 8 = 19$, on peut affirmer qu'il s'agit là d'une flot maximal résolvant le problème d'assignation des tables.

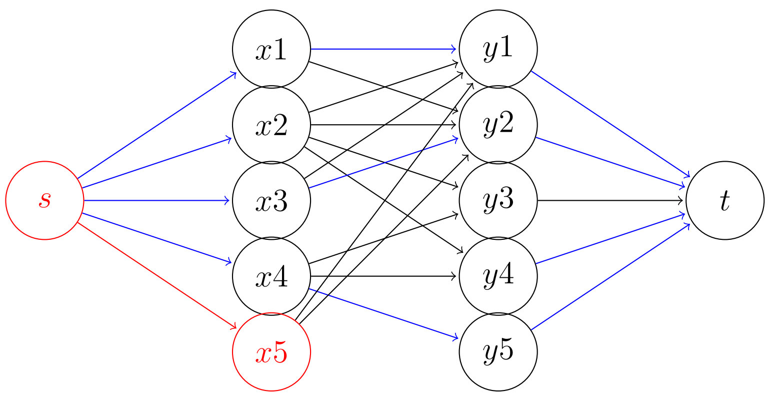
Exercice 5 :

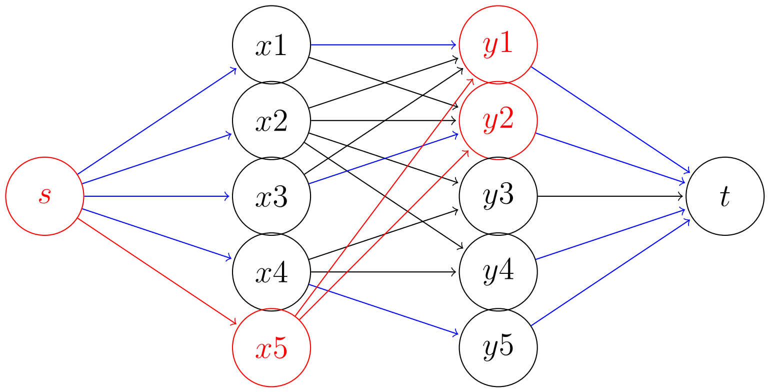
Question 5.1 :
Exemple d'un flot maximal $\varphi^* = 4$:

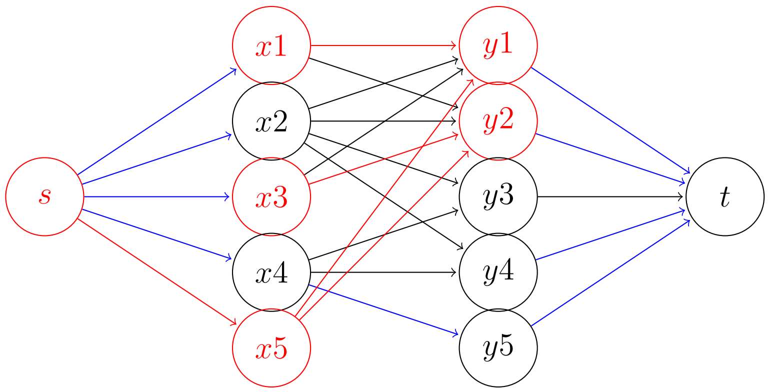


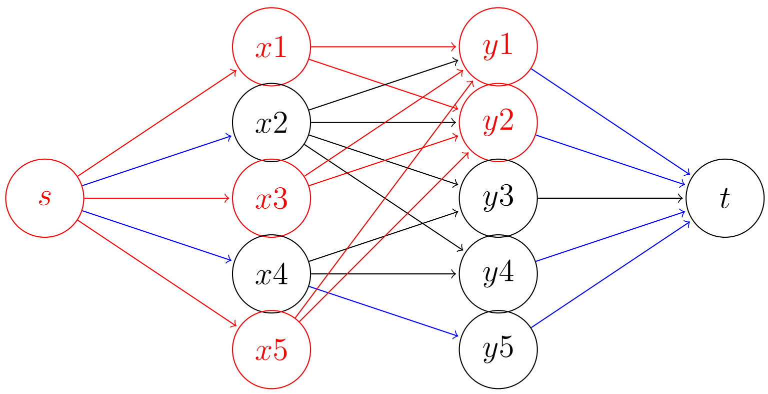
Preuve de la maximalité de la coupe, réalisation d'un marquage :







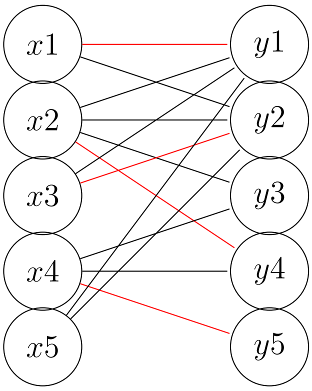




Ce marquage terminé, on obtient la coupe $\{s, x_1, x_3, x_5, y_1, y_2\}$. Cette coupe contient comme arcs sortants : $\{sx_2, sx_4, y_1t, y_2t\}$, soit $\varphi^* = 4$.

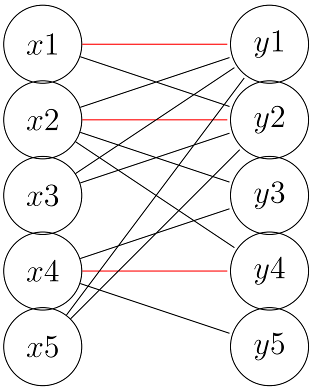
Question 5.2 :
Soit E l'ensemble des arcs de G qui correspondent à des arcs ayant un flux de 1 sur le flot.
Alors, si E contient des arrêtes adjacentes en x_i, le flux sortant vaudrait ≥ 2, ce qui impliquerait que le flux entrant vaut ≥ 2 également, ce qui est impossible puisqu'il n'y a qu'un seul arc entrant de capacité 1.
(Raisonnement similaire pour les y_i.)

Question 5.3 :
Le couplage associé au flot maximal trouvé en 5.1 est : $\{x_1y_1,\ x_2y_4,\ x_3y_2,\ x_4y_5\}$:



Question 5.4 :
Ce couplage est maximal : il n'est pas possible d'y ajouter d'arrêtes.
Il est également maximum : son cardinal vaut 4 (=φ^*).

Question 5.5 :
Soit le couplage suivant : $\{x_1y_1,\ x_2y_2,\ x_4y_4\}$:

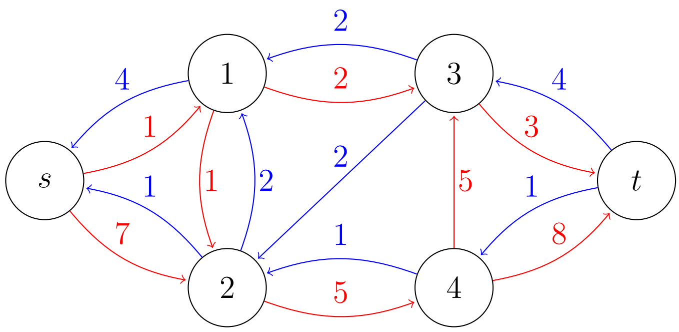


Ce couplage est maximal : il n'est pas possible d'y ajouter d'arrêtes.
Il n'est cependant pas maximum : son cardinal vaut 3 ($< \varphi^* = 4$).

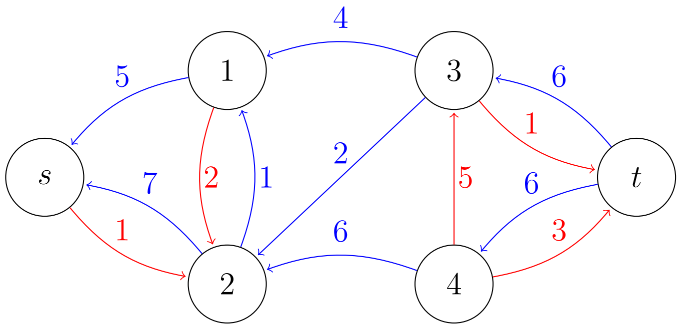
3 Graphe d'écart et flots à coûts

Exercice 7 :

Question 7.1 & 7.2 :



Il existe par exemple la chaîne {s12t} donc le flot n'est pas maximum.

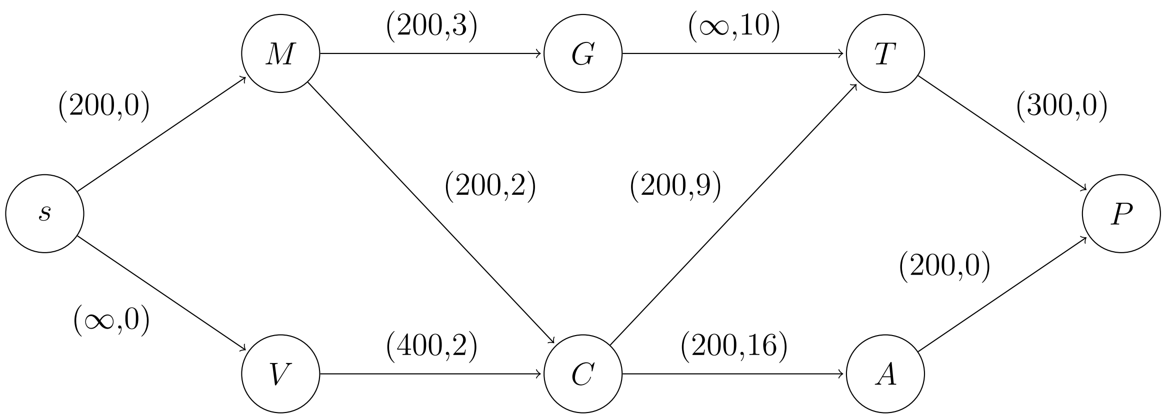


Il n'existe aucune chaîne de s à t, donc le flot est maximum.

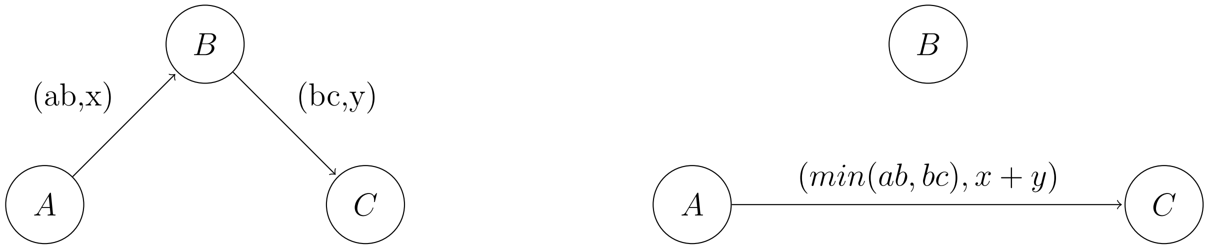
Exercice 8 :

Un grossiste en fleurs assure leur transport depuis Mougins et Vallauris vers Paris. Ce transport est effectué par camionnettes de Vallauris à Cannes et de Mougins à Cannes ou Grasse, puis par train ou avion de Cannes à Paris et par train de Grasse à Paris. On désire étudier le transport global par semaine durant l'été de façon à envoyer un maximum de fleurs à Paris pour un coût minimal.
Les camionnettes disponibles à Vallauris permettent d'acheminer au plus 400 cartons par semaine vers Cannes au coût unitaire de 2€ ; celles de Mougins, 400 cartons dont la moitié vers Grasse au coût de 3€ et l'autre moitié vers Cannes au coût de 2€. De plus, on doit limiter les envois par le train, trop lent, à 300 cartons dont 200 au plus depuis Cannes, le coût unitaire étant alors de 10€ de Grasse et de 9€ de Cannes. Par avion, le coût est de 16€ mais leur nombre ne permet pas d'expédier plus de 200 cartons de Cannes à Paris. Enfin, la production de Mougins ne peut dépasser 200 cartons, celle de Vallauris étant toujours excédentaire.

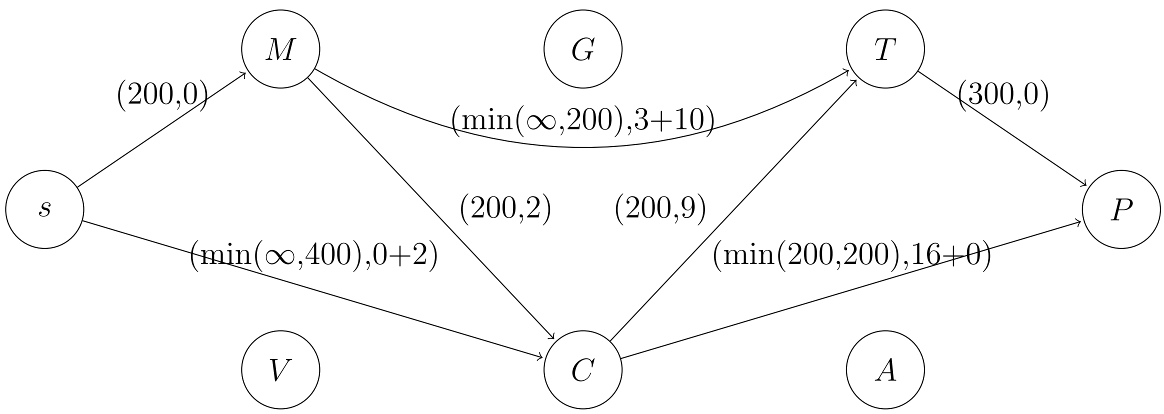
Question 8.1 :



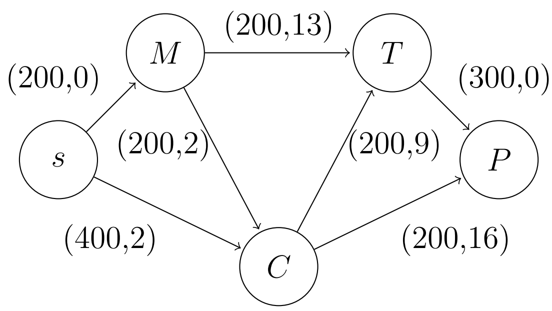
Question 8.2 :
On peut éliminer tous les nœuds ayant 1 arc entrant et un arc sortant. Le simplification se faisant comme suit :



On obtient alors pour le graphe de l'exercice :

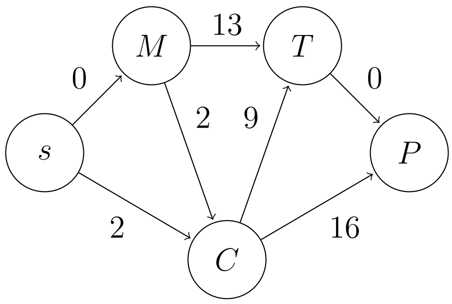


Soit :



Application de l'algorithme *Roy* :

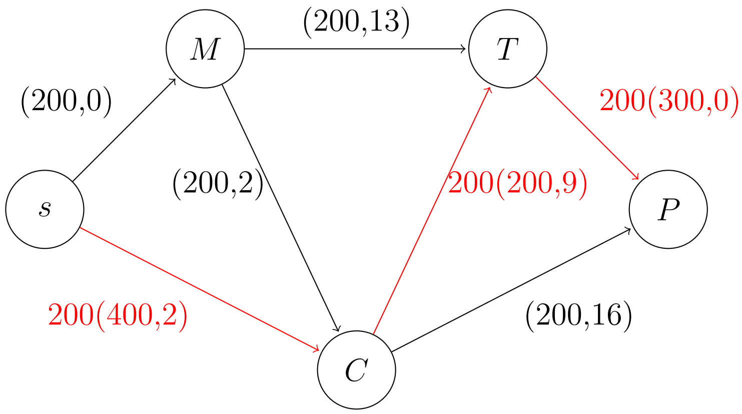
1. On démarre toujours d'un flot nul. De ce fait, le premier choix de chaîne augmentante peut se faire à partir du graphe des coûts, que l'on dessine ci-dessous :



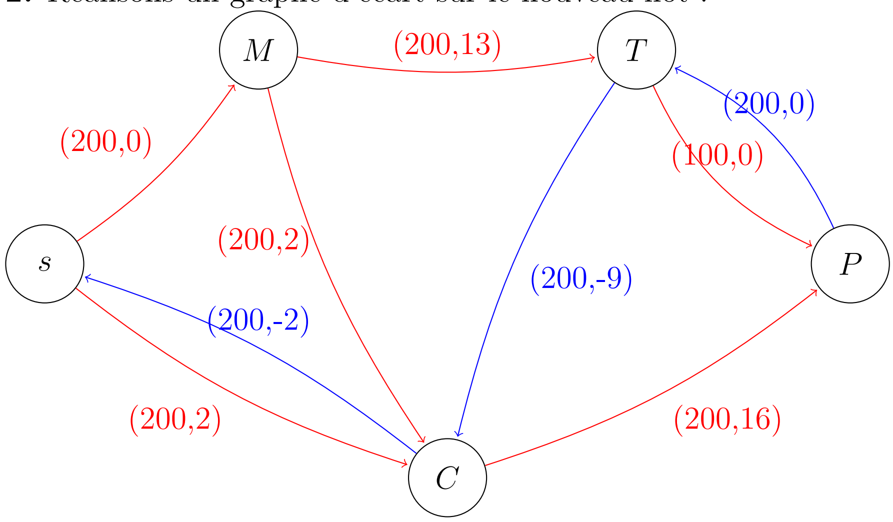
Listons tous les parcours possibles ainsi que leur coût total :

parcours	coût
sMTP	13
sMCTP	11
sMCP	18
sCP	18
sCTP	11

Prenons un premier chemin de coût minimal, par exemple sCTP (on aurait pu prendre sM-CTP). On obtient alors le flot suivant :



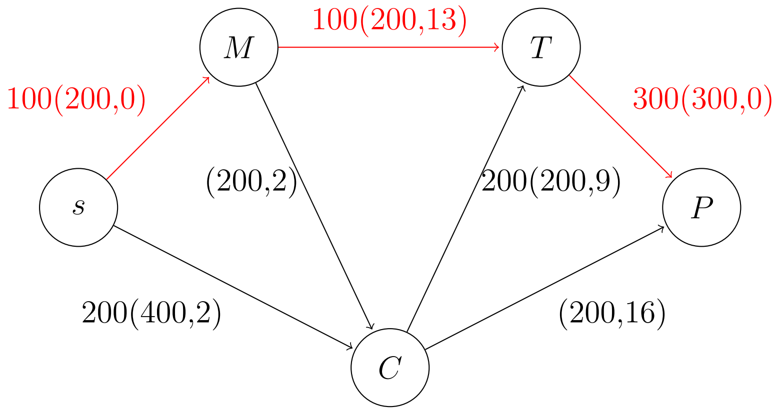
2. Réalisons un graphe d'écart sur le nouveau flot :



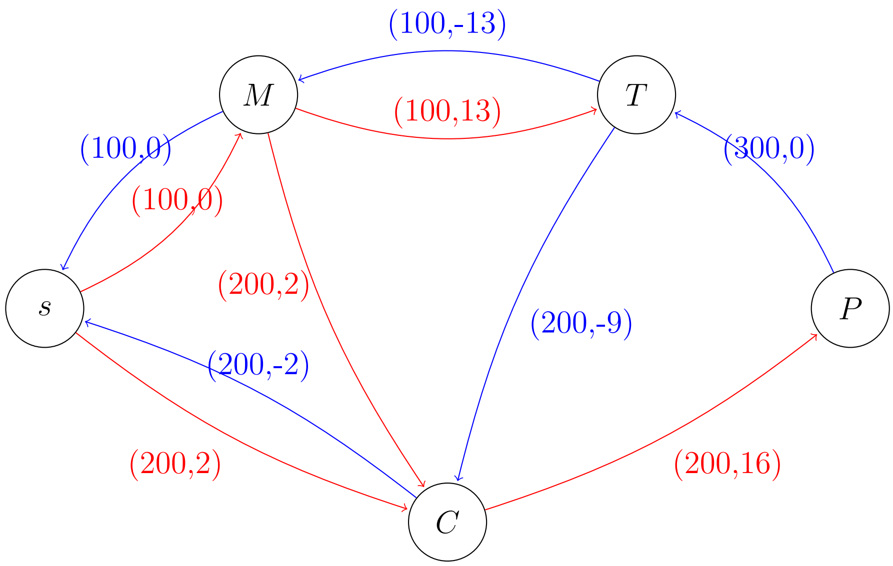
Listons tous les parcours possibles ainsi que leur coût total :

parcours	coût
sMTP	13
sM(TC)P	21
sMCP	18
sCP	18

Prenons sMTP :



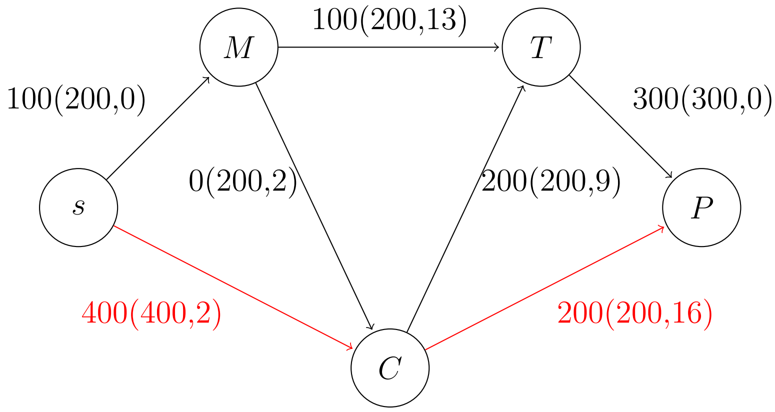
3. Réalisons un graphe d'écart sur le nouveau flot :



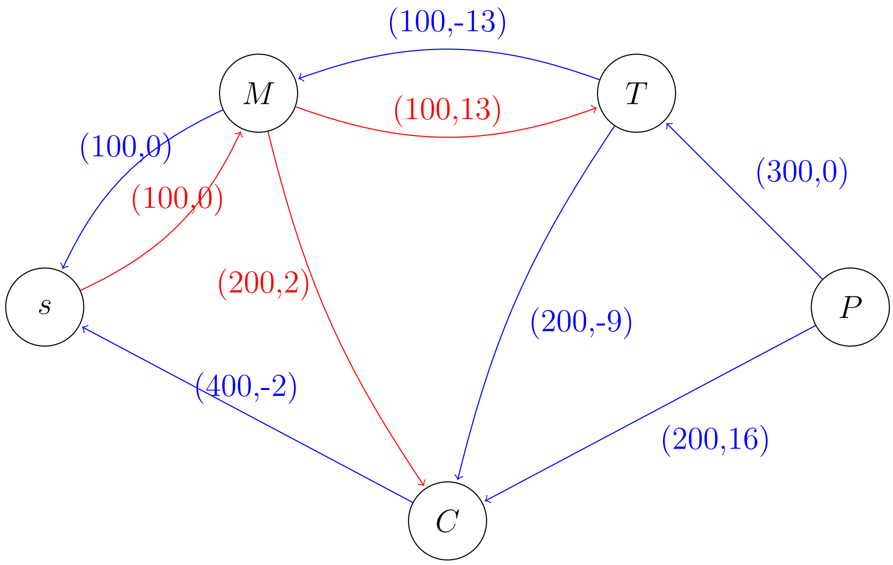
Listons tous les parcours possibles ainsi que leur coût total :

parcours	coût
sMCP	18
sCP	18

Les deux parcours augmentant restants sont de même coût, mais sCP permet d'augmenter le flux de 200 contre 100 avec sMCP, donc prenons sCP :

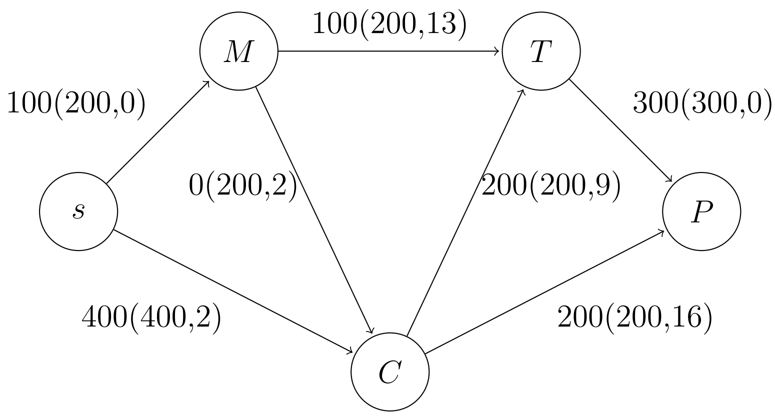


4. Réalisons un graphe d'écart sur le nouveau flot :



Il n'y a plus de chaîne augmentante sur le graphe, l'algorithme est donc terminé.

On obtient donc le flot maximum ($\varphi = 500$) pour un coût minimal suivant:



Le coût total étant :

$$100 * 0(sM) + 400 * 2(sC) + 0 * 2(MC) + 100 * 13(MT) + 200 * 9(CT) + 200 * 16(CP) + 300 * 0(TP)$$
$$= 800 + 1300 + 1800 + 3200 = 7100.$$

Note : cette dernière partie n'a pas été corrigée en TD, mais le résultat est a priori correct.