Licence Informatique — Algorithmique — Mai 2007
Durée 3 heures — documents et calculatrices interdits

Prenez le temps de faire les calculs au brouillon avant de les recopier sur votre copie d'examen. Efforcez–vous de traiter les questions dans l'ordre. Indiquez bien le numéro de chaque question avant la réponse. Les copies mal présentées seront pénalisées.

1 Flot et gestion de stock (5 points)

On s'intéresse à l'activité d'une entreprise sur deux mois, désignés par les numéros 1 et 2. Chaque mois, l'entreprise fabrique entre 0 et 10 produits et les vend. Les ventes sont comprises entre 0 et 10 aussi. L'entreprise peut aussi stocker des produits d'un mois sur l'autre. Le stock initial et le stock final sont compris entre 0 et 5. En raison d'un aménagement des entrepôts, la capacité des stocks est réduite pendant le mois 1. Le réseau de transport (incomplet) suivant modélise cette activité. Il comporte trois sources et trois destinations.

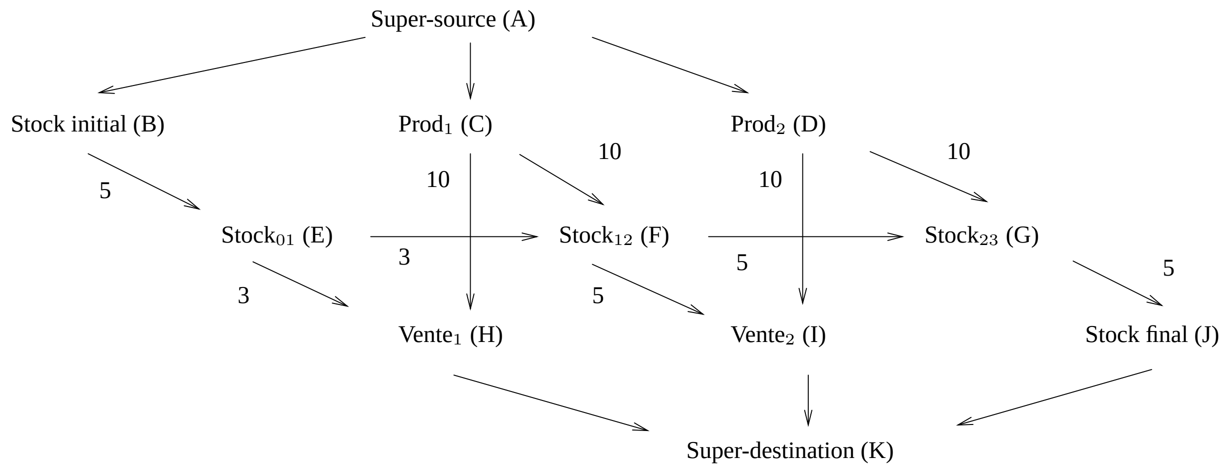
Question 1 [1 pt]. Déduire de l'énoncé les capacités à donner aux arcs partant de la super–source et à ceux arrivant sur la super–destination.

> SOLUTION. Donner la capacité 10 aux arcs $A \to C$, $A \to D$, $H \to K$ et $I \to K$. Donner la capacité 5 aux arcs $A \to B$ et $J \to K$.

Question 2 [4 pts]. Appliquer l'algorithme d'Edmonds–Karp sur le réseau ainsi obtenu. À chaque itération, déterminer la chaîne améliorante la plus courte possible. Au cas où deux chaînes améliorantes de même longueur seraient disponibles, choisir celle qui passe le plus à gauche possible dans le réseau.

Utiliser la feuille en annexe de l'énoncé. À chaque itération, surligner la chaîne et la valeur du flot avant augmentation des flux. Préciser si la chaîne comporte un arc indirect. À la fin de l'algorithme, indiquer la coupe exhibée par l'algorithme. Que peut–on dire de la capacité de cette coupe ?

> SOLUTION. Valeur du flot 0. Chaîne $A \to C \to H \to K$. Valeur du flot 10. Chaîne $A \to D \to I \to K$. Valeur du flot 20. Chaîne $A \to B \to E \to F \to G \to J \to K$. Valeur du flot 23. Chaîne $A \to B \to E \to H \leftarrow C \to F \to G \to J \to K$ (un arc indirect). Valeur du flot 25. Pas de chaîne améliorante. La coupe exhibée par l'algorithme coupe les trois arcs partant de A. Sa capacité est égale à la valeur du flot.



2 Complexité (8 points)

La procédure en pseudo–code suivante est paramétrée par un graphe orienté G et un sommet distingué s de G. On note n le nombre de sommets et m le nombre d'arcs de G. La variable locale E est un ensemble de sommets.

```
procédure chmorglurs (G, s)
début
  colorier tous les sommets en bleu, sauf s
  colorier s en vert
  E := {s}
  tant que E ≠ ∅ faire                                  boucle extérieure
    choisir un sommet x ∈ E au hasard et le retirer de E
    pour tous les successeurs y de x faire             boucle intérieure
      si y est bleu alors
        colorier y en vert
        E := E ∪ {y}
      finsi
    fait
    colorier x en rouge
  fait
fin
```

Les opérations d'ensembles utilisées dans la procédure sont :

1. l'initialisation : $E := \{s\}$
2. le test d'égalité avec l'ensemble vide
3. la sélection et le retrait d'un élément $x \in E$ pris au hasard
4. l'union avec un singleton : $E := E \cup \{x\}$

Chacune de ces opérations a une complexité en temps, dans le pire des cas, en $O(n)$ (on suppose la structure de données implantée naïvement). Chaque opération de coloriage d'un sommet a une complexité en temps, dans le pire des cas, en $O(1)$.

Question 3 [1 pt]. Que fait la procédure ? Que peut–on dire au sujet des sommets rouges, à la fin de l'exécution de la procédure.

> SOLUTION. La procédure réalise un parcours du graphe G à partir du sommet s. À la fin de l'exécution, les sommets rouges sont les sommets accessibles à partir de s.

Question 4 [1 pt]. Donner un invariant de la boucle extérieure concernant les sommets verts et l'ensemble E.

> SOLUTION. Un sommet est vert si et seulement s'il appartient à E.

Question 5 [1 pt]. Donner une estimation de la complexité en temps, dans le pire des cas, de la séquence des trois premières instructions (initialisations).

> SOLUTION. $O(n) + O(1) + O(n) = O(n)$.

Question 6 [1 pt]. Combien de fois un sommet peut–il, au plus, être inséré dans E ? Justifier en une phrase ou deux. En déduire une estimation (majoration) du nombre d'itérations effectuées par la boucle extérieure.

> SOLUTION. Au plus une fois. En effet, un sommet ne peut être inséré dans E que s'il est bleu. Un sommet inséré dans E n'est plus bleu. On en déduit que le nombre d'itérations de la boucle extérieure est en $O(n)$.

Question 7 [1 pt]. Déduire de la question précédente une estimation de la complexité en temps, dans le pire des cas, de l'ensemble de toutes les exécutions de la première instruction de la boucle extérieure (sélection et retrait d'un élément $x \in E$).

> SOLUTION. Complexité en $O(n^2)$.

Question 8 [1 pt]. En supposant le graphe implanté par des listes de successeurs, donner une estimation du nombre total d'itérations effectuées par la boucle intérieure, c'est–à–dire le nombre total de fois où la condition du si est évaluée. Justifier en une phrase.

> SOLUTION. Le nombre total d'itérations de la boucle intérieure est en $O(m)$. Passe pour $O(n^2)$. En effet, cette boucle parcourt *in fine* tous les successeurs de tous les sommets accessibles à partir de s (cf. question précédente).

Question 9 [1 pt]. En déduire une estimation de la complexité en temps, dans le pire des cas, de l'ensemble de toutes les exécutions du corps de la boucle intérieure.

> SOLUTION. Nombre d'itérations en $O(m\,n)$. Passe pour $O(n^3)$.

Question 10 [1 pt]. En déduire une estimation de la complexité en temps, dans le pire des cas, de la procédure *chmorglurs*.

SOLUTION. Complexité en $O(n) + O(m\,n) + O(n^2) = O(m\,n + n^2)$. Passe pour $O(n) + O(n^3) + O(n^2) = O(n^3)$.

3 AMPL (8 points)

On s'intéresse à l'activité d'un artisan sur $N = 5$ mois. Chaque mois, l'artisan fabrique des produits (d'un seul type) et les vend. Le tableau suivant donne le prix de vente et le nombre maximal de produits qu'il est possible pour les N mois.

mois	prix de vente (euros/produit)	vente maximale (produits)
1	110	13
2	90	15
3	120	11
4	125	12
5	140	12

L'artisan peut aussi stocker des produits d'un mois sur l'autre mais stocker un produit d'un mois sur l'autre coûte 30 euros par produit. Initialement, 4 produits sont disponibles en stock. L'artisan souhaite maximiser son profit. L'extrait de code AMPL ci–dessous constitue un début de modélisation. Attention au sens de la variable *qte_stock* :

$$\textit{qte\_stock}[m] = \text{le nombre de produits mis en stock } \underline{\text{à la fin du mois } m}.$$

```
param N integer > 0;
set MOIS := 1 .. N;

param stock_initial >= 0;
param cout_stock >= 0;
param prix_vente {MOIS} >= 0;
param vente_max {MOIS} >= 0;

var qte_produite {MOIS} >= 0;
var qte_vendue {m in MOIS} >= 0, <= vente_max [m];
var qte_stock {0 .. N} >= 0;
```

Question 11 [1 pt]. Écrire l'objectif économique (ne compter le coût des quantités en stock que pour $m \geq 1$).

SOLUTION.

```
maximize profit :
    (sum {m in MOIS} qte_vendue [m] * prix_vente [m]) -
    (sum {m in MOIS} qte_stock [m] * cout_stock);
```

Question 12 [1 pt]. Écrire la contrainte qui fixe la valeur du stock initial.

SOLUTION.

```
subject to stock_initial_impose :
    qte_stock [0] = stock_initial;
```

Question 13 [1 pt]. Écrire la contrainte qui exprime la relation entre les quantités produites, vendues et celles mises en stock.

SOLUTION.

```
subject to conservation {m in MOIS} :
    qte_stock [m-1] + qte_produite [m] = qte_stock [m] + qte_vendue [m];
```

3.1 Limitation de la production

Dans le modèle précédent, aucune limite n'est fixée à la production de chaque mois.

Question 14 [1 pt]. Que peut–on dire du modèle ? Est–il non borné ? S'il est borné, est-ce que le polygone des solutions réalisables est ouvert ? fermé ? Justifier en une phrase ou deux.

SOLUTION. Le modèle est borné (la vente est limitée). Le polygone des solutions réalisables est ouvert parce que les quantités produites peuvent devenir arbitrairement grandes. Remarque (pas demandée) : cela ne signifie pas pour autant qu'il existe une infinité de solutions optimales puisque les quantités produites en surplus doivent alors être mises en stock, et feraient baisser l'objectif réalisé.

Question 15 [1 pt]. Donner la déclaration d'un paramètre *prod_max*, indicé par les mois, et ajouter au modèle précédent une contrainte qui limite la production (ne pas modifier la déclaration de la variable *qte_produite*).

SOLUTION.

```
param prod_max {MOIS} >= 0;
subject to production_limitee {m in MOIS} :
     qte_produite [m] <= prod_max [m];
```

3.2 Contraintes sur le stock

Pour éviter que les produits ne se détériorent s'ils restent trop longtemps en stock, on souhaite que la quantité de produits mis en stock soit nulle au moins une fois tous les trois mois. Pour cela, on commence par introduire une variable booléenne *stock_actif*, destinée à valoir 1 si *qte_stock* est non nulle, 0 sinon.

```
var stock_actif {0 .. N} binary;
```

Question 16 [1 pt]. Écrire une contrainte qui traduise l'implication

$$\textit{qte\_stock}\,[m] = 0 \Longrightarrow \textit{stock\_actif}\,[m] = 0.$$

SOLUTION.

```
subject to zero_implique_zero {m in 0 .. N} :
    qte_stock [m] >= stock_actif [m];
```

Question 17 [1 pt]. Écrire une contrainte qui traduise l'implication

$$\textit{qte\_stock}\,[m] \neq 0 \Longrightarrow \textit{stock\_actif}\,[m] = 1.$$

Remarque : il peut être utile, pour écrire cette contrainte, de disposer d'une borne sur la valeur de *qte_stock*. Cette borne peut se calculer à partir du paramètre *prod_max*.

SOLUTION. On peut prendre la somme des productions maximales comme borne.

```
subject to nzro_implique_nzero {m in 0 .. N} :
    qte_stock [m] <= stock_actif [m] * sum {p in MOIS} prod_max [p];
```

Question 18 [1 pt]. En supposant les deux contraintes précédentes écrites, écrire une contrainte qui impose que, quel que soit $0 \leq m \leq N-2$ on ait

$$\textit{qte\_stock}\,[m] = 0 \quad \text{ou} \quad \textit{qte\_stock}\,[m+1] = 0 \quad \text{ou} \quad \textit{qte\_stock}\,[m+2] = 0.$$

SOLUTION.

```
subject to vider_les_stocks {m in 0 .. N-2} :
   stock_actif [m] + stock_actif [m+1] + stock_actif [m+2] <= 2;
```

4 Le simplexe (5 points)

Question 19 [2 pts]. Résoudre le programme linéaire suivant par l'algorithme du tableau simplicial (il y a moins de quatre itérations et les dénominateurs des fractions sont inférieurs à 5). À chaque itération, indiquer le pivot et la solution de base.

$$\left\{\begin{array}{rcll} 3\,x_1 - x_2 & = & z & [\max] \\ x_1 + 3\,x_2 & \leq & 12 & \\ 2\,x_1 + x_2 & \leq & 9 & \\ x_1 - x_2 & \leq & 3 & \\ x_1,\, x_2 & \geq & 0. & \end{array}\right.$$

SOLUTION. Voici au moins les itérations.

```
linear program equivalent to a tableau
3*x1 - x2 = z[max]
x1 + 3*x2 + x3 = 12
2*x1 + x2 + x4 = 9
x1 - x2 + x5 = 3
x1, x2, x3, x4, x5 >= 0
slack variables = [x3, x4, x5]
------------------------------------------------------------

APPLICATION OF THE DANTZIG ALGORITHM

    x1  x2  x3  x4  x5
   ---------------------------
x3   1   3   1   0   0  |  12
x4   2   1   0   1   0  |   9
x5   1  -1   0   0   1  |   3
   ---------------------------
     3  -1   0   0   0  | 0 = - z0

    x1  x2  x3  x4  x5
   --------------------------
x3   0   4   1   0  -1  |  9
x4   0   3   0   1  -2  |  3
x1   1  -1   0   0   1  |  3
   --------------------------
     0   2   0   0  -3  | -9 = - z0

    x1  x2  x3    x4    x5
   ------------------------------
x3   0   0   1  -4/3   5/3  |  5
x2   0   1   0   1/3  -2/3  |  1
x1   1   0   0   1/3   1/3  |  4
   ------------------------------
     0   0   0  -2/3  -5/3  | -11 = - z0

Basis solution: x1 = 4, x2 = 1, x3 = 5, x4 = 0, x5 = 0
Objective 11

Solution of the initial program
Basis solution: x1 = 4, x2 = 1
Objective 11
```

Question 20 [2 pts]. Écrire le dual du programme linéaire précédent. Déduire la solution optimale et l'objectif réalisé à l'optimum du dual à partir du résultat de la question précédente.

> SOLUTION. La solution optimale du dual (avec les notations du cours) est $y_1 = 0$, $y_2 = 2/3$, $y_3 = 5/3$. On la trouve sur la ligne de l'objectif. Le primal et le dual réalisent le même objectif à l'optimum.

```
DUAL PROGRAM
```

```
linear program under canonical form
12*y1 + 9*y2 + 3*y3 = w[min]
y1 + 2*y2 + y3 >= 3
3*y1 + y2 - y3 >= -1
y1, y2, y3 >= 0
```

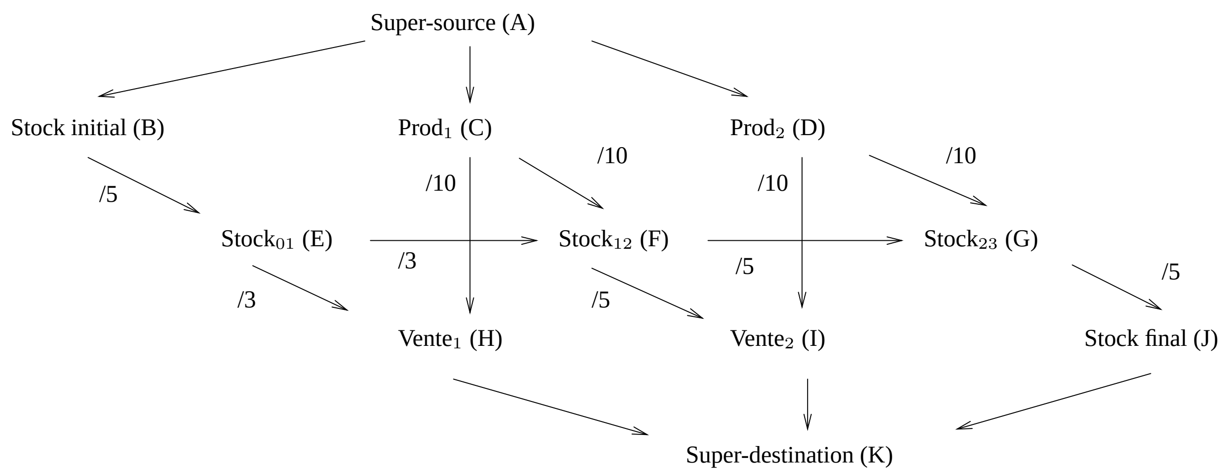
Question 21 [1 pt]. Quelle est la valeur marginale de la deuxième contrainte du primal ? Comment interpréter cette valeur ?

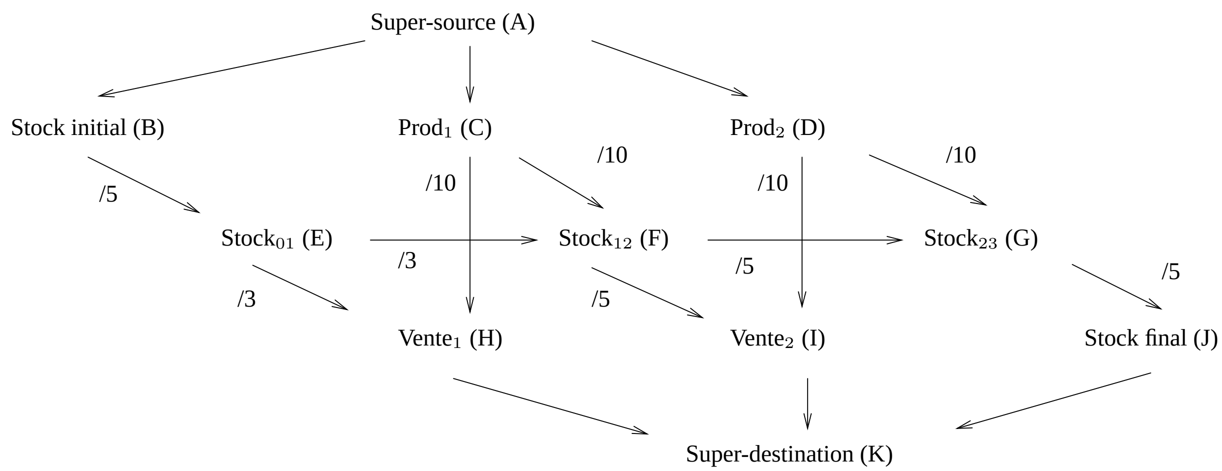
SOLUTION. La valeur marginale de la deuxième contrainte est la valeur de la variable duale y_2 de la contrainte, à l'optimum, c'est–à–dire $2/3$. Interprétation : si on change le second membre de la contrainte de 9 en $9 + \varepsilon$ (avec ε suffisamment petit) alors l'objectif réalisé passe de 11 à $11 + (2/3)\,\varepsilon$.

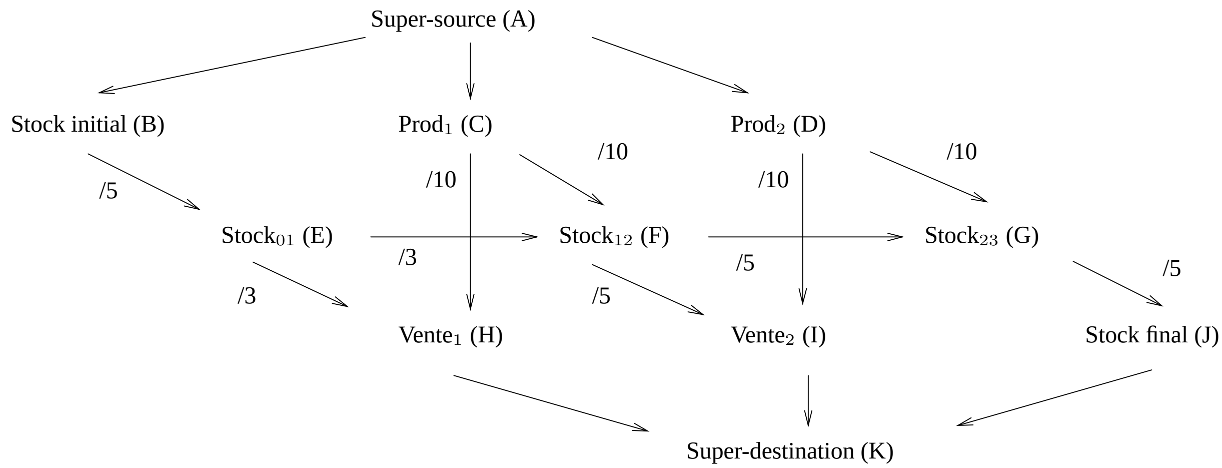
Annexe pour la question 2

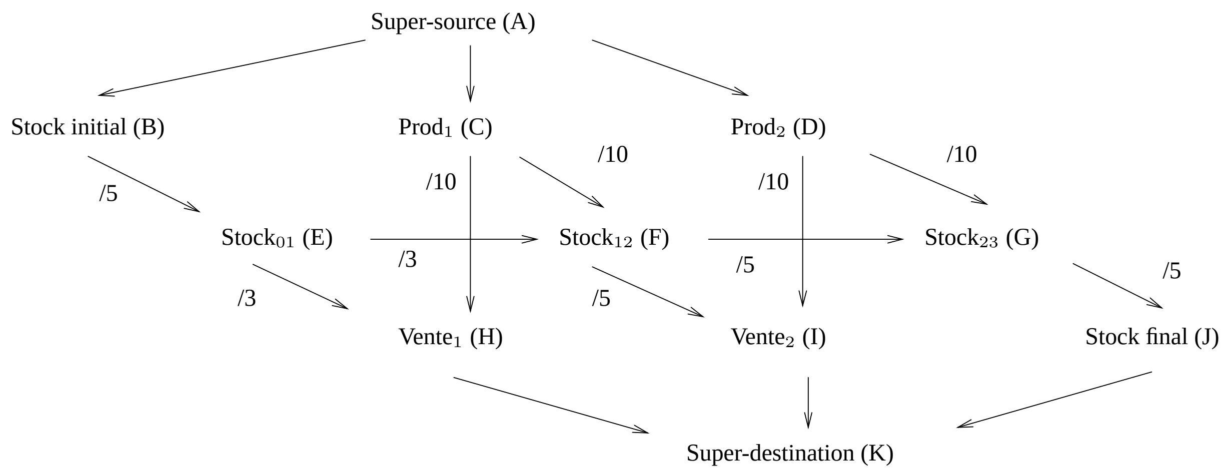
Numéro de place (seul indice de recherche en cas de perte) :

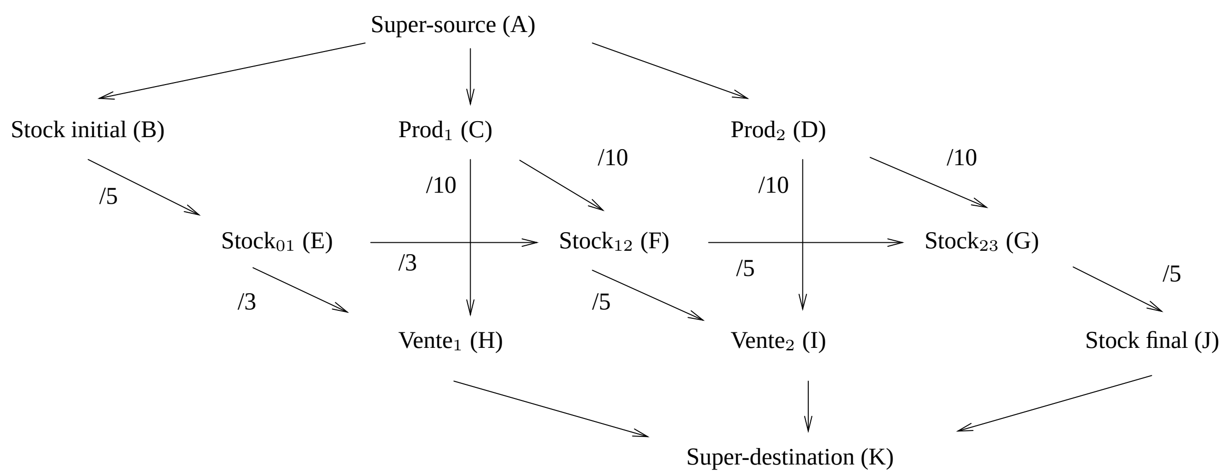
Au début de chaque itération, reporter les flux, indiquer la valeur du flot et surligner la chaîne améliorante. Préciser si la chaîne comporte des arcs indirects.






Super-source (A)
Stock initial (B)
Prod$_1$ (C)
Prod$_2$ (D)
Stock$_{01}$ (E)
Stock$_{12}$ (F)
Stock$_{23}$ (G)
Vente$_1$ (H)
Vente$_2$ (I)
Stock final (J)
Super-destination (K)
/5
/10
/10
/10
/10
/3
/3
/5
/5
/5


Super-source (A)
Stock initial (B)
Prod$_1$ (C)
Prod$_2$ (D)
Stock$_{01}$ (E)
Stock$_{12}$ (F)
Stock$_{23}$ (G)
Vente$_1$ (H)
Vente$_2$ (I)
Stock final (J)
Super-destination (K)
/5
/10
/10
/10
/10
/3
/3
/5
/5
/5


Super-source (A)
Stock initial (B)
Prod$_1$ (C)
Prod$_2$ (D)
Stock$_{01}$ (E)
Stock$_{12}$ (F)
Stock$_{23}$ (G)
Vente$_1$ (H)
Vente$_2$ (I)
Stock final (J)
Super-destination (K)
/5
/10
/10
/10
/10
/3
/3
/5
/5
/5