MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ MOULOUD MAMMERI, TIZI-OUZOU

FACULTÉ DES SCIENCES

DÉPARTEMENT DE MATHÉMATIQUES

Mémoire de Master

Filière : Mathématiques Appliquées

Spécialité : Recherche Opérationnelle

Présenté par :

GHOUTI Thiziri et OUZERDINE Chahrazade

Sujet :

PROBLÈME DE FLOT DANS LE RÈSEAU DE TRANSPORT ET APPLICATION

Devant le jury d'examen composé de :

AOUANE M.	MAA	UMMTO	Président
TALEM DJ.	MCB	UMMTO	Encadreur
KHEFFACHE R.	MCB	UMMTO	ExaminatRICE

Année universitaire 2022-2023

TABLE DES FIGURES

TABLE DES MATIÈRES

Introduction

L'optimisation combinatoire est un domaine des mathématiques appliquées et de l'informatique. Elle s'appuie essentiellement sur la théorie des graphes, l'algorithmique, la programmation linéaire, la programmation dynamique et la théorie de la complexité pour modéliser mathématiquement des problèmes dans de nombreux domaines : le transport, l'économie, le commerce, la biologie, la chimie, l'informatique....

Un problème d'optimisation combinatoire consiste à optimiser la valeur d'un paramètre sur un ensemble discret, dit ensemble des solutions réalisables. En général, cet ensemble est fini, mais peut compter un très grand nombre d'éléments. Pour définir la notion de meilleure solution (ou solution optimale), une fonction, dite **fonction objectif**, est introduite. Ainsi, la solution optimale est celle qui minimise ou maximise la fonction objectif. Dans un réseau routier, déterminer un plus court chemin entre deux villes quelconques v_1 et v_2 est un exemple classique de problème d'optimisation combinatoire. En effet, on associe à ce problème un graphe dont les sommets sont les villes et les arêtes sont les chemins reliant ces villes. L'ensemble des solutions réalisables est l'ensemble des chemins entre v_1 et v_2 tandis que la fonction objectif est l'application qui à chaque chemin allant de v_1 à v_2 associe ça longueur. Bien sûre, un problème d'optimisation combinatoire peut avoir un seul solutions optimales.

Les problèmes d'optimisation combinatoire se formulent d'une manière très simple et leurs solutions se calculent par des algorithmes. Trouver une solution optimale, dans un ensemble discret et fini, est un problème facile en théorie : il suffit d'essayer toutes les solutions, et de comparer leurs qualités pour voir la meilleure. Cependant, en pratique, l'énumération de toutes les solutions peut prendre trop de temps ; or, le temps de recherche de la solution optimale est un facteur très important et c'est à cause de lui que certain problèmes d'optimisation combinatoire sont réputés si difficiles. La théorie de la complexité donne des outils pour mesurer ce temps de recherche. De plus, comme l'ensemble des solutions réalisables est défini de manière implicite, il est aussi parfois très difficile de trouver ne serait-ce qu'une solution réalisable.

La théorie des graphes , fondée par Euler en 1736, est un très vaste domaine, en évolution constante tant du point de vue des recherches fondamentales que de celui des applications. Elle fournit des outils très puissants pour la résolution des problèmes d'optimisation combinatoire. Les problèmes de flot et leurs dérivés, problèmes de transport, d'affectation ... constituent un très important domaine d'application de la théorie des graphes. Sous leur forme la plus simple, ils consistent d'organiser de façon optimale, sous di-

verses contraintes, les mouvements de certaines quantités d'un bien dans un réseau. Ces mouvements concernent, par exemple, l'acheminement d'un produit depuis les centres de production vers les centres de distribution (réseau géographique), la répartition des communications téléphoniques entre les différents centres de gestion (réseau téléphonique), l'organisation de la circulation routière entre plusieurs villes (réseau routier), la distribution des taches au sein d'un ensemble de personnes (problème d'affectation) ...

Pendant les cent années qui suivent, rien ne fut fait dans ce domaine de recherche, En 1847, Kirchhoff (1824-1887) développa la théorie des arbres pour l'appliquer à l'analyse de circuits électriques. Dix ans plus tard, Cayley (1821-1895) découvrit la notion d'arbre alors qu'il essayait d'énumérer les isomères saturés des hydrocarbures de type C_nH_{2n+2}. A cette époque, deux autres problèmes d'importance majeur pour la théorie des graphes furent également proposés et partiellement résolus. Le premier est la conjecture des 4 couleurs qui affirme que quatre couleurs suffisent pour colorier n'importe quelle carte géographique telle que les pays ayant une frontière commune soient de couleurs différentes. Möbius (1790-1868) présenta le premier ce problème dans l'un de ses cours en 1840, Environ dix. ans après, De Morgan (1806-1871) essaya de résoudre ce problème. Ce problème devient célèbre après sa publication par Cayley en 1879. À partir de 1946, La théorie des graphes a connu un développement intense sous l'impulsion de chercheurs motivés. Parmi eux, citons de manière privilégiée Kuhn (1955), Ford et Fulkerson (1956) et Roy (1959). Parallèlement, un important effort de synthèse a été opéré en particulier par Claude Berge dans son ouvrage « Graphes et Hypergraphes ».

La théorie des graphes constitue un domaine des mathématiques qui, historiquement, s'est aussi développé au sein de disciplines diverses telles que la chimie (modélisation de structures), la biologie (génome), les sciences sociales (modélisation des relations) ou en vue d'applications industrielles (problème du voyageur de commerce). Elle constitue l'un des instruments les plus courants et les plus efficaces pour résoudre des problèmes discrets posés en « Recherche opérationnelle » ou « mathématiques discrètes ».

De manière générale, un graphe permet de représenter simplement la structure, les connexions, les cheminements possibles d'un ensemble complexe comprenant un grand nombre de situations, en exprimant les relations, les dépendances entre ses éléments (exp : réseaux de communication, réseaux ferroviaire ou routier, arbre généalogique, diagramme de succession de tâches en gestion de projet).

• Pour le mener à bien, le plan a été élaboré et s'articule autour des points

suivants : introduction générales sur l'optimisation combinatoire et la théorie des graphes.

• Premier chapitre : Au cours de ce chapitre, nous rappelons toute les définitions et notation fondamentale relatives à la théorie des graphes qui seront utile dans le chapitre suivant.

• Deuxième chapitre : problème de cheminement.

• Troisième chapitre : problème de flot.

• Quatrième chapitre : quelques applications.

• En fin, nous achèverons notre travail par une conclusion générale.

CHAPITRE 1

DÉFINITIONS ET NOTATIONS

Introduction

Dans ce premier chapitre,nous présentons quelques définitions et notations fondamentales relatives aux graphes.Notons qu'il ya deux concepts de graphes : graphe orienté et graphe non orienté.

1.1 Graphe orienté et graphe non orienté

Définition 1.1. Un graphe orienté ou digraphe $G = (X; U)$ est déterminé par la donnée : D'un ensemble $X = \{x_1; x_2; \ldots; x_n\}$ dont les éléments sont appelés **sommets**, et d'un ensemble $U = \{u_1; u_2; \ldots; u_m\}$ dont les éléments sont des couples ordonnés de sommets appelés **arcs**. Graphiquement, les sommets sont représentés par des points et et tout arc $u = (i; j)$ sera représenté par une courbe orienté de du point i vers le point j.

Définition 1.2. Soit $u = (i, j)$ un arc de G

1 le sommet i est l'**extrémité initiale** de l'arc u ;noté I.
2 le sommet j est l'**extrémité terminale** de u ;noté T.
3 si les extrémités de u coïncident u est appelé une **boucle**.

Exemple 1.1. La figure 1.1 donne un exemple de digraphe dont l'ensemble des sommets est

$$V = \{a, b, c, d, e, g\}$$

et celui des arcs est

$$U = \{(a, b), (a, c), (d, c), (a, d), (c, e), (g, c)(e, d)\}$$

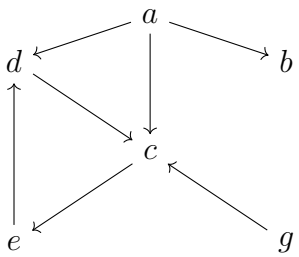


FIGURE 1.1 – Digraphe

Définition 1.3.

1. On dit que le sommet j est un successeur de i ; s'il existe un arc $u = (i, j)$. L'ensemble des successeurs du sommet $i \in X$ est
$$\Gamma^+(i) = \{j \in X, (i, j) \in U\}$$
2. On dit que le sommet j est un prédécesseur i ; s'il existe un arc $u = (j, i)$. L'ensemble des prédécesseurs du sommet i est
$$\Gamma^-(i) = \{(j, i) \in X, (j, i) \in U\}$$
3. On dit que le sommet j est un voisin de i si
$$j \in \Gamma(i) = \Gamma^+(i) \cup \Gamma^-(i)$$

Dans la figure 1.1, les voisins du sommet c sont $\Gamma(c) = \{a, d, e, g\}$.

Dans certains situations, la notion d'orientation n'est pas importante, on utilise alors la notion de graphe non orienté ou graphe tout court.

Définition 1.4. Un **graphe** non vide $G = (V, E)$ est un couple constitué par un ensemble fini
$$V = \{v_1, v_2, \ldots ...v_n\}$$
dont les éléments sont des **sommets** et d'un ensemble
$$E = \{e_1, e_2, \ldots, e_m\}$$
dont les éléments sont des **arêtes**. Une arête e est définie par une paire de sommets $\{x, y\}$ non ordonnés, appelés les extrémités de e.

Définition 1.5. Un graphe est dit simple :
— s'il est sans boucle
— entre deux sommets, il y a au plus une arrête (resp. un arc).
Les graphe définis précédemment sont simple.

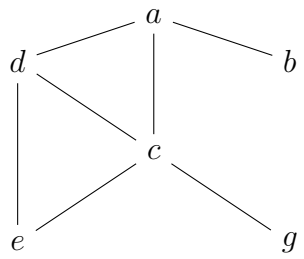


FIGURE 1.2 – Graphe non orienté

Définition 1.6. Soit $G = (X; U)$ un graphe et soit $u = (i; j) \in U$:
— les sommets i et j sont dits **adjacents**.
— l'arc u **incident** aux sommets i et j.

Définition 1.7. Á tout sommet i du graphe $G = (X; U)$, on associe :
— son **demi-degré extérieur** qui est

$$d^+(i) = \Gamma^+(i)$$

.
— son **demi-degré intérieur**

$$d^-(i) = \Gamma^-(i)$$

— son **degré** qui est

$$d(i) = d^+(i) + d^-(i)$$

Remarque 1.1. Un sommet de degré zéro est dit isolé.

1.2 Sous-graphe

Définition 1.8. Soit $G = (V(G), E(G))$ un graphe. On appelle **sous-graphe** de G tout graphe $H = (V(H), E(H))$ tels que $V(H) \subseteq V(G)$ et $E(H) \subseteq E(G)$. Un sous-graphe H est dit **graphe partiel** de G, si $V(H) = V(G)$ et $E(H) \subseteq E(G)$. Autrement dit, le graphe partiel est obtenu par la suppression des arêtes sans toucher aux sommets.
Un sous-graphe H est un sous-graphe **induit** par l'ensemble de sommets A, si $V(H) = A$ et $E(H)$ est l'ensemble de toutes les arêtes de G ayant leurs extrémités dans A. Autrement dit, le sous graphe induit par l'ensemble A est obtenu par la suppression des sommets de $V - A$.

1.3 Graphe connexe et fortement connexe

Définition 1.9. Soit G un graphe ou un digraphe simple.

1. Une **chaine** allant d'un sommet x vers un sommet y est une séquence finie de n sommets
$$\mu = (x_0, x_1; \ldots, x_k)$$
tels que $x_0 = x$, $x_k = y$ et pour tout i dans $\{0, 1, \ldots, k-1\}$, x_i et x_{i+1} sont adjacents.
2. Un **chemin** allant de x_1 à x_k est une séquence de sommets $(x_1, x_2, ...x_k)$ telle que (x_i, x_{i+1}) est un arc.
3. Une chaine (resp. chemin) est dite **élémentaire** si elle ne rencontre pas le même sommets deux fois ; une chaine élémentaire passant par tous les sommets de G est dite **hamiltonienne**.
4. Un **cycle** (resp. **circuit**) est une chaine (resp. chemin) ayant l'extrémité initiale confondue avec l'extrémité terminale.
Définition 1.10. On dit qu'un graphe G est **connexe**, s'il existe une chaine entre toute paire de sommets. Si G n'est pas connexe, il contient au moins, deux sous-graphes connexes, maximaux au sens de l'inclusion, appelés **Composantes connexes**.

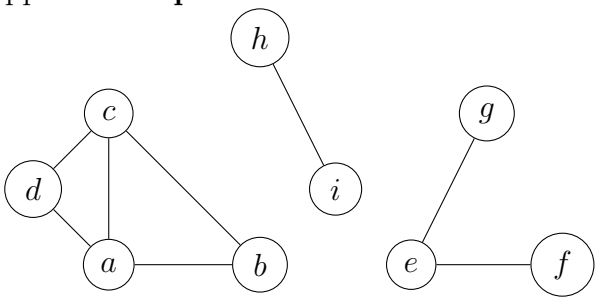


FIGURE 1.3 – Graphe non connexe avec trois composantes connexes : $C_1 = \{a, b, c, d\}, C_2 = \{e, f, g\}$ et $C_3 = \{i, h\}$.

Définition 1.11. Un graphe orienté $G = (V, U)$ est dit **fortement connexe**, si quels que soient les sommets u et v de V, il existe un chemin orienté allant de u à v et un autre chemin allant de v à u. C'est-à-dire il existe un circuit passant par les deux sommets. Un graphe qui n'est pas fortement connexe est constitué par des composantes fortement connexes.

1.4 Représentation des graphes en machine

Les graphes peuvent être considérés comme une structure de données. C'est pourquoi, il est fondamental de s'intéresser à comment les présenter en machine. Plusieurs modes de représentation peuvent être envisagées selon la nature des traitements que l'on souhaite appliquer au graphe considéré.

1. **Représentation par matrice d'adjacences :** La matrice d'adjacence d'un graphe simple $G = (V, U)$ d'ordre n est la matrice $M = (b_{ij})_{1 \leq i,j \leq n}$ de dimension $(n \times n)$ (une ligne pour chaque sommet et une colonne pour chaque sommet) telle que :

$$b_{i,j} = \begin{cases} 1 & si \quad ij \in U \\ 0 & sinon \end{cases}$$

Exemple 1.2. Soient le graphe non orienté G et le graphe H obtenu par l'orientation de G. Ces deux graphes ont la même matrice d'adjacence. La matrice d'adjacence des deux graphes est la

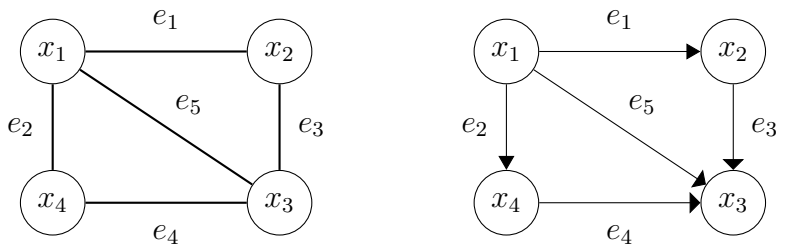


FIGURE 1.4

suivante :

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

2. **Représentation par la matrice d'incidence :** Soit $G = (V, U)$ un graphe simple, et la matrice $B = (n \times m)$, une matrice ayant n ligne et m colonne, c'est à dire une ligne pour chaque sommet et une colonne pour chaque arête telle que :

(a) pour un graphe orienté, $b_{ij} = 1$ si le sommet i est l'extrémité initiale de l'arc j, $b_{ij} = -1$ si i et l'extrémité terminale de l'arc j et $a_{ij} = 0$ sinon.

(b) pour un graphe non orienté, $b_{ij} = 1$ si le sommet i est l'extrémité de l'arête j, $b_{ij} = 0$ sinon.

Pour un graphe orienté, la matrice d'incidence est définie :

$$b_{ij} = \begin{cases} 1 & si \quad x_i \text{ est l'extrémité initiale de } \mathrm{u}_j; \\ -1 & si \quad x_i \text{ est l'extrémité terminale de } \mathrm{u}_j; \\ 0 & si \quad x_i \text{ n'est pas une extrémité de } \mathrm{u}_j. \end{cases}$$

Pour un graphe non orienté, la matrice d'incidence est définie par :

$$b_{ij} = \begin{cases} 1 & si \quad x_i \text{ est une extrémité de } \mathrm{u}_j; \\ 0 & sinon. \end{cases}$$

Exemple 1.3. La matrice d'incidence associée au graphe orienté H de la figure 1.4 est la suivante :

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{pmatrix} +1 & +1 & 0 & 0 & +1 \\ -1 & 0 & +1 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & -1 & 0 & +1 & 0 \end{pmatrix} \end{array}$$

La matrice d'incidence associée au graphe non orienté G de la figure 1.4 est la suivante :

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

CHAPITRE 2

PROBLÈMES DE CHEMINEMENT

Introduction
Dans ce chapitre, nous allons s'intéresser à un problème concret "comment trouver un plus court chemin entre deux points quelconques ? Ce problème connu sous le nom de problème de cheminement peut être modéliser par un graphe valué $G = (V, U, l)$. Pour résoudre ce problème, nous allons étudier trois algorithmes efficaces : l'algorithme de FORD qui s'applique pour un graphe quelconque, l'algorithme de BELMAN dans un graphe sans circuit, et l'algorithme de DIJKSTRA lorsque les arcs sont valués positivement.

2.1 Définitions

Dans toute la suite de ce chapitre, $G = (V, U, l)$ désigne un graphe valué où l est une application sur l'ensemble des arcs (ou arêtes) U tel que pour tout arc $(x, y) \in U$, $l(x, y)$ peut être la longueur, le poids, le coût, la capacité... de l'arc (x, y). Rappelons qu'un chemin est une séquence de sommets $C = (v_1, v_2...v_k)$ tels que pour $i = 1...k-1, (v_i, v_{i+1})$ est un arc. Si (v_k, v_1) est un arc, C est un circuit. Dans un graphe valué, la longueur d'un chemin C est la somme des longueurs de ses arcs, c'est à dire $l(C) = \sum_{(x,y)\in C} l(x, y)$.
Remarque 2.1. Dans un graphe non orienté, une arête xy est vu comme deux arcs, elle est vu comme l'arc (x, y) et aussi comme l'arc (y, x). Ainsi, dans un graphe non orienté une chaine (resp. un cycle) est aussi un chemin (resp. un circuit).

Définition 2.1. On appelle **distance** entre deux sommets u, v dans le graphe valué G, la quantité $d(u, v)$ qui est égale à la longueur du plus court chemin allant de u à v si celui ci existe ou $+\infty$ si un tel chemin n'existe pas.

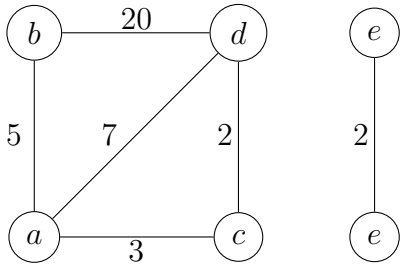


FIGURE 2.1 – La distance entre deux sommets

Dans le graphe 2.1, la distance entre les sommets b et d est $d(b, d) = 10$. En effet, entre ces deux point il y a trois chemins : $\mu_1 = (b, d)$, $\mu_2 = (b, a, d)$ et $\mu_3 = (b, a, c, d)$ de longueurs respectivement 20, 5+7 et 5+3+2, donc $d(b, d) = \min(20, 12, 10) = 10$. Les sommets c et e appartiennent à deux composantes connexes, donc il n'existe aucun chemin entre ces deux point, par conséquent $d(c, e) = \infty$

Lemme 2.1. *Tout sous-chemin d'un chemin de valeur minimale est un chemin de valeur minimale.*

Démonstration. Soit $C = (x_1, x_2....x_k)$ le plus court chemin allant de x_1 à x_k, c'est à dire $d(x_1, x_k) = k-1$. Soit t tel que $1 < t < k$ et soient $C_1 = (x_1, x_2...x_t)$, $C_2 = (x_t, x_{t+1}...x_k)$ les sous chemins de C, c'est à dire $C = C_1 \cup C_2$ et $l(C) = l(C_1) + l(C_2)$.
Supposons maintenant que C_1, par exemple, n'est pas un plus court chemin allant de x_1 à x_t, donc il existe un autre chemin C' allant de x_1 à x_t tel que $l(C') < l(C_1)$. Ceci implique que $C" = C' \cup C_2$ est un chemin allant de x_1 à x_2 de longueur $l(C") = l(C') + l(C_2) < l(C_1) + l(C_2) = l(C)$. Mais, ceci contredit le fait que C est de valeur minimale. □

Définition 2.2. On appelle **absorbant** tout circuit C de longueur strictement négative. Dans le graphe valué de la figure 3.1, $C = (b, c, d, b)$ est un circuit absorbant, car $l(C) = l(bc) + l(cd) + l(db) = 1 + 1 - 4 = -2 < 0$.

Lemme 2.2. *Soit $\mu = (x_1, x_2...x_k)$ un chemin allant de x_1 à x_k. S'il existe un circuit absorbant C tel que $\mu \cap C \neq \emptyset$, alors C ne peut pas être un plus court chemin entre x_1 et x_k.*

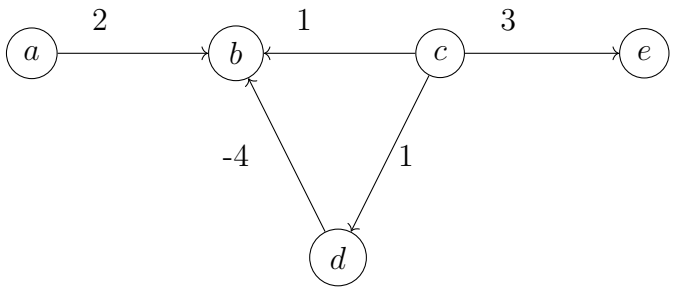


FIGURE 2.2 – Cercle absorbant

Démonstration. □

Définition 2.3. Soient x et y deux sommets dans un graphe orienté G.

1. On dit que y est un successeur de x s'il existe un chemin allant de x à y. Dans ce cas, x est un prédécesseur de y non immédiate.
2. On dit que x est une **source** s'il n'a pas de prédécesseur, il est un **puits** s'il n'a pas de successeur.

Remarque 2.2. Notons que

1. Si x est une source, alors $d^-(x) = 0$,
2. si x est un puits, alors $d^+(x) = 0$;
3. Un graphe orienté peut avoir plusieurs sources et puits.

Définition 2.4. Dans un graphe G orienté, un sommet x est une **racine** (resp. **anti-racine**)si pour tout $y \in V$, il existe un chemin dans G joignant x à y (resp. y à x).

Lemme 2.3. *Dans un graphe orienté sans circuits, il existe un sommet source et un sommet puits.*

Démonstration. Considérer un sommet quelconque x_l, puis, s'il existe, un prédécesseur x_2 de x_1, de même un prédécesseur x_3 de x_2 et ainsi de suite tant qu'il existe un prédécesseur du dernier sommet considéré. Cette construction d'une suite de sommets s'arrête nécessairement au bout d'un nombre fini de sommets car le graphe étant fini et n'ayant, par hypothèse, pas de circuits, on ne peut pas retrouver un sommet déjà rencontré précédemment. Et lorsque cette construction s'arrête on se trouve en un sommet qui par construction n'a pas de prédécesseur, c'est-à-dire en un sommet source du graphe dont l'existence est ainsi montrée. On peut procéder de la même façon pour un sommet puits. □

Remarque 2.3. On peut vérifier facilement qu'un graphe orienté avec ou sans circuit peut ne pas avoir ni racine ni anti racine.
Définition 2.5. On appelle tri ou ordre topologique d'un graphe orienté G, une numérotation ou liste de sommets $L = x_1, x_2, \ldots, x_n$ tel que pour $i = 1...n$, x_i est une racine dans G_i, le sous graphe induit par les sommets $x_i, x_{i+1}...x_n$. Autrement dit, (x_i, x_j) est un arc implique $i < j$. Dans le graphe de la figure 2.3, $L_1 = abdfec$ n'est pas

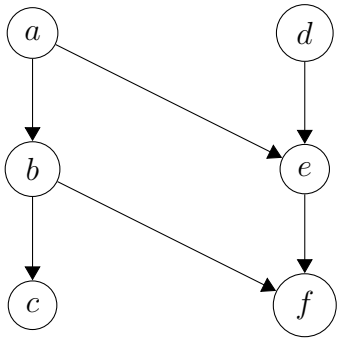


FIGURE 2.3 – Tri topologique d'un graphe sans circuit

un tri topologique, car le sommet f n'est pas une racine dans le sous graphe induit par les sommets f, e et c. Cependant, on peut vérifier que $L_2 = abdefc$ et $L_3 = abcdef$ sont des tris topologiques.
Proposition 2.4. *Un graphe orienté G est sans circuits si et seulement s'il admet un tri topologique de ses sommets.*

Le tri topologique est utilisé pour planifier une exécution de taches. En effet, on associe un sommet à chaque tache. Si une tache t_i doit être exécutée avant la tache t_j, ces deux taches sont reliées par un arc (t_i, t_j). Ainsi, pour exécuter ces taches sans violer la contrainte de prudence, on doit calculer un tri topologique du graphe associé.

2.2 Algorithme de Dijkstra

L'algorithme de Dijkstra est utilisé dans un graphe valué orienté ou non orienté et dont les poids sur les arcs ou arêtes sont positifs. Cet algorithme est utilisé pour calculer la distance d'un sommet donné r par rapport à tous les autre sommets du graphe.
Le principe de l'algorithme de Dijkstra peut être résumé comme suit : Initialement, le sommet r est le seule marqué en vert, $d(r) = 0$ et $d(v) = +\infty$ pour tout sommet $v \neq r$.

Il utilise deux couleurs : le vert pour les sommets dont la distance calculée n'est pas définitive et une couleur rouge pour les sommets dont la distance calculée est définitive.
A chaque itération, on choisit un sommet vert v dont la distance actuelle est minimale sur l'ensemble des sommets verts ; pour tout voisin w de v non rouge, on met à jour sa distance comme indiqué dans la ligne 7 de l'algorithme et on le marque en vers s'il n'est pas vert ; on marque en suit v en rouge.

Algorithme : Dijkstra
Données : *un graphe $G = (V, E)$, une valuation l strictement positive sur les arêtes de G et un sommet $r \in V$.*
Résultat : *les plus courts chemins allant de r vers tout sommet v de G.*

1. *Début*
2. *$d(r) = 0$, $d(v) = +\infty$ pour tout sommet $v \neq r$*
3. *Marquer r en vert*
4. *Tant que il reste des sommets verts faire*
5. *Choisir un sommet vert v qui minimise la fonction d*
6. *Pour tout voisin w de v non marqués en rouge faire*
7. *$d(w) = \min\big(d(w), d(v) + l(vw)\big)$*
8. *Si w n'est pas vert, marquer w en vert*
9. *Fin Si*
10. *Fin Pour*
11. *Marquer v en rouge*
12. *Fin Tant que*
13. *Renvoyer la fonction d*
14. *Fin*

Exemple 2.1. Appliquons l'algorithme de Dijkstra pour le graphe de la figure 2.4
Etape 1 : Un seul sommet vert, le sommet r, $d(r) = 0$. Deux voisins non marqués de r, les sommets a et c et dont les nouvelles distances sont respectivement 10 et 3. Les sommets a, c deviennent verts et le sommet r devient rouge.

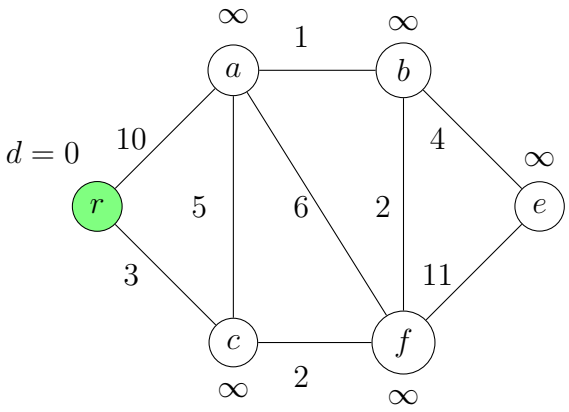
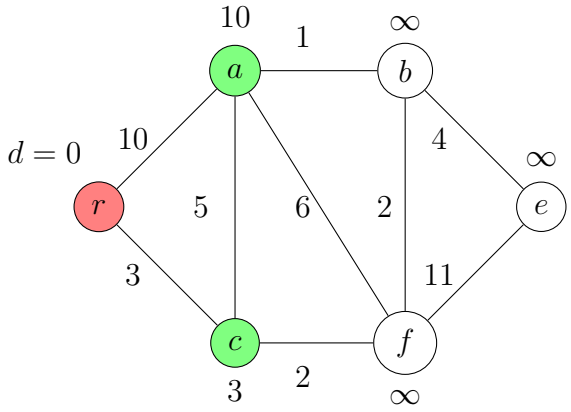
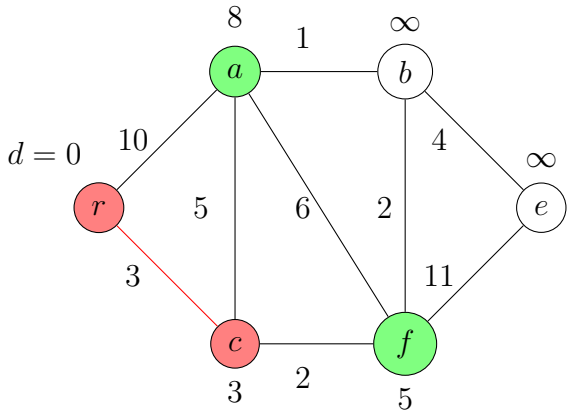


FIGURE 2.4

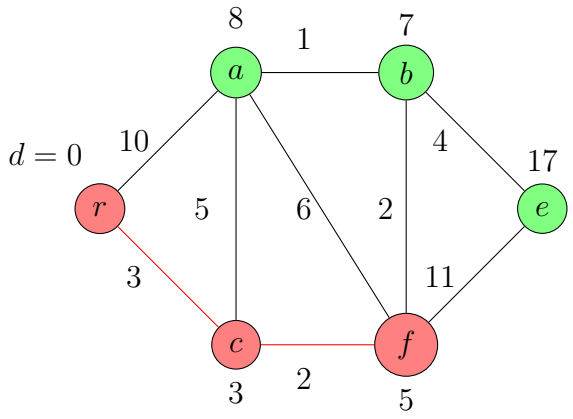


Etape 2 : Le minimum des distances sur les sommets verts est $d(c) = 3$. Les voisins de c non rouges sont a et f. Les nouvelles distances de a et f sont respectivement 3+5=8 et 3+2=5. Le sommet c devient rouge et le sommet f devient vert.

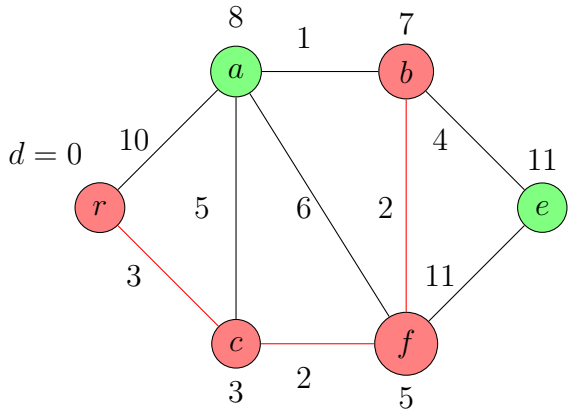


Etape 3 : Le minimum des distances sur les sommets verts est $d(f) = 5$. Les voisins de f non

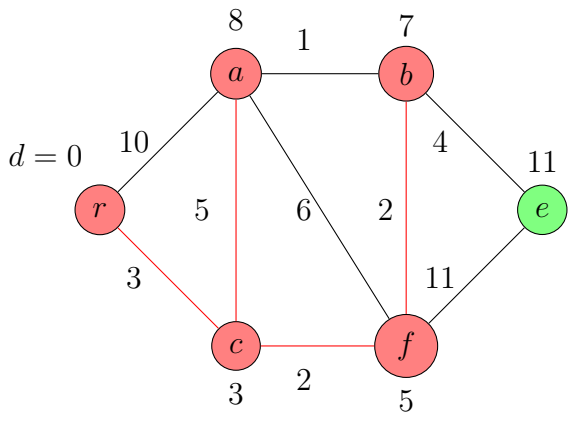
rouges sont a b et e. Les nouvelles distances de a , b et e sont respectivement 8, 5+2=7 et 5+11=16. Le sommet f devient rouge et les sommets b , e deviennent verts.



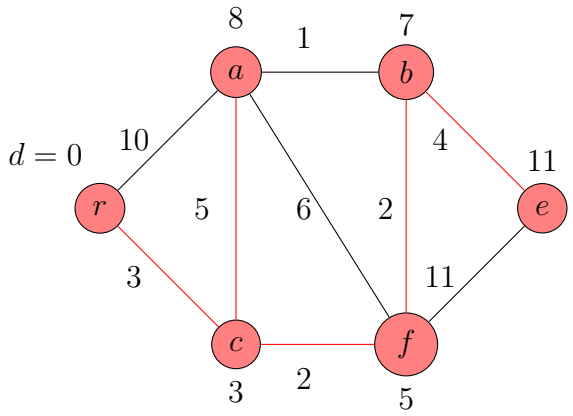
Etape 4 : Le minimum des distances sur les sommets verts est $d(b) = 7$. Les voisins de b non rouges sont a et e. Les nouvelles distances de a et e sont respectivement 8, 7+4=11. Le sommet b devient rouge.



Étape 5 : Le minimum des distances sur les sommets verts est $d(a) = 8$ (cette distance est la même si on passe par le sommet c ou par le sommet b). Le sommet a n'a pas de voisin vert. Le sommet a devient rouge.



Étape 6 : Le minimum des distances sur les sommets verts est $d(e) = 11$. Tous les voisins de e sont rouge. Le sommet e devient rouge.



2.3 Algorithme de Bellman

L'algorithme de Bellman s'applique pour des graphes valués orientés et sans circuit. On donne en entrée un graphe orienté valué et sans circuit et un sommet r. On aura en sortie une arborescence des plus courts chemins issus de r si et seulement si r est une racine de G. Son principe est le suivant :
On construit progressivement l'arborescence des plus courts chemins à partir de r, on initialise $S = \{r\}$ et $A = \emptyset$. A chaque itération on choisit un sommet v n'appartenant pas à S et dont les prédécesseurs sont dans S. Après avoir calculer la distance de v à r (voir la ligne 5 de l'algorithme), le sommet v sera rajouté à S et l'arc (u, v) sera rajouté à A (l'arc (u, v) est le dernier arc du plus cours chemin allant de r à v).
Algoritme de Bellman :

Entrée : Un graphe $G = (V, U, l)$ sans circuit et un sommet r, $|U| = n$.
Sortie : Une arborescence $T = (V, A)$ des plus courts chemins d'origine r si et seulement si r est une racine.

1. $S = \{r\}$, $\pi(r) = d(r, r) = 0$, $\pi(v) = d(r, v) = \infty$, pour $v \neq r$, $A = \emptyset$
2. Tant que $|S| \neq n$ faire
3. Choisir $v \in U - S$ ayant tous ses prédécesseurs dans S.
4. Si un tel sommet n'existe pas, terminer r n'est pas une racine.
5. Sinon poser

$$\pi(v) = \min\{\pi(s) + l(s, v), (s, v) \in U\} = \pi(u) + l(u, v)$$

6. Fin si
7. $S = S \cup u$, $A = A \cup (u, v)$
8. Fin tant que
9. Retourner A
10. Fin.

Exemple 2.2. Appliquons l'algorithme de Bellman pour le graphe de la figure 2.5

Etape 1 : $S = \{r\}$, $A = \emptyset$. Le seul sommet ayant tous ses prédécesseurs dan S est le sommet c. $\pi(c) = 0+3 = 3$, $S = \{r, c\}$, $A = \{(r, c)\}$

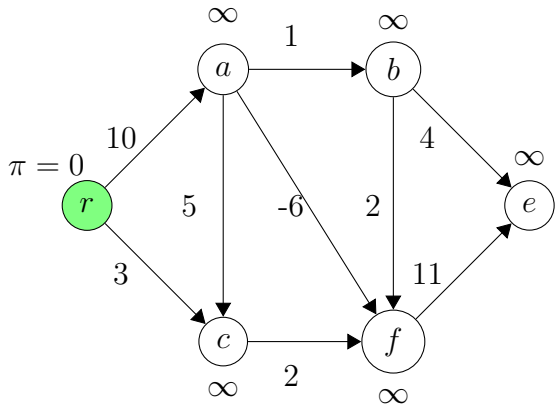
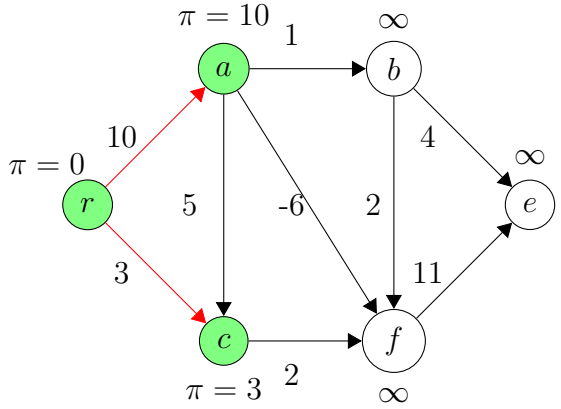
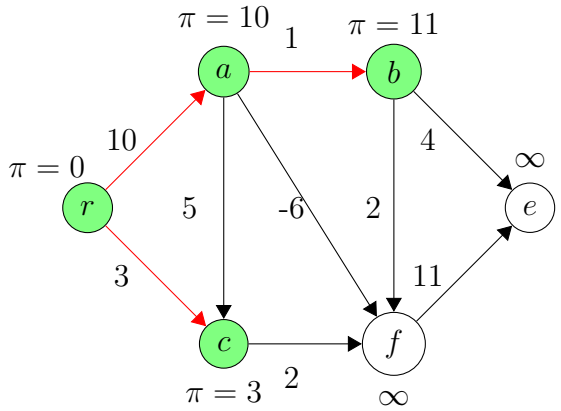


FIGURE 2.5

Etape 2 : Les prédécesseurs du sommet a sont dans $S = \{r, c\}$. $\pi(a) = 0 + 10 = 10$, $S = \{r, c, a\}$, $A = \{(r, c), (r, a)\}$



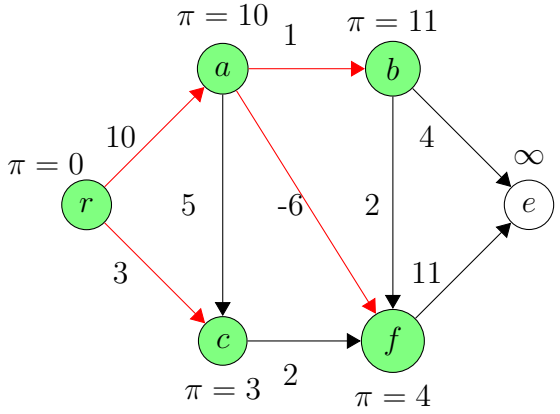
Etape 3 : Les prédécesseurs du sommet b sont dans $S = \{r, c, a\}$. $\pi(b) = \pi(a) + 1 = 11$,
$S = \{r, c, a, b\}, A = \{(r, c), (r, a), (a, b)\}$.



Etape 4 : Les prédécesseurs du sommet f sont dans $S = \{r, c, a, b\}$.

$$\pi(f) = \min\{3 + 2, 10 - 6, 11 + 2\} = \pi(a) - 6 = 4$$

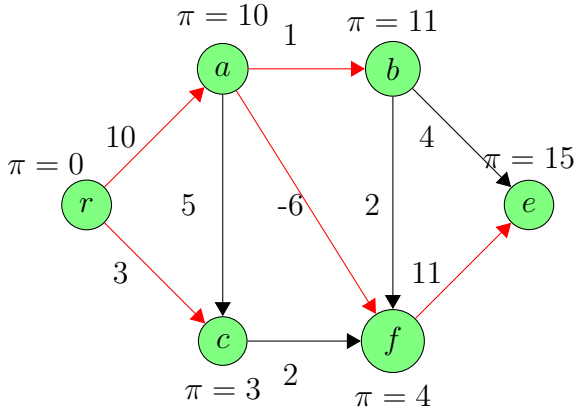
, $S = \{r, c, a, b, f\}$, $A = \{(r, c), (r, a), (a, b), (a, f)\}$.



Etape 5 : Les prédécesseurs du sommet e sont dans $S = \{r, c, a, b, f\}$.

$$\pi(e) = \min\{11 + 4, 4 + 11\} = \pi(f) + 4 = 15$$

, $S = \{r, c, a, b, f, e\} = V$, $A = \{(r, c), (r, a), (a, b), (a, f), (f, e)\}$.



L'algorithme s'arrête, car $S = V$ et l'arborescence des plus courts chemins est définie par les arcs rouges. Notons que à la dernière étape, on pouvait aussi prendre l'arc (b, e), ceci montre que cette arborescence n'est pas unique .

CHAPITRE 3

PROBLÈME DE FLOT

Introduction
La notion de flot dans les graphes est un outil très puissant en optimisation combinatoire. Elle présente un aspect algébrique très intéressant et se prête à de grandes applications. En effet, un flot dans un réseau peut-être vu comme l'écoulement d'une substance le long des arêtes d'un graphe. Ainsi la circulation d'eau dans des conduites, d'électricité dans les câbles, d'appels téléphoniques, de courriers électroniques, d'informations dans l'internet ou de véhicules peut être modélisée par un flot dans un réseau.

3.1 Définitions et notations

Soient $G = (V, U)$ un graphe orienté sans boucle. Pour tout ensemble non vide de sommets S, on pose

1. $w^+(S) = \{a = (x, y) \in U \ ; \ x = I(a) \in S, y = T(a) \notin S\}$;
2. $w^-(S) = \{a \in U \ ; \ I(a) \notin S, T(a) \in S\}$;
3. $w(S) = w^+(S) \cup w^-(S)$

tel que : I(a) est l'extrémité initiale de a et T(a) l'extrémité terminale de a.
Ainsi, $w^+(S)$ désigne l'ensemble des arcs sortant de S et $w^-(S)$ désigne l'ensemble des arcs entrant dans S
Définition 3.1. Un ensemble A d'arc est appelé **cocycle**, s'il existe un ensemble de sommets S tel que

$$A = w(S)$$

On dit alors que A est un cocycle défini par S. Ainsi, le cocycle $w(S)$ est l'ensemble des arcs ayant une extrémité dans S et l'autre extrémité en dehors de S.

Par abus d'écriture, $w^+(v)$ désigne l'ensemble des arcs sortant de v ; $w^-(v)$ désigne l'ensemble des arcs entrant en v, donc $w(v)$ désigne le cocycle définie par le sommet v.
Une **source** (ou une **entrée**) est un sommet s vérifiant $w^-(s) = \emptyset$; une **sortie** (ou un **puits**) est un sommet t vérifiant $w^+(t) = \emptyset$.

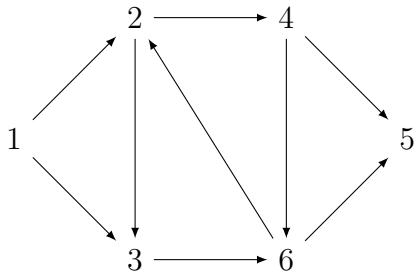


FIGURE 3.1

Dans le graphe de la figure 3.1, $s = 1$ est une source et $t = 5$ est un puits.
Pour $S = \{3, 6\}$,
$w(S) = \{(1, 3), (2, 3), (6, 2), (4, 6), (6, 5)\}$;
$w^+(S) = \{(6, 2), (6, 5)\}$;
$w^-(S) = \{(1, 3), (2, 3), (4, 6)\}$.
Définition 3.2. (Réseau de transport)
Un **réseau de transport** est un graphe value $R = (G, c)$ où $G = (V, U)$ est un graphe orienté contenant une source s et une sortie t tel qu'il existe au moins un chemin de s vers t, et c est une capacité sur les arcs. Une **capacité** c dans un réseau R est une application sur l'ensemble U associant à tout arc $(u, v) \in U$ un nombre positif $c(u, v)$. La capacité $c(u, v)$ d'un arc (u, v) peut être vue comme la quantité maximale d'eau, de véhicules, d'informations... qu'on peut faire circuler sur celui-ci. Les sommets appartenant à $V - \{s, t\}$ sont appelés **sommets intermédiaires**.

Le graphe représenté par la figure 3.2 est un réseau de transport dont les nombres entre deux crochets sont des capacités.
Définition 3.3. (Coupe)
On appelle une **coupe** dans un réseau de transport toute partition

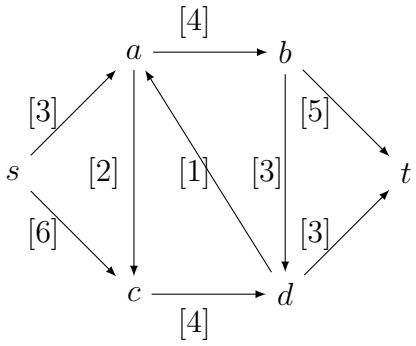


FIGURE 3.2 – Réseau de transport

$\mathcal{C} = \{X, Y\}$ de l'ensemble des sommets V en deux parties telle que $s \in X$ et $t \in Y$.

Définition 3.4. (Capacité d'une coupe)
La **capacité** de la coupe $\mathcal{C} = \{X, Y\}$ est donnée par la somme des capacités des arcs sortant de X (et qui est égale à la somme des capacités sur les arcs entrant dans Y) :

$$\begin{aligned} c(\mathcal{C}) = c(X, Y) &= \sum_{x \in X, y \in Y} c(x, y) \\ &= \sum_{(x,y) \in w^+(X)} c(x, y) \\ &= \sum_{(x,y) \in w^+(X)} c(x, y) \end{aligned}$$

Dans le réseau de la figure 3.2, $\{X = \{s\}, Y = \{a, b, c, d, t\}\}$, $\{X_1 = \{s, c, d\}, Y_1 = \{a, b, t\}\}$ sont deux coupes ; leurs capacités respectives sont :

$$\begin{aligned} c(X, Y) &= c(s, a) + c(s, c) \\ &= 3 + 6 = 9 \end{aligned}$$

$$\begin{aligned} c(X_1, Y_1) &= c(s, a) + c(b, a) + c(d, t) \\ &= 3 + 1 + 3 = 7 \end{aligned}$$

Définition 3.5. (Coupe minimum)
Une coupe $\mathcal{C} = \{X, Y\}$ est **minimum** si sa capacité est minimum : pour toute coupe $\mathcal{C}'$,

$$c(\mathcal{C}) \leq c(\mathcal{C}')$$

Définition 3.6. (flot)
Un **flot** dans un réseau de transport est une application $f : U \to \mathbb{R}_+$ vérifiant

1. La contrainte de capacité : pour tout arc a,
$$0 \leq f(a) \leq c(a)$$
2. La loi de conservation des flux : pour tout sommet $v \in V - \{s, t\}$,
$$\sum_{a \in w^+(v)} f(a) = \sum_{a \in w^-(v)} f(a)$$

Remarque 3.1. Pour distinguer entre un flot et une capacité, les capacités sont représentés entre deux crochés.

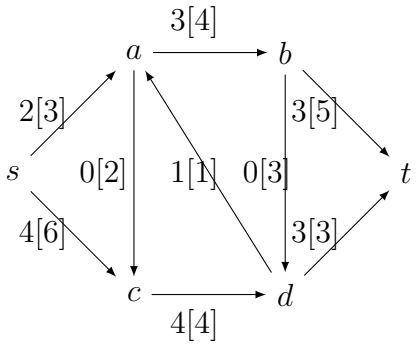


FIGURE 3.3 – Réseau avec des capacités entre crochets et un flot.

Dans la figure 3.3, les nombres se trouvant à gauche des capacités définissent un flot.

Définition 3.7. (flux entrant et flux sortant)
On appelle **flot sortant** d'un ensemble de sommets S la quantité
$$f^+(S) = \sum_{a \in w^+(S)} f(a)$$

Le **flot entrant** dans S est
$$f^-(S) = \sum_{a \in w^-(S)} f(a)$$

En particulier, le flot sortant et le flot entrant d'un sommet v sont respectivement
$$f^+(v) = \sum_{a \in w^+(v)} f(a)\,;\; f^-(v) = \sum_{a \in w^-(v)} f(a)$$

Définition 3.8. Pour deux ensembles de sommets S, T, $f(S,T)$ désigne la quantité de flot qui sort de S vers T, c'est à dire

$$f(S,T) = \sum_{(x,y)\in S\times T} f(x,y)$$

Dans la figure 3.3, pour $S = \{s,a\}$, $T = \{c,d\}$
$f(S,T) = f(s,c) = 4$, $f(T,S) = f(d,a) = 1$. Ceci implique directement que

$$f(S,T) \neq f(T,S)$$

La proposition suivante montre que la loi de conservation de flot est vérifiée en tout ensemble de sommets S ne contenant pas la source et le puits.

Proposition 3.1. *Pour tout ensemble de sommets $S \subseteq V - \{s,t\}$,*

$$f^+(S) = f^-(S)$$

Démonstration. Ceci est vrai si S est réduit à un seul sommet (par définition). Supposons que ceci reste vrai pour tout ensemble S et montrons qu'il le reste aussi pour l'ensemble $S \cup x$, avec $x \neq s$ et $x \neq t$.
On a

$$f^+(S\cup x) = f^+(S) - f(S,x) + f^+(x) - f(x,S)$$
$$f^-(S\cup x) = f^-(S) - f(x,S) + f^-(x) - f(S,x)$$

Ainsi, après simplification, on aura

$$f^+(S\cup x) - f^-(S\cup x) = f^+(S) + f^+(x) - f^-(S) - f^-(x)$$

On en déduit, d'après la définition et l'hypothèse de récurrence que

$$f^+(S\cup x) - f^-(S\cup x) = 0$$

□

corollaire 3.2. *Dans un réseau de transport,*

$$f^+(s) = f^-(t)$$

Démonstration. Posons $S = V - \{s,t\}$. Il est claire que le flot qui sort de s est égale au flot qui entre dans S, donc $f^+(s) = f^-(S)$; de même, le flot qui sort de S est égale au flot qui entre dans t, donc $f^+(S) = f^-(t)$. On en déduit, d'après la loi de conservation de flot par S que

$$f^+(s) = f^-(S) = f^+(S) = f^-(t)$$

□

3.2 Flot maximum et coupe minimum

Notons qu'il existe toujours un flot réalisable, le flot nul. Le problème est de savoir si un flot maximum existe et comment le calculer, dans le cas où la réponse est affirmative. Le théorème suivant montre que dans un réseau de transport, un flot maximum existe.

Théorème 3.3. *(Ford et Fulkerson, 1956)*
La valeur maximum d'un flot sur un réseau de transport $\mathcal{R}$ est égale à la capacité d'une coupe minimum.
Pour démontrer ce théorème, nous allons besoin de quelques notions et propriétés suivantes :

Définition 3.9. On appelle **valeur du flot** f, la quantité

$$v(f) = f^+(s) = f^-(t)$$

Proposition 3.4. *Pour toute coupe $\mathcal{C} = (X, Y)$,*

$$v(f) = f^+(X) - f^-(X)$$

Démonstration. Soit $\mathcal{C} = (X, Y)$ une coupe et soit $S = X - \{s\}$. Par définition, il est claire (voir la figure 3.4) que

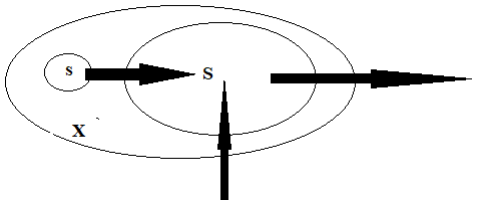


FIGURE 3.4

— tout le flot qui sort de s entre en S, c'est-ào-dire

$$f^+(s) = f(s, S)$$

— le flot qui sort de X est le même celui qui sort de S, ainsi

$$f^+(X) = f^+(S)$$

— le flot qui entre en S est égale au flot qui entre en X plus le flot qui sort de s, c'est-à-dire

$$f^-(S) = f^-(X) + f^+(s)$$

D'après la loi de conservation de flot en S et le fait que $v(f) = f^+(s)$, on en déduit que

$$f^+(X) = f^-(X) + f^+(s)$$

et donc

$$v(f) = f^+(s) = f^+(X) - f^-(X)$$

□

corollaire 3.5. *Sous l'hypothèse de la proposition ci-dessus, nous avons aussi*

$$v(f) = f^-(Y) - f^+(Y)$$

Proposition 3.6. *Pour toute coupe (X, Y) et pour tout flot f,*

$$v(f) \leq c(X, Y)$$

Démonstration. D'après la proposition ci-dessus, pour toute coupe $\mathcal{C} = \{X, Y\}$,

$$v(f) = f^+(X) - f^-(X) \leq f^+(X) \leq c(\mathcal{C})$$

□

Par la suite, c'est le corollaire suivant qui nous permettra de savoir si un flot est maximum.

corollaire 3.7. *Étant donnés un flot f et une coupe $\mathcal{C}$, si*

$$v(f) = c(\mathcal{C})$$

alors f est un flot maximum et $\mathcal{C}$ une coupe minimum.

Rappelons que dans un graphe orienté une chaine est une séquence de sommets

$$\mu = (x_1, x_2..., x_k)$$

telle que pour $i = 1...k-1$, x_i et x_{i+1} sont adjacents.

Définition 3.10. Un arc a est

— **positif** pour la chaine μ s'il existe i tel que

$$a = (x_i, x_{i+1})$$

— **négatif** s'il existe i tel que

$$a = (x_{i+1}, x_i)$$

;

— **saturé** si

$$c(a) = f(a)$$

.

Définition 3.11. (chaine améliorante)
Dans un réseau de transport, une chaine μ allant de s à t est une chaine **améliorante** si elle ne contient pas d'arc saturé. Autrement dit, pour tout arc $a \in \mu$, on a

$$\begin{cases} f(a) < c(a) \text{ si l'arc } a \text{ est positif} \\ f(a) > 0 \text{ si non} \end{cases}$$

Une chaine non améliorante est dite **saturée**.
Soit μ une chaine allant de s à t. Posons $\mu = \mu^+ \cup \mu^-$ où μ^+ (resp. μ^-) est l'ensemble des arcs positifs (resp. des arc négatifs) de μ.

$$\epsilon^+ = \min_{a \in \mu^+}(c(a) - f(a)) \; ; \; \epsilon^- = \min_{a \in \mu^-} f(a)$$

$$\delta_\mu = \min(\epsilon^+, \epsilon^-)$$

Remarque 3.2. Il est claire que la chaine μ est améliorante si et seulement si $\delta_\mu > 0$.
Lemme 3.8. *Soit f un flot dans un réseau. S'il existe une chaîne améliorante, alors il existe un flot f' tel que*

$$v(f) < v(f')$$

Démonstration. Soit μ une chaine améliorante et soit f' une application définie sur l'ensemble des arcs U comme suit

$$f'(a) = \begin{cases} f(a) + \delta_\mu \text{ si } a \in \mu^+ \\ f(a) - \delta_\mu \text{ si } a \in \mu^- \\ f(a) \text{ si } a \notin \mu \end{cases}$$

Vérifions que l'application f' est un flot et que

$$v(f') = v(f) + \delta_\mu$$

1. Contrainte de capacité :
   Soit a un arc. Puisque μ est une chaine améliorante, alors, par définition de δ_μ, on a

$$\begin{array}{l} 0 < f'(a) = f(a) + \delta_\mu \leq c(a) \text{ si } a \in \mu^+ \\ 0 \leq f(a) - \delta_\mu < c(a) \text{ si } a \in \mu^- \\ 0 \leq f'(a) = f(a) \leq c(a) \text{ si } a \notin \mu \end{array}$$

Ainsi, on a

$$\forall a \in U, 0 \leq f'(a) \leq c(a)$$

et ceci vérifie la contrainte des capacités.

2. Loi de conservation de flot :
   Soit x un sommet différent de la source et le puits.
   Si $x \notin \mu$, alors

$$f'^{+}(x) = f^{+}(x) = f^{-}(x) = f'^{-}(x)$$

   Si $x \in \mu$, alors il existe deux arcs de μ tels que
   — un arc entrant dans x et l'autre sortant de x ;
   — les deux arcs entrent dans x ;
   — les deux arcs sortent de x.

On peut vérifier facilement que dans tous les cas,

$$f'^{+}(x) = f'^{-}(x)$$

Par conséquent, f' est un flots. La valeur de f' est

$$v(f') = f'(s) = f(s) + \delta_\mu = v(f) + \delta_\mu$$

□

Proposition 3.9. *Un flot f dans un réseau de transport est maximum si et seulement s'il n'existe pas de chaine améliorante.*

Démonstration. D'après le lemme ci-dessus, la condition est nécessaire.
Pour montrer que la condition est suffisante, supposons que f est un flot pour lequel il n'existe aucune chaine améliorante. Soit S un ensemble de sommet constitué par la source s et tous les sommets x pour lesquels il existe une chaine insaturée allant de s à x. Par hypothèse, $t \notin S$, donc $\mathcal{C} = \{S, S^c\}$ est une coupe ; de plus, le flot sur $w^-(S)$ est nul et les arcs de $w^+(S)$ sont saturés, donc $f^-(S) = 0$ et $f^+(S) = c(\mathcal{C})$, et donc

$$v(f) = f^{+}(S) - f^{-}(S) = c(\mathcal{C})$$

Par conséquent f est maximum. □

Remarque 3.3. La démonstration du Théorème de Ford et fulkerson découle du lemme 3.8 et la proposition 3.9.

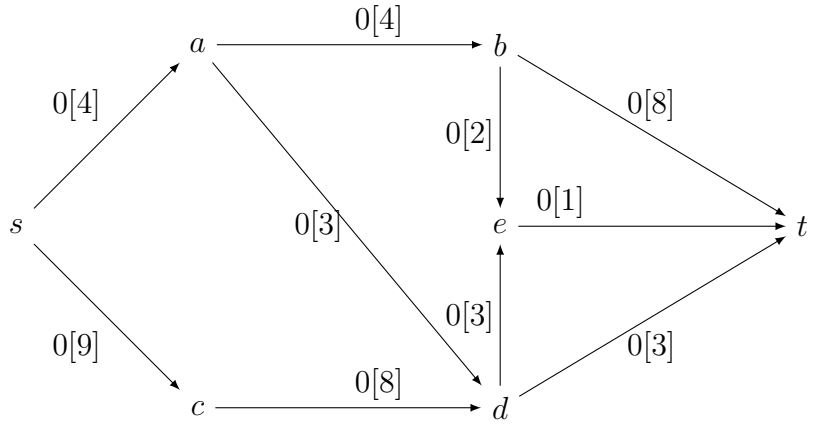
3.3 Calcul de flot maxsimum dans un réseau

Soit un réseau de transport $\mathcal{R} = (G, c)$ et soit f un flot qui, initialement nul, c'est à dire $f(a) = 0, \forall a \in U$. Soit X l'ensemble des sommets contenant s et les sommets x pour lesquels il existe une chaine non saturée allant de s à x.

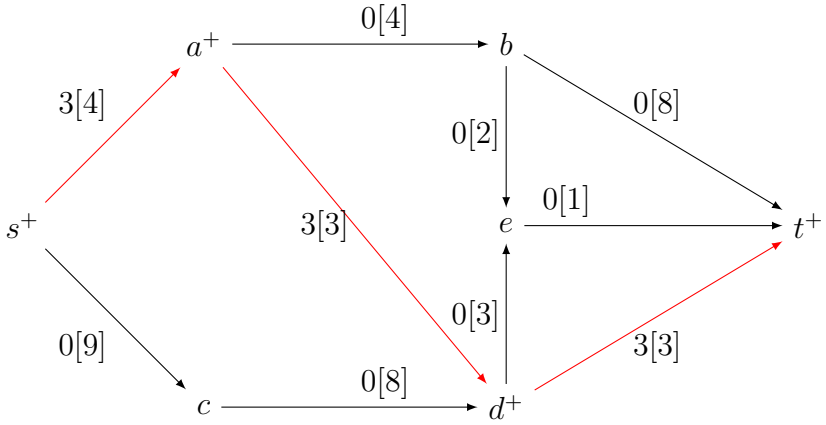
1. Si $t \notin X$, alors f est un flot maximum et $\mathcal{C} = (X, X^c)$ est une coupe minimum.
2. Sinon, on sélectionne une chaîne améliorante μ ; on détermine l'écart minimal δ_μ sur cette chaîne. Puis on met à jour la valeur du flot sur μ en remplaçant $f(a)$ par $f(a) + \delta_\mu$ si a est un arc positif et par $f(a) - \delta_\mu$ si a est un arc négatif. On retourne alors à l'étape initiale.

Remarque 3.4. L'algorithme de Ford-Fulkerson est composé de 2 procédures : la procédure de marquage et la procédure d'augmentation. La procédure de marquage marque les sommets x qui sont accessibles depuis le sommet s dans le réseau par une chaine non saturée. C'est cette procédure qui permet de déterminer s'il existe une chaine améliorante μ de s à t. Si une telle chaine n'existe pas, on s'arrête pour conclure que la valeur du flot actuel est maximum. Sinon, on applique la procédure d'augmentation qui permet d'augmenter le flot actuel sur la chaine μ. Ensuite on recommence avec la procédure de marquage après avoir mis à jour le réseau.

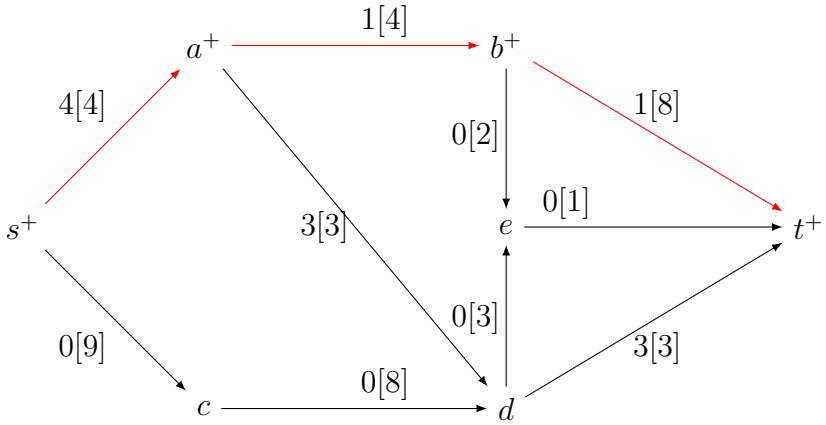
Exemple 3.1. Calculer le flot maximum sur le réseau suivant.



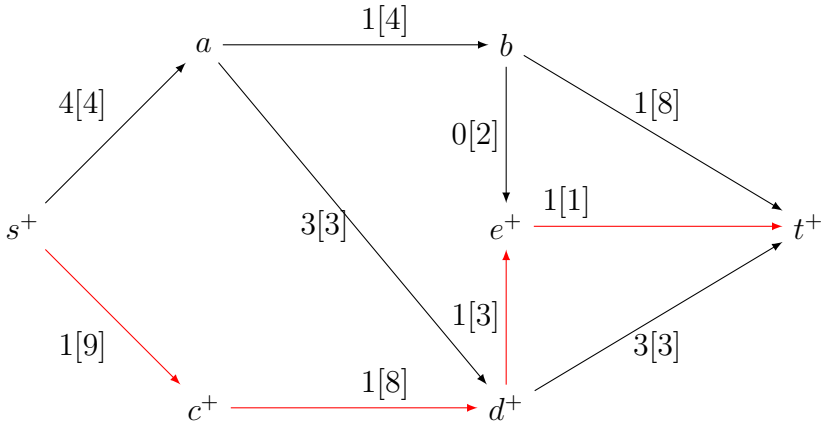
La chaine $\mu = sadt$ est améliorante.
$\delta_\mu = \min\{3, 4\} = 3$
Après la mise à jour du flot sur μ, on trouve $v(f) = 0 + 3 = 3$



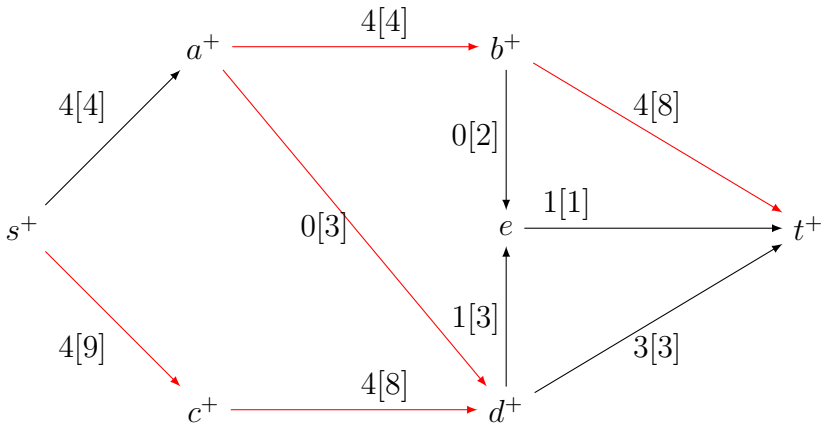
La chaine $\mu = sabt$ est améliorante.
$\delta_\mu = \min\{4-3, 4, 8\} = 1$
Après la mise à jour du flot sur μ, on trouve $v(f) = 3+1 = 4$



La chaine $\mu = scdet$ est améliorante.
$\delta_\mu = \min\{9, 8, 3, 1\} = 1$
Après la mise à jour du flot sur μ, on trouve $v(f) = 4+1 = 5$



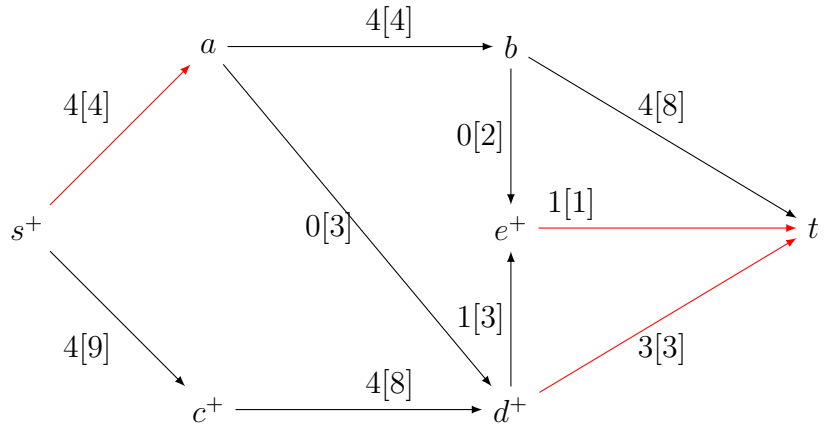
La chaine $\mu = scdabt$ est améliorante.
$\delta_\mu = \min\{9-1, 8-1, 3, 4-1, 8-1\} = 3$
Après la mise à jour du flot sur μ, on trouve $v(f) = 5+3 = 8$



On reprend la procédure de marquage, et on voit qu'il est impossible de marquer le puits t. Par ailleurs si on pose $X = \{s, c, d, e\}$, l'ensemble des sommets marqués, on aura, $\{X, X^c\}$ est une coupe dont la capacité est la somme des capacité sur les arcs rouges qui sortent de X, c'est-à-dire

$$c(X, X^c) = 8 = v(f)$$

Donc la valeur du flot maximum est $v(f) = 8$.



3.4 Flot maximum de coût minimum

Nous étudierons dans cette section le problème de flot quand des coûts sont affectés aux arcs du réseau. Dans de nombreux problèmes, outre les capacités limitées de transport, le coût...

Définition 3.12. (**Graphe d'écart**)
Soient $\mathcal{R} = (G, c)$ un réseau de transport et f un flot. On appelle graphe d'écart associé à f, le graphe $G_e(f)$ ayant les mêmes sommets que G, et tout arc (x, y) de G de capacité $c(x, y)$ est remplacé par deux arcs dans $G_e(f)$, (x, y) et (y, x) avec des capacités respectivement

$$c'(x, y) = c(x, y) - f(x, y) \text{ et } c'(y, x) = f(x, y)$$

Remarque 3.5. Notons que les arcs de capacité nulle sont inutiles et donc ils sont supprimés. Ainsi, un arc (x, y) tel que
— (x, y) est saturé dans $\mathcal{R}$, alors il est remplacé dans G_e par le seul arc (y, x) de capacité $c'(y, x) = f(x, y)$ (car dans ce cas $c'(x, y) = 0$;
— le flot sur (x, y) est nul ($f(x, y) = 0$), alors il est remplacé dans G_e par le seul arc (x, y) de capacité $c'(x, y) = c(x, y)$.
Par conséquent, si le flot f est nul, le graphe G coïncide avec son graphe d'écart.

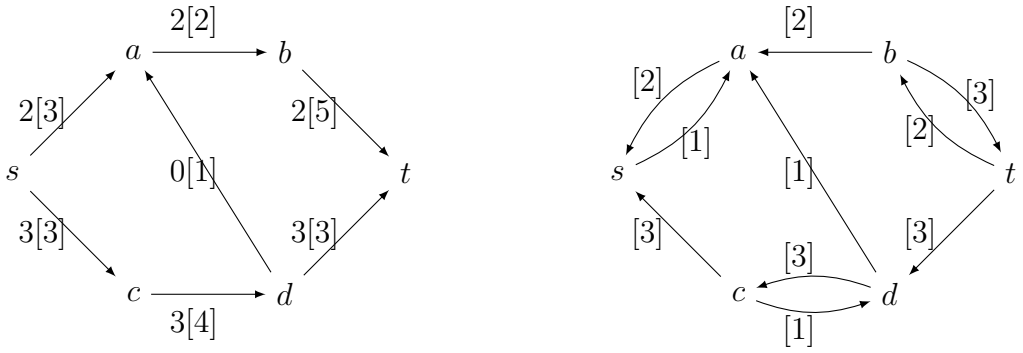


FIGURE 3.5 – Réseau de transport à gauche ; son graphe d'écart à droite.

Remarque 3.6. Par définition d'un graphe d'écart, il est facile de voir que le réseau $\mathcal{R}$ admet une chaine améliorante si et seulement si le graphe d'écart G_e admet un chemin de la source s au puits t. Ainsi, un flot f est maximum si et seulement s'il n'existe pas de chemin allant de s à p dans G_e. Dans le graphe d'écart de la figure 3.5, il n'y a aucun chemin allant de s à t, donc le flot défini sur $\mathcal{R}$ est maximum avec la valeur $v(f) = 5$.
Définition 3.13. (Flot avec coût)
On appelle réseau de transport avec coût le réseau $\mathcal{R} = (G, c, p)$ où $G = (V, U)$ et $p : U \longrightarrow \mathbb{R}$ est une application qui à tout arc (x, y) fait associer le coût unitaire $p(x, y)$, qui est le coût de transport d'une unité de flot sur l'arc (x, y). On utilise la notation $f[c, p]$ pour designer

le flot, la capacité et le coût unitaire sur chaque arc. Par exemple, dans la figure 3.6, sur l'arc (s, a), on a $f(s, a) = 2, c(s, a) = 3$ et son coût unitaire est $p(s, a) = 2$; ainsi, le coût du flot sur cet arc est $f(s, a) \times p(s, a) = 2 \times 2 = 4$.

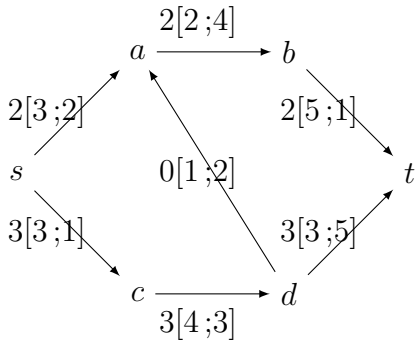


FIGURE 3.6 – Réseau de transport avec coût

Puisque f et p sont des applications sur l'ensemble des arcs U, donc f et p sont des vecteurs de $\mathbb{R}^m$, avec $m = |U|$. Ainsi, le coût total de f sur $\mathcal{R}$ est donné par le produit cartésien de f et p, c'est-à-dire

$$p(f) =< f, p >= \sum_{(x,y)\in U} f(x, y) \times p(x, y)$$

Le coût du flot sur le réseau de la figure 3.6 est

$$\begin{aligned} p(f) =< f, p > &= f(s, a).p(s, a) + f(s, c).p(s, c) + f(a, b).p(a, b) + f(d, a).p(d, a) \\ &\quad + f(c, d).p(c, d) + f(b, t)p(b, t) + f(d, t)p(d, t) \\ &= 2.2 + 3.1 + 2.4 + 0.2 + 3.3 + 2.1 + 3.5 \\ &= 41 \end{aligned}$$

Définition 3.14. (Graphe d'écart associé à un réseau avec coût)
Soit G_e le graphe d'écart associé à un réseau avec coût $\mathcal{R} = (G, c, p)$. Le coût des arcs sur le graphe d'écart G_e est défini comme suit : pour tout arc $(x, y) \in U$, on a

— si $f(x, y) < c(x, y)$, l'arc (x, y) est remplacé par les arcs (x, y) et (y, x) de capacité $p'(x, y) = p(x, y)$ et $p'(y, x) = -p(x, y)$;
— si $f(x, y) = c(x, y)$, l'arc (x, y) est remplacé par l'arc (y, x) de capacité $p'(y, x) = -p(x, y)$;
— si $f(x, y) = 0$, l'arc (x, y) est remplacé par l'arc (x, y) de capacité $p'(x, y) = p(x, y)$.

La figure 3.7 montre le réseau de transport avec coût, à gauche ; et son graphe d'écart, à droite.

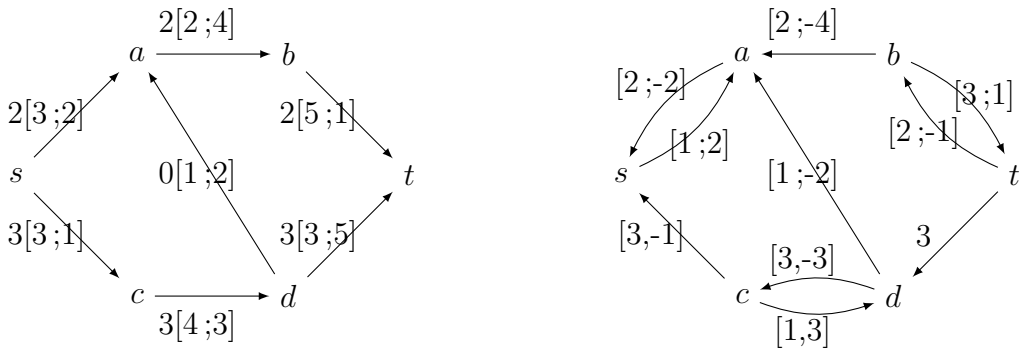


FIGURE 3.7 – Graphe d'écart d'un réseau avec coût

Il est clair que dans un réseau de transport, plusieurs flots peuvent avoir une valeur maximale, dans ce cas le choix de l'un ou l'autre n'importe pas. Cependant, si le réseau est muni d'un coût de transport, le choix se fera sur un flot qui minimise le coût. Étant donné un flot maximum, comment peut-on savoir s'il est à coût minimum ? Le théorème suivant répond à la question.

Théorème 3.10. *(théorème de Roy)*
Pour qu'un flot f soit maximum et à coût minimum il faut et il suffit que, dans le graphe d'écart, il n'existe ni de chemin allant de s à t ni de circuit de coût strictement négatif.

On a vu que la valeur et le coût du flot f de la figure 3.7 sont respectivement $v(f) = 5$ et $p(f) = 42$. Ainsi, f est maximum à coût minimum d'après le théorème de Roy. Pour calculer le flot maximum à coût minimum, nous utilisons l'algorithme de Roy ci-dessous

Algorithme 3.1. *1. Poser $f = 0$;*

2. *Construire $G_e(f)$;*
3. *Chercher un chemin de coût minimum dans $G_e(f)$ allant de s à p ;*
4. *S'il n'y a pas de chemin de s à p, le flot est maximum de coût minimum, alors FIN.*
5. *Sinon, améliorer le flot f sur la chaine qui correspond au chemin trouvé dans 3) et retourner en 2).*

Exemple 3.2. Soit $\mathcal{R}$ le réseau de transport avec coût défini par la figure 3.8 On applique l'algorithme de Roy pour calculer un flot maximum de coût minimum.

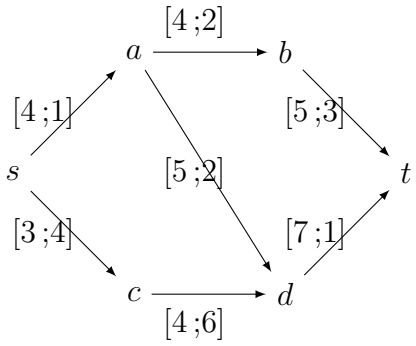


FIGURE 3.8

Itération 1. $f = 0$
Le graphe d'écart associé pour $f = 0$ coïncide avec le réseau $\mathcal{R}$ de la figure 3.8. Le plus court chemin allant de s à t est

$$\mu = (s, a, d, t)$$

; le même chemin est améliorant dans le réseau $\mathcal{R}$ et la quantité de flot supplémentaire sur ce chemin est

$$\delta = \min(c(s, a), c(a, d), c(d, t)) = 4$$

. Ainsi $v(f) = 0 + 4 = 4$.

Itération.2 $f = 4$. La figure 3.10 montre le réseau $\mathcal{R}$ avec le flot f de valeur 4 ainsi que le graphe d'écart associé. Le chemin de coût minimum dans $G_e(f)$ allant de s à t est

$$\mu = (s, c, d, a, b, t)$$

; dans le réseau $\mathcal{R}$, $\mu = (s, c, a, b, t)$ est une chaine améliorante et la quantité de flot supplémentaire sur cette chaine est

$$\delta = \min(c(s, c), c(c, d), f(d, a), c(a, b), c(b, t)) = 3$$

. Ainsi $v(f) = 4 + 3 = 7$.

Itération.3 $f = 4$. La figure 3.10 montre le réseau $\mathcal{R}$ avec le flot f de valeur 7 ainsi que le graphe d'écart associé.
Il n'y a aucun chemin dans $G_e(f)$ allant de s à t, donc le flot actuel f est maximum de coût minimum avec

$$v(f) = 7, p(f) = 55$$

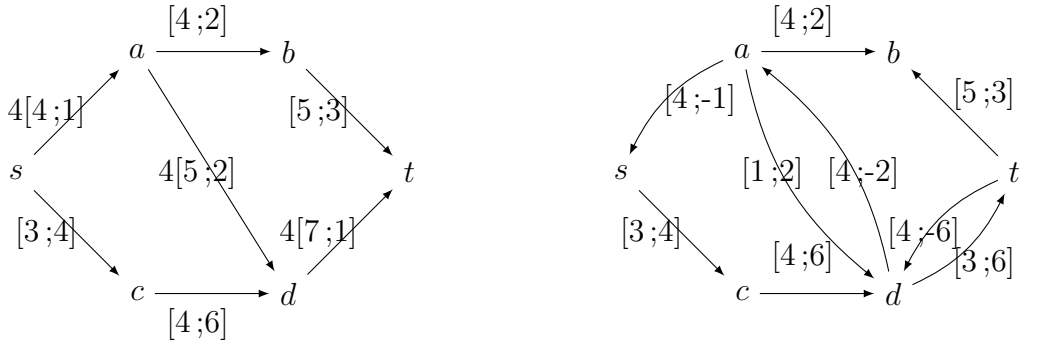


FIGURE 3.9 – Réseau de transport à gauche ; son graphe d'écart à droite.

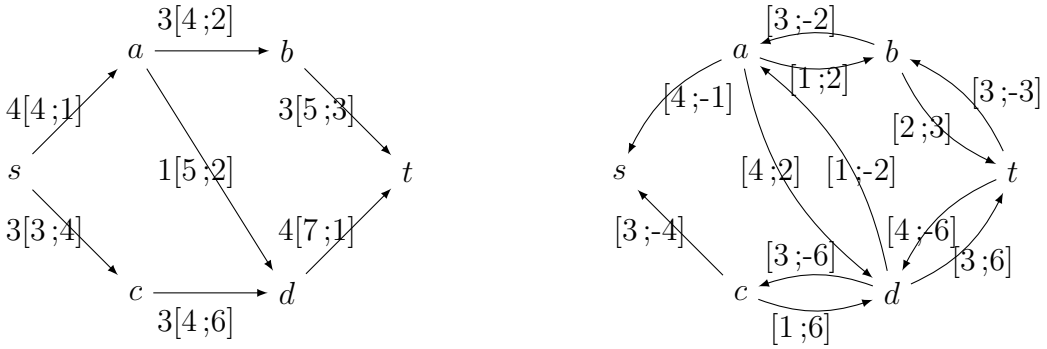


FIGURE 3.10 – Réseau de transport à gauche ; son graphe d'écart à droite.

CHAPITRE 4

QUELQUES APPLICATIONS

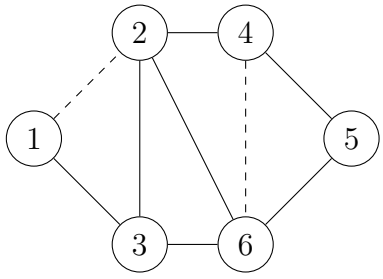
Introduction
Dans ce chapitre, nous présentons le lien et l'intérêt du flot avec d'autres notions en théorie des graphes, en particulier, avec celles du couplage et d'affectation dans les graphes bipartis. Nous donnons aussi quelques problèmes concrets dont la solution se ramène au calcul d'un flot maximum avec ou sans coût dans un réseau de transport.

4.1 Couplage dans les graphe biparti

Définition 4.1. Un **couplage** (matching) dans un graphe $G = (V, E)$ est un ensemble d'arêtes $M \subseteq E$ qui sont deux à deux disjointes ou non incidentes à un même sommet.
Dans la figure 4.1 l'ensemble des arêtes $M = \{12, 46\}$ est un couplage.

Un sommet du graphe est dit **saturé** par un couplage M (ou M -saturé), s'il est extrémité d'une arête de M ; sinon, il est dit **insaturé** par M (ou M-insaturé). Un couplage M est parfait (perfect matching) s'il sature tous les sommets de G. Les sommets 1,2,4 et 6 sont saturés par le couplage $M = \{12, 46\}$, les sommets 3 et 5 sont M-insaturés donc M n'est pas parfait.

Un couplage M est dit **maximal** si on ne peut pas l'augmenter, c'est à dire $M \cup \{v\}, \forall v \in V$ n'est pas un couplage. Autrement dit, il n'existe pas un autre couplage contenant strictement M. Un couplage M est **maximum** si son cardinal $|M|$ est maximum, c'est à dire il



G

FIGURE 4.1 – Couplage

n'existe pas d'autre couplage ayant plus d'arêtes que M. On peut vérifier que le couplage M de la figure 4.1 est maximal mais il n'est pas maximum, car $M' = \{13, 26, 45\}$ est un autre couplage avec une cardinalité $|M'| = 3 > |M| = 2$.

Remarque 4.1. 1. Il est claire qu'un couplage maximum est maximal et qu'un couplage parfait est maximum ;

2. Un graphe avec un nombre de sommets impair ne peut pas avoir un couplage parfait.

3. Un couplage parfait n'existe pas toujours même si le nombre de sommets est pair.

Définition 4.2. le nombre d'arêtes dans un couplage maximum d'un graphe G est appelé l'**indice de couplage** de G, et il est dénoté $\nu(G)$ (on lit nu de G).

D'un point de vue combinatoire, le problème consiste à trouver un couplage maximum :

Problème du couplage maximum (Maximum Matching)
Instance : Un graphe $G = (V, E)$
Objectif : Déterminer un couplage maximum de G (ou calculer $\nu(G)$.

Définition 4.3. Une chaîne **alternée** relativement au couplage M, ou chaîne M -alternée, est une chaîne élémentaire μ dont les arêtes sont alternativement dans M et dans $E \setminus M$. Une chaîne M-alternée est dite **augmentant** pour M, ou M-augmentant, si ses extrémités sont insaturées.

Remarque 4.2. 1. Par définition, si μ est une chaine alternée pour M, $\mu \setminus M$ est aussi un couplage.

2. La longueur d'une chaine M-augmentante est impaire (attention, une chaine alternée impaire n'est pas nécessairement augmen-

tante).
Dans la figure 4.1, $\mu_1 = (1, 2, 6, 4, 5)$ et $\mu_2 = (3, 1, 2, 4, 6, 5)$ sont des chaines alternées. La chaine μ_2 est une chaine augmentante pour le couplage $M = \{12, 46\}$.

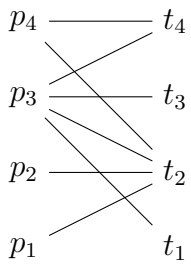
Le Théorème suivant donne une condition nécessaire et suffisante pour qu'un couplage M soit maximum.
Théorème 4.1. *(Berge)*
Un couplage M d'un graphe $G = (V, E)$ est maximum si et seulement s'il n'existe pas de chaîne M-augmentante.
Le calcul d'un couplage maximum dans un graphe biparti peut se ramener au calcul d'un flot maximum dans un réseau de transport. En effet, soit $G = (X, Y, E)$ un graphe biparti. On ajoute une source s qu'on connecte à tous les sommets de X par des arcs, un puits t tel que tout sommet de Y est connecté à t par un arc et chaque arête de G est remplacée par un arc orienté de X vers Y. On rajoute une capacité 1 sur chaque arc. Ainsi, si f est un flot maximum calculé par l'algorithme de Ford et Fulkerson, alors le couplage maximum dans G est donné par les arcs saturés allant de X vers Y.

Exemple 4.1. Soit $P = \{p_1, p_2, p_3, p_4\}$ un ensemble de personnes à qui on demande de faire l'ensemble des taches $T = \{t_1, t_2, t_3, t_4\}$.
La matrice A ci-dessous est définie comme suit : $a_{ij} = 1$ si et seulement si p_i est capable de faire t_j. On suppose qu'une même personne ne peut pas réaliser plus d'une tache simultanément.

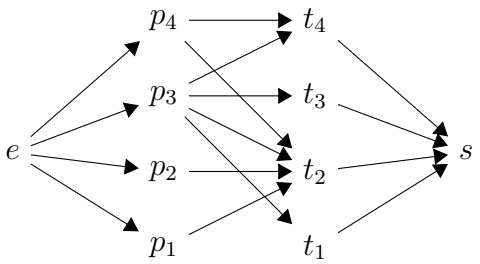
$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

P représente les lignes et T les colonnes. Utilisons la notion de flot pour calculer le nombre de tâches qui puissent être exécutées en même temps.

On associe à la matrice A le graphe biparti $G = (P, T, E)$ où $P = \{p_1, p_2, p_3, p_4\}$ représente les lignes et $T = \{t_1, t_2, t_3, t_4\}$ représente les colonnes et tel que deux sommets p_i et t_j sont adjacents si et seulement si la personne p_i peut exécuter la tache t_j. On trouve ainsi le graphe suivant :



Le nombre de tâches qui puissent être exécutées en même temps est donné par un couplage maximum sur ce graphe biparti. Pour calculer ce couplage on procède comme suit : on rajoute une source e qu'on connecte à tous les sommets de P par des arcs, un puits t tel que tout sommet de T est connecté à s par un arc et chaque arête de G est remplacée par un arc orienté de P vers T. Enfin, nous associons à chaque arc une capacité 1. On aura ainsi le réseau de transport ci-dessous.



En appliquant l'algorithme de Ford et Fulkerson, on trouve $v(f) = 3$. Par conséquent, un couplage maximum est constitué par 3 élément, et donc le nombre de tâches qu'on peut exécuter en même temps est 3.

Définition 4.4. Un ensemble de sommets T est un **transversal** ou **couverture de sommets** (vertex cover) si toute arête de E possède, au moins, une extrémité dans T. Autrement dit $T \cap e \neq \emptyset$ pour toute arête $e \in E$. Un transversal T est minimal si $\forall v \in T, T \cap v$ n'est pas un transversal, c'est à dire le sous ensemble de T obtenu par suppression d'un seul élément n'est pas un transversal. Un transversal T est minimum si pour tout transversal T', $|T'| \geq |T|$. Le nombre de sommets dans un transversal minimum de G est noté $\tau(G)$.

Proposition 4.2. *Pour tout graphe G on a*

$$\nu(G) \leq \tau(G) \tag{4.1}$$

Démonstration. Soit M un couplage maximum, donc $|M| = \nu(G)$. Puisque les arêtes de M sont deux à deux disjointes, pour les couvrir il faut au moins $\nu(G)$ sommets. Ainsi, tout transversal a au moins

$\nu(G)$ sommets et donc $\nu(G) \leq \tau(G)$. □

Remarque 4.3. Notons que si M est un couplage et T un transversal tels que $|M| = |T|$ alors M est un couplage maximum et T est un transversal minimum.

Exemple 4.2. La figure 4.2 présente un couplage maximal $M = \{14, 36\}$. Le couplage M n'est pas maximum !

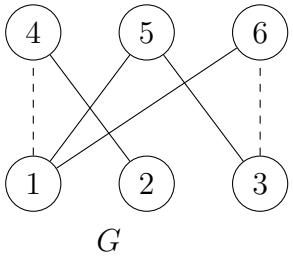


FIGURE 4.2 – Couplage

4.2 Théorème de König

Théorème 4.3. *(KÖNIG)*
Soit G un graphe biparti. Alors

$$\nu(G) = \tau(G)$$

Ce résultat est aussi une expression d'égalité entre la valeur d'un flot maximum et la capacité d'une coupe minimum dans un réseau biparti. En effet, on construit le réseau $\mathcal{R}$ à partir du graphe biparti $G = (X, Y, E)$ comme suite : chaque arrête de G est orienté de X vers Y et reçoit une capacité ∞. On ajoute une source s qui sera adjacente à tous les sommets de X par des arcs $(s, x), x \in X$ avec une capacité 1 pour chacun des ces arcs. D'autre part, on ajoute un puits t qui sera adjacent à tous les sommets de Y par des arcs $(y, t), y \in Y$ avec une capacité 1 pour chaque arc. Ainsi, on peut vérifier facilement que

1. si f est un flot sur ce réseau, alors l'ensemble des arcs de flot égal à 1 et qui connectent les sommets de X aux sommets de Y, est un couplage dans G ; et la valeur de f est égale aux nombre d'éléments dans ce couplage.
2. si $\mathcal{C} = \{X', Y'\}$ est une coupe, alors l'ensemble $T = (X \setminus X') \cup (X' \cap Y)$ est un transversal dont le nombre de sommets est égal à la capacité de cette coupe.

Ainsi, de 1) et 2), on déduit :
— si f est un flot maximum dans $\mathcal{R}$ alors

$$v(f) = \nu(G)$$

— si $\mathcal{C}$ est une coupe minimum dans $\mathcal{R}$, alors

$$c(\mathcal{C} = \tau(G)$$

par conséquent, le théorème de Ford-Fulkerson donne alors directement l'égalité citée dans le théorème de König.
Exemple 4.3. Dans la figure 4.3, nous avons un réseau de transport

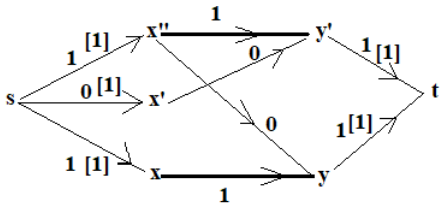


FIGURE 4.3 – Théorème de König

défini à partir du graphe biparti $G = (X, Y, E)$, où $X = \{x, x', x"\}$; $Y = \{y, y'\}$, et un flot f de valeur $v(f) = 2$. Par ailleurs, $\mathcal{C} = \{X', Y'\}$, avec $X' = \{s\}, Y' = \{x, x', x", y, y', t\}$, est une coupe avec une capacité $c(\mathcal{C}) = 2$, donc le flot f est maximum.
On peut voir facilement que l'ensemble des arcs connectant X à Y de flot égal à 1 est $M = \{(x", y'), (x, y)\}$ est un couplage maximum dont le nombre d'éléments égal à $2 = v(f)$; et l'ensemble des sommets $T = (X \setminus X') \cup (X' \cap Y) = \{y, y'\}$ est un transversal minimum dans G, avec $|T| = c(\mathcal{C}) = 2$.

4.3 Réseau de distribution d'eau

Soient trois châteaux d'eau, A, B et C alimentant quatre villages D, E, F et G. Le château d'eau A bénéficie d'une alimentation et d'une réserve capables de débiter 45 l/s ; le château d'eau B peut seulement débiter 25 $1/s$ et le château d'eau C, 20 $1/s$. Les différentes canalisations connectant les châteaux aux villages ainsi que leurs débits sont données par le tableau ci-dessous.

	D	E	F	G
A	10	15	0	20
B	10	5	10	0
C	0	0	10	10

Remarque 4.4. Sur le tableau ci-dessus, le nombre T_{ij}, donné par l'intersection de ligne i et la colonne j, représente le débit (ou capacité) maximal, en l/s, de la canalisation connectant le château i au village j. Par exemple, $T_{11} = 10l/s$ est la capacité maximale de la conduite allant du château A au village D. Ainsi, $T_{ij} = 0$ signifie qu'il n'y a aucune conduite allant du château i au village j, $T_{24} = 0l/s$, donc pas de conduite entre le château B et le village G.
Les besoins des village en débit sont comme suit : 30 l/s pour le village D, le village E, 10 $1/s$, le village F, $201/s$ et enfin le village G, 30 $1/s$.

Question : Est ce que la demande de chaque village en eau est satisfaite ? Dans le cas contraire, entre quels points il conviendrait de construire des canalisations supplémentaires ?

Pour répondre à cette question, soit $\mathcal{R} = (G, s, t, c)$ le réseau de transport biparti, donné par la figure 4.4, et définie comme suit :
— $G = (X, Y, U)$ est un graphe biparti, avec $X = \{A, B, C\}$, $Y = \{D, E, F, G\}$, et pour $x \in X, y \in Y$ tel que $T_{xy} \neq 0$, $(x, y) \in U$ et $c(x, y) = T_{xy}$;
— le sommet s est connecté à l'ensemble des châteaux par des arcs $(s, A), (s, B), (s, C)$ dont les capacités sont respectivement 45, 25 et 20.
— l'ensemble des villages est connecté au puits t par les arcs $(D, t), (E, t), (F, t)$ et (G, t), et dont les capacités sont respectivement 30, 10, 20 et 30.
L'ensemble des village aura besoin d'un débit total qui est égal à 90 l/s. Ce débit est satisfait si et seulement s'il existe un flot sur le réseau $\mathcal{R}$ de valeur 90. Pour le savoir, on peut appliquer l'algorithme de Ford et Fulkerson. La figure 4.5 présente le même réseau après 8 itérations de l'algorithme de FORD-FULKERSON. Les arcs rouge sont tous saturés et la valeur actuel de flot f est

$$v(f) = f^+(s) = 20 + 25 + 35 = 80$$

. Remarquons que la procédure de marquage permet de marquer uniquement les deux sommets s et A, et que $\mathcal{C} = \{S, S^c\}$, avec $S = \{s, A\}$ est une coupe de capacité $c(\mathcal{C}) = 80 = v(f)$. Ceci montre que la valeur

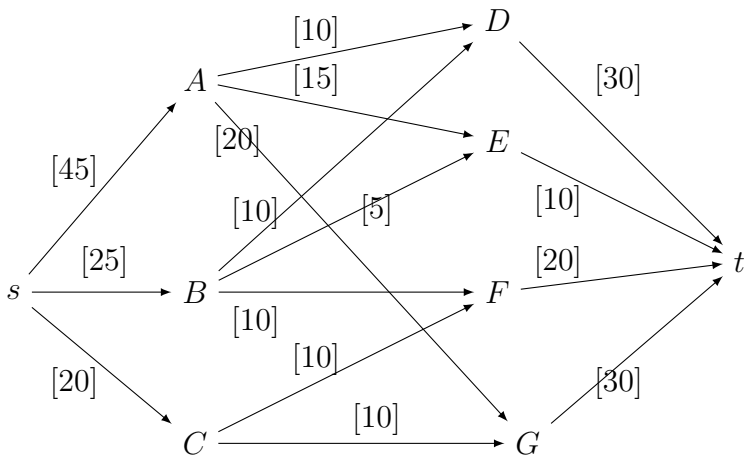


FIGURE 4.4 – Réseau de distribution d'eau

maximale de f est 80. Par conséquent, la demande des villages qui est de 90 l/s n'est pas satisfaite, car $v(f)_{\max} = 80 < 90$. Le seule village dont le besoin en débit n'est pas satisfait est le village D. En effet, son besoin est 30 l/s, mais sur ce réseau ne peut pas recevoir plus de 20 l/s. D'après le réseau de la figure 4.5, pour que le besoin du village D soit satisfait, il faut détruire la conduite (A, D) de capacité 10 l/s et la remplacer par une autre conduite (A, D) de capacité au moins 20 l/s

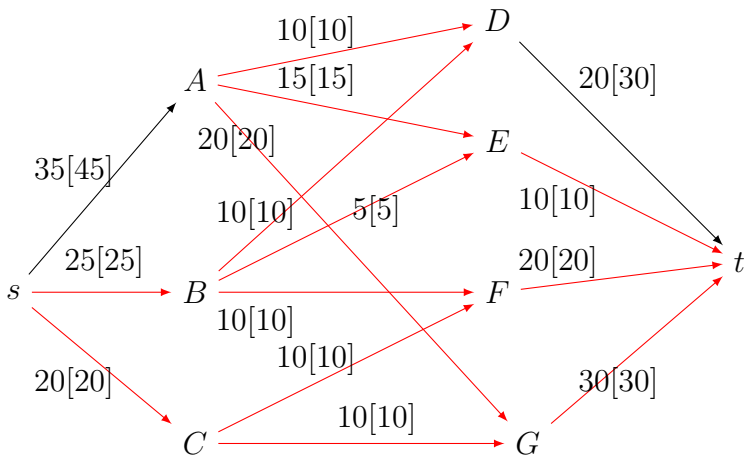


FIGURE 4.5 – Réseau de distribution d'eau après 8 itérations

CHAPITRE 5

CONCLUSION GÉNÉRAL

Dans ce mémoire, nous nous sommes intéressés au problème de flot qui est l'un des problèmes les plus important de l'optimisation combinatoire. Le problème de flot dans un réseau de transport peut être abordé par plusieurs approches, et nous avons choisi celle utilisant les graphes. Après un rappel détaillé de quelques notions de graphes et quelques algorithmes sur le problème de cheminement, nous avons passé à l'étude du problème de flot maximum. Par la suite, nous nous sommes intéressés à l'étude du problème du flot maximum de coût minimum. Les applications du problème de flot sont multiples, nous nous sommes limités à son application pour calculer un couplage maximum dans un graphe biparti, pour étudier un réseau de distribution d'eau et aussi pour retrouver le théorème de König.

BIBLIOGRAPHIE

[1] Claude Berge, Graphe et hypergraphes, édition 3, Dunod, 1983.

[2] T. Corman, C. Leiserson, R. Rivest, C. Stein, Introduction à l'algorithmique, Cours et exercices, édition 2, Dunod 2001.

[3] M.Sakarovitch, Optimisation Combinatoire et programmation discrète, Hermann, 1984.

[4] J.C. Fournier, Théorie des graphes et application, Édition 2, Lavoisier.

[5] D. Jungnickel, Graphs, Networks and Algorithms, Springer, Berlin Heidelberg New York Dordrecht London (2013).