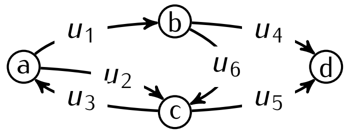
TDs : Flots

1 Loi de Kirchhoff et algorithme de Ford-Fulkerson

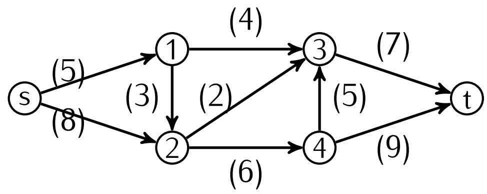


Exercice 1. Question 1.1. Donnez trois exemples de flots (qui ne sont pas des cycles) sur le graphe ci-contre.

Question 1.2. Démontrez la loi de Kirchhoff généralisée : étant donné un flot ϕ sur un graphe orienté (X, U), et un sous-ensemble A de X :

$$\sum_{u\in\omega^+(A)} \varphi(u) = \sum_{u\in\omega^-(A)} \varphi(u).$$

Question 1.3. *(Travail Personnel)* Montrez que toute combinaison linéaire de flots est un flot.



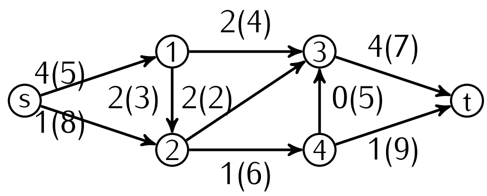
Exercice 2 (Coupe min. et flot max., algo. de Ford-Fulkerson)**.** Soit le réseau ci-contre avec les capacités indiquées entre parenthèse.

Question 2.1. Donnez des exemples de coupes avec leur capacité.

> $(\{s, 1, 3\}, \{2, 4, t\})$ *capacité = 8+3+7 = 18 (on ne prend que les arcs sortants)*

Question 2.2. Quelle est la coupe de capacité minimum ?

> *On essaie de couper un peu partout on ne trouve pas mieux que 12 avec* $(\{s, 1, 2\}, \{3, 4, t\})$.



Question 2.3. On considère le vecteur φ_0 ci-contre, ses composantes sont indiquées sur les arcs suivies par les capacités de ces arcs indiquées entre parenthèses. Vérifiez que φ_0 est un flot sur le réseau de transport associé, et qu'il est compatible, donnez sa valeur $v(\varphi_0)$.

> *On vérifie qu'en tout sommet (y compris s et t du coup on doit ajouter l'arc de retour* u_0 *avec un flux de 5) la loi de Kirchhoff est vérifiée, et que pour tout arc* u, $0 \leq \varphi_0(u) \leq capa(u)$. *(en exam il suffit de dire qu'on a vérifié ça). La valeur de* φ_0, $v(\varphi_0) = \varphi_0(u_0) = 5$

Question 2.4. (*tr. pers.*) Montrez que tout flot a une valeur inférieure ou égale à la capacité de toute coupe
Indice : Vous considérerez un flot compatible quelconque φ *et une coupe quelconque* $C(A, X \setminus A)$ *sur le même réseau* $R(X, U)$*, vous exprimerez la loi de Kirchhoff généralisée sur* A*. Vous mettrez ensuite en évidence l'arc de retour* u_0 *dans la précédente équation, en utilisant le fait que le flot est compatible vous pourrez conclure..*

> *Loi de Kirchhof généralisée,*
>
> $$\sum_{u\in\omega^+(A)} \varphi(u) = \sum_{u\in\omega^-(A)} \varphi(u)$$
>
> u_0 *est un arc entrant dans A (qui contient s et pas t), donc :*
>
> $$\sum_{u\in\omega^+(A)} \varphi(u) = \sum_{u\in\omega^-(A), u\neq u_0} \varphi(u) + \varphi(u_0)$$
>
> φ *est un flot compatible* $\forall u \in U, 0 \leq \varphi(u) \leq capa(u)$.
> *Donc, en particulier,* $\sum_{u\in\omega^-(A),u\neq u_0} \varphi(u) \geq 0$ *et* $\sum_{u\in\omega^+(A)} \varphi(u) \leq \sum_{u\in\omega^+(A)} capa(u)$. *D'où,* $\varphi(u_0) \leq \sum_{u\in\omega^+(A)} capa(u) = C(\omega^+(A))$.

Question 2.5. Trouvez un flot maximum par marquage Ford Fulkerson, décrivez dans un tableau avec une colonne par arc, les vecteurs cycles (associés aux chaînes augmentantes utilisées) ainsi que les flots obtenus successivement.

On considère d'abord la chaîne augmentante $c_1 = [s, 1, 3, t]$ on ajoute u_0 pour avoir le cycle associé μ_1, on peut augmenter de $\varepsilon = 1$ sur cette chaîne sans dépasser les capacités : en effet $\varepsilon = \min(\varepsilon_1, \varepsilon_2)$ où $\varepsilon_1 = \min_{u \in \mu_1^+} capa(u) - \varphi(u) > 0$ et $\varepsilon_2 = \min_{u \in \mu_1^-} \varphi(u) > 0$ avec $\min_\emptyset = \infty$ (μ_1^+ est l'ensemble des arcs orientés dans le sens de μ_1 (toujours non vide car $u_0 \in \mu_1^+$) et μ_1^- est l'ensemble des arcs orientés dans le sens inverse du parcours de μ_1). La valeur de φ_1 est supérieure à φ puisque $\varphi_1(u_0) = \varphi(u_0) + \varepsilon.\mu(u_0) = \varphi(u_0) + 1 \times \varepsilon$. $\varphi_1 = \varphi_0 + \mu_1$. φ_1 est une combinaison linéaire de flots car tout vecteur cycle est un flot.
Ce flot est compatible (conséquence du choix de $\varepsilon = 1$).

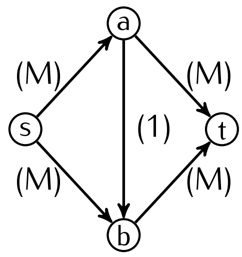
arcs	(s,1)	(s,2)	(1,2)	(1,3)	(2,3)	(2,4)	(3,t)	(4,3)	(4,t)	u_0=(t,s)
φ_0	4	1	2	2	2	1	4	0	1	5
μ_1	1			1			1			1
$\varphi_1 = \varphi_0 + \mu_1$	5	1	2	3	2	1	5	0	1	6
...										

Puis la chaîne [s,2,4,t] on peut l'augmenter de 5. Puis la chaîne [s,2,1,3,t] on peut l'augmenter de 1. On obtient une solution de 5,7,1,4,2,6,0,6,6 de valeur 12.

Question 2.6. (*travail personnel*) Soit φ_k le flot obtenu à la dernière étape de l'algorithme de Ford Fulkerson. Soit A l'ensemble des sommets que l'on peut marquer pour le flot φ_k. Montrez que $\varphi_k(u_0) = capa(\omega^+(A))$ et donc que φ_k est maximum.
Indice : considérez les arcs de $\omega^+(A)$ et les arcs de $\omega^-(A)$.

À la dernière étape de l'algo, on ne peut pas marquer la sortie ça veut dire que les arcs sortants de l'ensemble des sommets marqués A sont tous saturés, ça veut dire que le flot qui passent dans ces arcs = capacité des arcs. L'ensemble des sommets marqués contient s et pas t, on peut considérer la coupe $(A, X \setminus A)$ la capacité de cette coupe est égale au flot car ses arcs sont saturés et tout ce qui entre = ce qui sort donc ce qui passe dans cette coupe est = ce qui passe dans le réseau, donc à ce qui passe en u_0. ainsi $\varphi_k(u_0) = capa(\omega^+(A))$, or $\varphi_k(u_0) = v(\varphi_k)$ on a donc trouver un flot dont la valeur est égale à la capacité d'une coupe, d'après le théorème de la coupe il est donc maximum.

Question 2.7. *(Travail Personnel)* Il peut arriver qu'en choisissant mal les chaînes augmentantes, on ait besoin de beaucoup d'itération du marquage Ford-Fulkerson. On considère le graphe ci-contre (Edmonds et Karp 1972).Montrez qu'en choisissant les chaînes augmentantes d'une certaine façon, on peut avoir à faire $2M$ étapes de marquage avant d'obtenir un flot maximum (Ce qui rend l'algorithme dépendant de la valeur du flot, et donc non polynomial).



Valeur du flot max=2M, en choisissant la chaîne sabt puis sbat on augmentera de 1 le flot 2M fois, au lieu de trouver le flot max en deux chaînes augmentantes sat et sbt. Notons que le problème reste polynomial puisqu'on peut modifier l'algo en le forçant à choisir les chaînes augmentantes les plus courtes en nombre d'arcs.

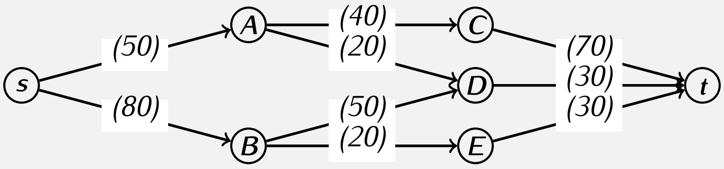
2 Modélisation

Exercice 3 (Châteaux d'eau). On considère 2 châteaux d'eau A et B alimentant 3 villages C, D et E. Le château d'eau A peut être alimenté avec 50 litres par secondes, et B avec 80 litres par seconde. Le village C a besoin au maximum de 70 l/s, D de 30 l/s et E de 30 l/s. Les capacités des canalisations en l/s sont :

canalisation	AC	AD	BD	BE
capacité en l/s	40	20	50	20

Question 3.1. On cherche une répartition de l'eau qui satisfasse au mieux ces contraintes, exprimer ce problème en termes de flot dans un réseau à dessiner.

On considère une source fictive s qui alimente A et B avec une capacité de débit max de 50 et 80 l/s. On considère que la consommation des villages est représentée par un écoulement vers un puits t avec une capacité de débit max respective de 70, 30 et 30.



Il s'agit de trouver un flot compatible maximum : flot car ce qui entre en chaque point du réseau devra en ressortir sans perte ni ajout, loi de Kirchhoff, maximum car on cherche à satisfaire au mieux les demandes de consommation des villages, on veut donc maximiser ce qui passe en Ct, Dt et Et sans dépasser les demandes (compatible). Il faut qu'il soit compatible avec les capacités physiques des canalisations existantes et avec les possibilités d'alimentation des châteaux d'eau.

Question 3.2. Le vecteur φ suivant est-il un flot compatible ? Quelle est sa valeur ?

canalisation	sA	sB	AC	AD	BD	BE	Ct	Dt	Et
φ	50	30	30	20	10	20	30	30	20

Vérifie la loi de Kirchhoff en tout sommet, si on ajoute l'arc de retour avec 80 dessus : donc flot. Compatible car $\forall$ arc u, $0 \leq \varphi(u) \leq capa(u)$. valeur=80.

Question 3.3. Déterminer un flot maximum grâce à l'algorithme de Ford-Fulkerson.

chaine sBDACt sur laquelle on peut augmenter de 10. donne un flot de 90.

Question 3.4. Tous les villages pourront-ils subvenir à leurs besoins ?

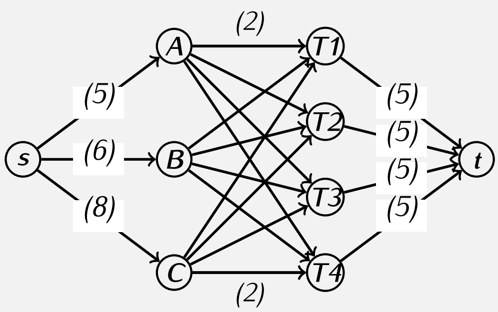
non C et E n'auront pas ce qu'ils ont demandé mais n'ont pas mis d'assez grosse canalisations (AC et BE devraient être changées).

Exercice 4 (Lorsque dîner est un problème !)**.** 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 puisse manger.

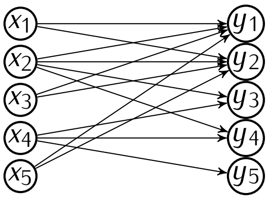
Montrer que ce problème peut se formuler comme un problème de recherche de flot maximum dans un réseau que vous dessinerez (vous préciserez les capacités des arcs du réseau sur le graphe, vous expliquerez pourquoi trouver un plan de table correspond à un flot maximum dans ce réseau avec ces capacités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

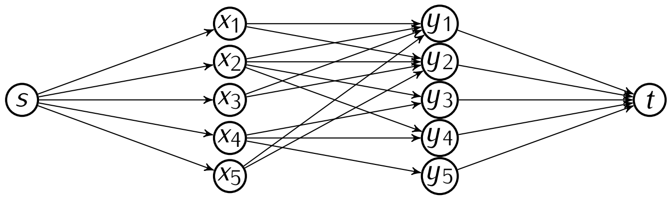
On ne demande pas de résoudre le problème mais seulement de le représenter.

On peut proposer la formalisation suivante (inverser les arcs et inverser s et t est aussi une solution recevable) avec des capacités de 2 sur les arcs du milieu. Il s'agit que chaque personne d'une famille soit placée à une table, ce qui entre en A= nombre de personne que l'on doit placer de la famille A. Ce qui entre en T_i = nombre de personne assises à la table T_i. De s vers une famille au max nb personnes de la famille, de T_i vers t : au max nombre de places assises. On cherche un flot compatible de valeur max : compatible car respect des contraintes (nb de membre de chaque famille, nb max de membre par table, nb max de places par tables), max car placer le plus de personnes en tout.



Exercice 5 (Graphe biparti)**.** Soit G le graphe biparti (à gauche) et R_G le réseau de transport biparti "associé" (à droite) dans lequel on suppose les capacités $capa(u)$ toutes égales à 1.





Question 5.1. Trouver un flot maximum φ^* dans R_G.

on part d'un flot nul, on arrive facilement à un flot de valeur 3, par exemple en utilisant les chaînes augmentantes sx1y1t, sx2y2t, sx4y3t. Puis on trouve une chaine augmentante sx3y2x2y4t puis on ne peut plus marquer la sortie (les sommets marqués $A = \{s, x1, x3, x5, y1, y2\}$ forment une coupe (A,X $\setminus$ A) de capa 4). Le flot max φ^* obtenu est de valeur 4.

Un couplage d'un graphe biparti est un ensemble d'arcs (ou d'arêtes) 2 à 2 non adjacent(e)s.

Question 5.2. Vérifier (et expliquer) que pour tout flot φ sur R_G, l'ensemble E des arcs u de G en lesquels le flux $\varphi(u)$ est égal à 1 est un couplage de G.

Pour tout arc $u = (x, y)$ de G, si $\varphi(x, y) = 1$ dans R_G alors comme $\varphi(s, x) \leq 1$ (puisque les capacités des arcs sont de 1 sur tous les arcs) on peut en déduire d'après la loi de Kirchhoff appliquée au sommet x que $\forall z \neq y$, $\varphi(x, z) = 0$. De même en utilisant la loi de Kirchoff en y et sachant que $\varphi(y, t) \leq 1$, on a $\forall v \neq x$, $\varphi(v, y) = 0$. Ainsi E ne contient pas d'arcs adjacents.

Question 5.3. Donnez le couplage associé au flot maximum φ^*, quel est sa cardinalité ?

$E = \{(x1, y1), (x2, y4), (x3, y2), (x4, y3)\}$ cardinalité 4.

On appelle *couplage maximum* un couplage de cardinalité maximum parmi tous les couplages possibles, on appelle *couplage maximal* un couplage E tel que l'ajout de tout nouvel arc de G à E n'est plus un couplage. On appelle *couplage parfait* un couplage tel que tout sommet est adjacent à au moins un arc du couplage.

Question 5.4. Conclure sur le couplage obtenu.

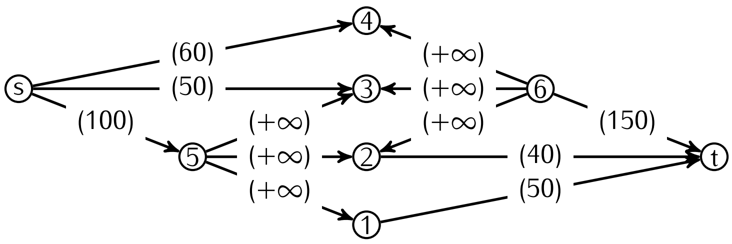
Le couplage obtenu est de cardinalité maximum car la cardinalité du couplage est la valeur du flot qui est maximum, il est également maximal puisqu'on ne peut plus augmenter le flot. Il n'est pas parfait.

Question 5.5. (*Travail personnel*) Proposez un couplage maximal non maximum.

$\{(x1, y1)(x2, y2), (x4, y4)\}$

Exercice 6 (Mine à ciel ouvert)**.** On considère le graphe ci-contre :

Question 6.1. Calculez un flot compatible maximum de s à t dans le réseau associé à ce graphe.



chaine augmentante s52t on peut faire passer 40, chaine augmentante s51t on peut faire passer 50, on ne peut plus marquer la sortie avec un marquage de Ford-Fulkerson depuis s donc flot max de valeur 90.

Question 6.2. Décrivez une coupe de capacité minimum ainsi que ses arcs et sa capacité.

coupe $C = \{s, 5, 1, 2, 3, 4\}, \{6, t\}$, capacité 90, arcs=$\{(2, t), (1, t)\}$.

Ce graphe est un codage d'un problème d'exploitation de Mines à Ciel Ouvert. Ce problème est un cas particulier parmi les problèmes de sélection d'ensembles avec des contraintes de dépendances. On désire extraire un ensemble de blocs qui aura le meilleur profit.

Pour des raisons de sécurité, pour retirer un bloc on doit d'abord retirer les 3 blocs qui sont au-dessus de lui (directement sur lui, au-dessus à gauche et au-dessus à droite). On considère la "mine" dessinée ci-dessous accompagnée des profits des blocs : pour extraire le bloc 5, il faudra extraire les blocs 1, 2 et 3.

1	2	3	4
	5	6	

bloc i	1	2	3	4	5	6
profit p_i (revenu-coût d'extraction)	-50	-40	50	60	100	-150

Le profit associé à un bloc est p_i (c'est le revenu du bloc moins le coût de son extraction). Le profit associé à un ensemble de blocs est la somme des profits de ces blocs. Ici les sommets du graphe représentent les blocs, et pour tout bloc i dont l'extraction nécessite celle de j, on crée un arc (i,j) de capacité $(+\infty)$, pour tout bloc i ayant un profit p_i positif, un arc (s,i) de capacité (p_i) et pour tout bloc j ayant un profit p_j négatif, un arc (j,t) de capacité $(-p_j)$.

Question 6.3. Soit profit(A) la somme des profits des blocs appartenant à A. Montrez que toute coupe $C = (A, X \setminus A)$ de capacité finie vérifie profit$(A) = \sum_{i|p_i \geq 0} p_i - \text{capa}(C)$.

Une coupe $C = (A, X \setminus A)$ de capacité finie ne contient aucun arc de capacité infinie, donc aucun arc (x,y) entre deux blocs x et y, il faut une coupe contenant seulement des arcs (s,x) et (y,t). Les arcs (s,x) qui appartiennent à C vérifient que $x \notin A$ (car par convention $s \in A$), pour les arcs (y,t) ils vérifient $y \in A$ (car par convention $t \in A$). D'où,

$$\begin{aligned} capa(C) &= \textstyle\sum_{x \notin A} capa(s,x) + \sum_{y \in A} capa(y,t) \\ &= \textstyle\sum_{x \notin A, p(x) \geq 0} p(x) - \sum_{y \in A, p(y) < 0} p(y) \end{aligned}$$

or $\sum_{x, p(x) \geq 0} p(x) = \sum_{x \notin A, p(x) \geq 0} p(x) + \sum_{x \in A, p(x) \geq 0} p(x)$
donc $capa(C) = \sum_{x|p(x) \geq 0} p(x) - \sum_{x \in A, p(x) \geq 0} p(x) - \sum_{y \in A, p(y) < 0} p(y)$
c'est-à-dire $capa(C) = \sum_{x|p(x) \geq 0} p(x) - \sum_{x \in A} p(x)$
donc $profit(A) = \sum_{x|p(x) \geq 0} p(x) - capa(C)$.

Question 6.4. En déduire les blocs à extraire pour obtenir un profit maximum, et le profit obtenu.

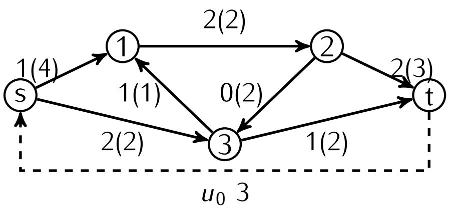
grâce à l'équation de la question 3, le profit est max quand la coupe est de capa min (puisque la somme des $p_i > 0$ est une constante). Donc on cherche la coupe de capacité min : c'est celle obtenue à la question 2), $C = (\{s, 1, 2, 3, 4, 5\}, \{6, t\})$, capacité 90, les blocs à extraire sont donc 1, 2, 3, 4 et 5, le profit max est $(50 + 60 + 100) - 90$ c'est-à-dire 120.

3 Graphe d'écart et flots à coûts

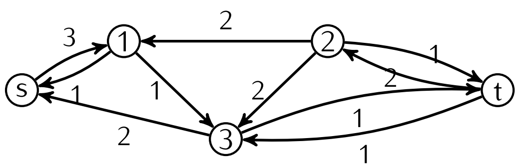
Le graphe d'écart $G_R(\varphi) = (X, U_\varphi, r_\varphi)$ associé à un réseau de transport $R = (X, U, capa)$ et à un flot compatible φ est défini par : tout arc $(x,y) \in U$ peut induire deux arcs dans U_φ :

- un arc dit "direct" : $(x,y) \in U_\varphi$ si $\varphi(x,y) < capa(x,y)$ avec $r_\varphi = capa(x,y) - \varphi(x,y)$
- un arc dit "indirect" : $(y,x) \in U_\varphi$ si $\varphi(y,x) > 0$ avec $r_\varphi = \varphi(y,x)$

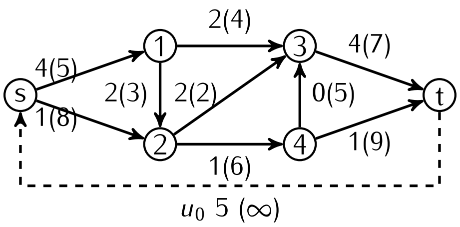
Réseau R initial avec son flot φ

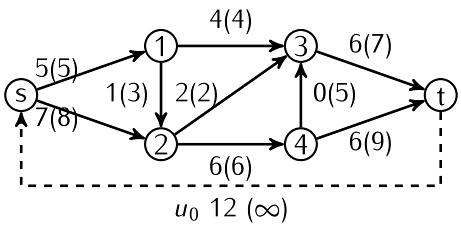


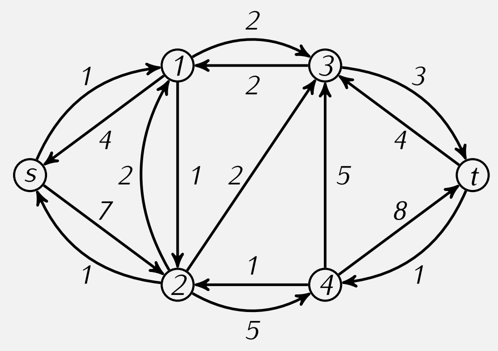
Graphe d'écart $G_R(\varphi)$



Exercice 7. Question 7.1. Dessinez les graphes d'écart associés au réseau suivant pour les deux flots proposés.







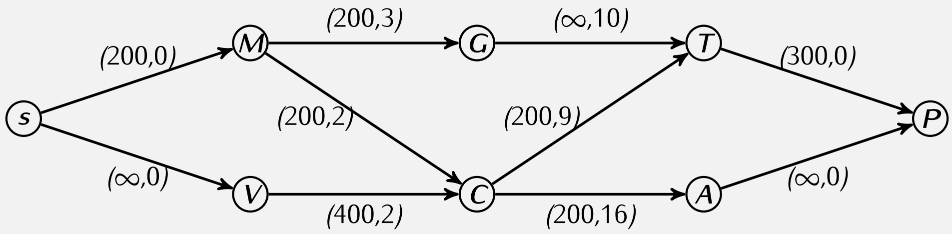
Question 7.2. Existe-t-il un chemin de s à t dans les deux graphes d'écart ? Que peut-on en déduire ?

il y a une bijection entre les chemins dans les graphes d'écarts et les chaines augmentantes dans le réseau, s'il n'y a pas de chemin (cas de φ_1) alors il n'y a pas de chaîne augmentante, on ne peut donc pas augmenter le flot donc il est max dans le cas contraire on peut augmenter (φ_0 est non max).

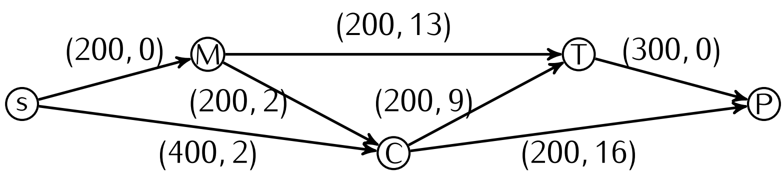
Exercice 8 (Les fleurs c'est périssable). Un grossiste en fleurs assure leur transport depuis Mougins et Vallauris vers Paris. Ce transport est effectué par camionnettes de Vallauris à Cannes et de Mougins à Cannes ou Grasse, puis par train ou avion de Cannes à Paris et par train de Grasse à Paris. On 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 lents, 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. Tracer un graphe de 8 sommets représentant le problème : sommets à créer s (entrée), V, M, G, C et P pour les cinq villes, T et A pour train et avion. Valuer les arcs (x,y) par des couples $(capa(x, y), \gamma(x, y))$, $capa(x, y)$ indiquant la capacité de l'arc et $\gamma(x, y)$ le coût unitaire de transport associé ; pour certains arcs $capa(x, y)$ pourra être infini.



Question 8.2. Quelles sont les simplifications à faire pour aboutir au graphe G ci-contre ? Les justifier.



quand un sommet n'a qu'un seul arc entrant et sortant on peut les fusionner avec min capa et somme des coûts.

Question 8.3. Déterminer un flot maximal de coût minimal avec l'algorithme de Roy, Busacker et Gowen.

on augmente de 200 sur sCTP (cout 11x200), puis 100 sur SMTP (cout 13x100 de plus), puis 200 sur SCP (cout 18x200 de plus). Flot max de 500 et de coût 7100.

Algorithm Roy (1960) [Roy69], Busacker and Gowen (1961) [BG61]
Max flow of min cost of a transport network $R = (X, U, capa, \gamma)$

1. Initialization : $\varphi \leftarrow (0, \ldots, 0)$ (null flow)
2. Current Step : Build the residual graph[a] $G_R(\varphi) = (X, U_\varphi, capa_\varphi, \gamma_\varphi)$ with $\gamma_\varphi(u) = \begin{cases} \gamma(u) & \text{if } u \text{ is a direct arc} \\ -\gamma(u) & \text{if } u \text{ is an indirect arc} \end{cases}$
3. If $\nexists$ directed path $[s, t]$ in $G_R(\varphi)$ Then END : φ is a max flow of min cost
   Else | find an elementary st-minimal directed path ν_{st} in $(X, U_\varphi, \gamma_\varphi)$
   $k \leftarrow \min_{u \in \nu_{st}} capa_\varphi(u)$
   $\mu_{st} \leftarrow \nu_{st} \cup \{u_0\}$
   $\varphi \leftarrow \varphi + k\mu_{st}$; $\gamma(\varphi) \leftarrow \gamma(\varphi) + k.\gamma_\varphi(\nu_{st})$
4. Reiterate 2 and 3 in order to obtain a flow of higher value

[a] "Residual graph" signifie "Graphe d'écart"

Références

[BG61] R. Busacker and P. Gowen. A procedure for determining minimal-cost network flow patterns. Technical Report ORO-15, Operational Research Office, John Hopkins University, 1961.

[Roy69] B. Roy. *Algèbre moderne et théorie des graphes*. Dunod, Paris, 1969.