UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES
Département des Mathématiques

Licence Sciences et Techniques (LST)

MATHEMATIQUES ET APPLICATIONS

MEMOIRE DE FIN D'ETUDES

Pour l'obtention du Diplôme de Licence Sciences et Techniques

Modélisation et Résolution des problèmes de flots dans les réseaux de transport : Application au réseau de transport urbain de Fès

Présenté par :

- ADROUJI Anass

Encadré par :

- Pr. ETTAOUIL Mohamed

Soutenu Le 7 Juin 2018 devant le jury composé de:

- Pr. ETTAOUIL Mohamed
- Pr. EZZAKI Fatima
- Pr. EL KHAOULANI EL IDRISSI Rachid
- Pr. RAHMOUNI HASSANI Aziza

Année Universitaire 2017 / 2018

FACULTE DES SCIENCES ET TECHNIQUES FES – SAISS
B.P. 2202 – Route d'Imouzzer – FES
☎ 212 (0)5 35 61 16 86 – Fax : 212 (0)5 35 60 82 14
Site web: http://www.fst-usmba.ac.ma

Table de Figures

Dédicace

Avant d'entamer ce rapport, je dédie ce modeste trava

Aux membres de ma famille, plus précisément à mes chers parents pour leurs soutiens et leurs encouragements moraux et financiers tout le long de mes études. Ainsi que mon frère et mes sœurs pour leurs soutiens incomparable.

A mes amis et collègues pour m'avoir apporté leur soutien, amitié et compréhension.

Aux toutes les personnes qui me reconnaissent et m'ont aidé et contribué à la réalisation de ce travail.

Remerciements

Louange à DIEU TOUT PUISSANT de m'avoir accordé la force d'accomplir cet humble travail.

Avant de commencer mon rapport, je tiens à remercier énormément les personnes qui ont intervenu pour m'aider à la réalisation de ce projet de fin d'étude :

Je tiens à exprimer mes vifs remerciements pour mon grand et respectueux professeur, **Mr. ETTAOUIL Mohamed**, d'avoir accepté de m'encadrer pour mon projet de fin d'études, ainsi que pour son soutien, ses remarques pertinentes et son encouragement.

Mes profonds remerciements vont aussi à Mr. Nour-eddine Joudar pour son aide, sa disponibilité et son encouragement.

Mes remerciements s'adressent également aux honorables membres du jury d'ayant accepté d'examiner notre modeste travail et de siéger à ma soutenance.

Mes remerciements vont aussi à tous mes professeurs, enseignants et toutes les personnes qui m'ont soutenus jusqu'au bout, et qui n'ont pas cessé de me donner des conseils très importants en signe de reconnaissance.

INTRODUCTION

La recherche opérationnelle est une discipline permettant de fournir des méthodes qui répondent à un type précis des problèmes, c'est-à-dire élaborer une démarche universelle qui aboutit à des solutions plus efficaces. Ces méthodes proposées sont des démarches basées sur des concepts ou des outils mathématiques (et/ou) statistiques.

Parmi les problèmes fréquents traités dans la recherche opérationnelle, les problèmes du plus court chemin et flot maximal sont les plus investis dans le réseau du transport urbain. En effet, ce service est un domaine très important et vital que tout collectif urbain est obligé d'assurer à ses habitants. Dans ce cadre, nous allons exploiter ces techniques pour résoudre l'un de secteur de transport dans la ville de Fès.

Parmi les moyens de transport les plus utilisés dans la ville de Fès, le secteur des bus constitue le monde de transport le plus courant et souvent le moins coûteux pour les habitants, le plus sécurisé et le plus assuré, il est dirigé par l'établissement privé « city bus », cette établissement doit fournir des lignes, dans le but de faciliter la tâche aux habitants de se déplacer vers leurs destinations sous des contraints d'embouteillage et de plus court chemin.

Ce rapport est réparti en quatre chapitres :

Le premier chapitre présente quelques notions de base de la théorie des graphes et des points généraux de la programmation linéaire.

Dans le deuxième et le troisième chapitres, nous allons définir les problèmes du plus court chemin et de flot maximal et leurs algorithmes de résolution.

Dans le dernier chapitre, nous allons exploiter les méthodes de Dantzig, Ford-Fulkerson et simplexe afin de résoudre un problème de réseau de transport urbain.

Nous allons terminer ce rapport par une conclusion générale.

Chapitre 1 : Eléments de la théorie des graphes

L'objectif de ce chapitre est de rappeler quelques notions de base de la théorie des graphes, et des points généraux de la programmation linéaire, qu'on va utiliser par la suite dans le traitement des problèmes de plus court chemin et de flot maximal.

1. Concept sur les graphes

<u>Définition d'un graphe :</u>

On appelle un graphe un couple G=(X, U) tel que :

- X un ensemble d'élément appelés **sommets** « nœuds » du graphe.
- U une famille d'élément du produit cartésien $X \times X$ appelés **arcs** du graphe s'il est orienté, sinon on les appelle **arêtes**.

<u>EXEMPLE :</u>

Un exemple de graphe à 8 sommets, nommés de « a » à « h », comportant 10 arêtes.

On le note le graphe par G=(X, U)

L'ensemble des sommets est X= {a, b, c, d, e, f, g, h}

Et l'ensemble des arêtes est : U={(a,d),(b,c),(b,d),(d,e),(e,c),(e,h),(h,d),(f,g),(d,g),(g,h)}

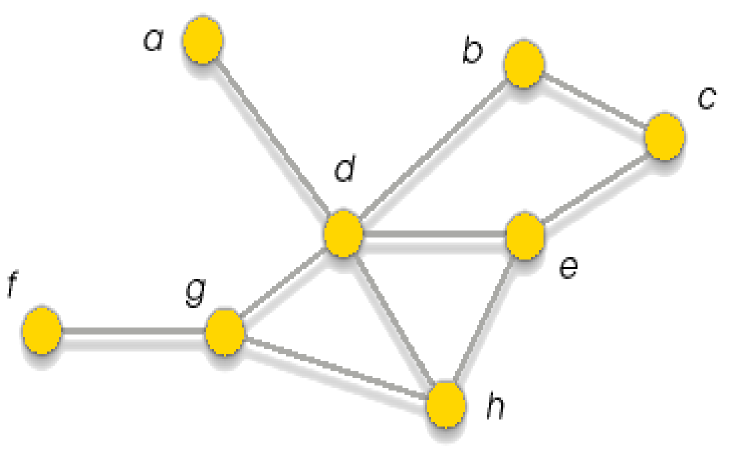


Figure 1 : Graphe de 8 sommets et 10 arêtes

1.1 Graphe orienté et Graphe non orienté

- Soit G=(X, U) un graphe, on dit que G est **orienté** si les couple (u_i, u_j) sont orientés où u_i est le sommet initiale et u_j le sommet terminal (u_i le prédécesseur de u_j et u_j le successeur de u_i)

  Le couple (u_i, u_j) est appelé un arc, représenté graphiquement par $u_i \rightarrow u_j$

- Soit G=(X, U) un graphe, si U est une relation symétrique sur X, on dira que G est un graphe **non orienté**. Autrement dit si $\forall u_1, u_2 \in U, (u_1, u_2) \in U \Rightarrow (u_2, u_1) \in U$

1.2 Graphe valué

Un graphe valué G=(X, U, c) est un graphe muni d'une application $c : U \rightarrow \mathbb{R}$

L'application c est appelée poids des arcs.

EXEMPLE :

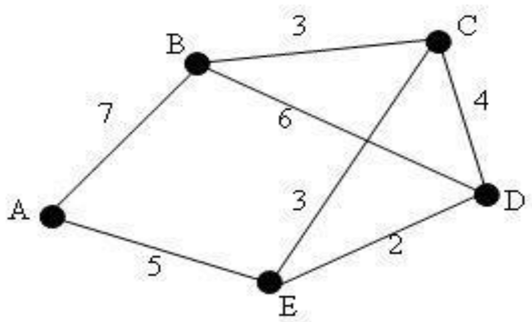


Figure 2 : Graphe valué

VOCABULAIRE :

- Deux sommets x et y sont adjacents s'il existe l'arête (x, y) dans U.
  Les sommets x et y sont alors dits voisins
  Par exemple dans le figure 1, a et d sont adjacents car (a, d) ϵ U, mais a et b ne sont pas adjacents car (a, b) $\notin$ U.
- On appelle ordre du graphe G = (X, U) le nombre de sommets du graphe
  L'ordre de G est donc le cardinal de X noté |X|.
  Dans l'exemple de la figure 1, on a un graphe d'ordre 8.
- Une boucle d'un graphe est une arête ou arc partant d'un sommet et allant vert lui-même.
  Soit $u = xx'$ un arc si $x = x'$alors u une boucle
- Le degré d'un sommet x de G est le nombre d'arêtes incidents à x.
  Il est noté d(x).
  Dans l'exemple de la figure 1 *d(g)* = 3 car on a 3 arêtes incidentes à g qui sont : (f, g), (h, g), (d, g).

Pour un graphe orienté : soit un x sommet du graphe G,
Le degré de x est : $d(x) = d^{+}(x) + d^{-}(x)$ avec $d^{-}(x)$ = nombre de sommets successeur de x et $d^{+}(x)$ = nombre de sommets prédécesseur de x.

Chemin

Un chemin est une liste $p = (x_1, \dots, x_n)$ des sommets telle qu'il existe dans le graphe une arête entre chaque paire de sommets successifs.

EXEMPLE :

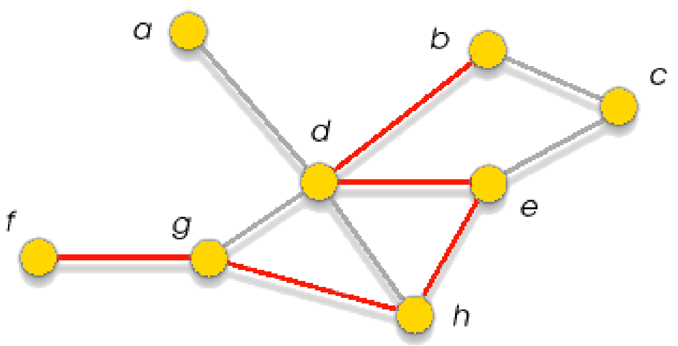


Figure 3 : Chemin reliant f à b

Le chemin relie les sommets « f » à « b » est : $p = (f, g, h, e, d, b)$

- La longueur d'un chemin est la somme des longueurs de chaque arc (ou arête) qui le compose.

Dans l'exemple précédent on a un chemin de longueur 5

Cycle

Un cycle est un chemin simple finissant à son point de départ $x_1 = x_n$

EXEMPLE :

Dans l'exemple précédent (d, h, e, d) est un cycle.

Chaine

Une chaine de longueur p une suite de p arcs de G telle que chaque arc de la suite ait au moins une extrémité avec l'arc suivant.

Circuit

C'est un chemin $\mu = (\mu_1, \dots, \mu_r)$ tel que l'extrémité initiale de l'arc μ_1 coïncide avec l'extrémité terminal de l'arc μ_r.

2. Matrices associées à un graphe

2.1 Matrice d'adjacence

La matrice d'adjacence ou d'incidence sommets-sommets d'un graphe G=(X, U) est une matrice carré où chaque ligne correspond à un sommet de G et chaque colonne correspond également à un sommet de G.

Le terme générale de la matrice $A = (a_{ij})$ « i, j ∈ {1,…, n} » est :

$$\begin{cases} a_{ij} = 1 \; si \; (i, j) \in U \\ a_{ij} = 0 \; Sinon \end{cases}$$

EXEMPLE :

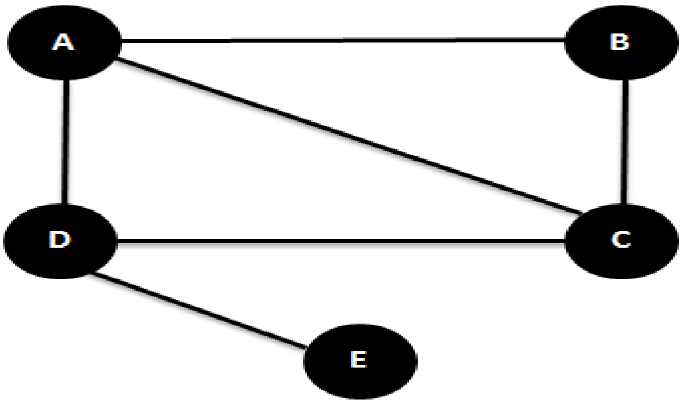


Figure 4 : Exemple de graphe non orienté

La matrice d'adjacence du graphe non orienté précédent est :

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

2.2 Matrice d'incidence sommets-arcs

La matrice d'incidence sommets-arcs d'un graphe G=(X, U) sans boucle est une matrice telle que chaque ligne correspond à un sommet de G et chaque colonne correspond à un arc de G; si u= (i, j) ∈ U, la colonne de la matrice A= (a_{ij}) a tous ses termes nuls sauf :

$$\begin{cases} a_{ij} = +1 \\ a_{ji} = -1 \end{cases}$$

<u>EXEMPLE :</u>

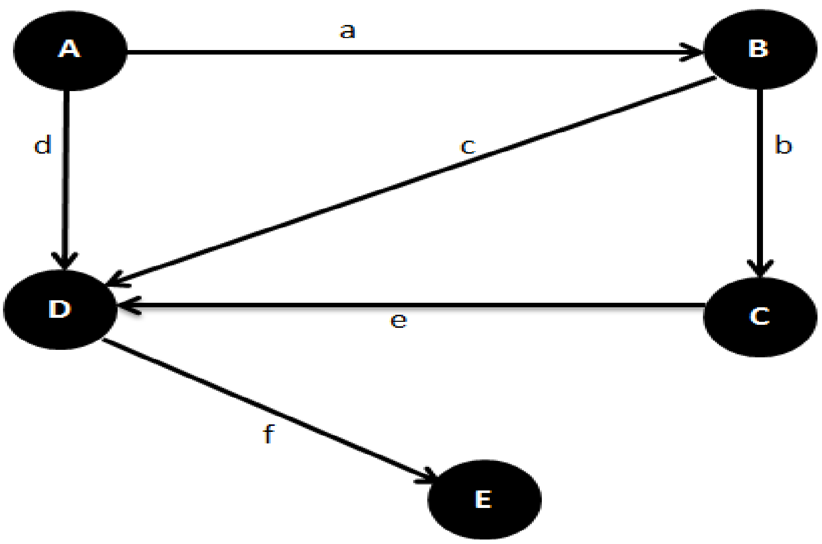


Figure 5 : Exemple de graphe orienté

La matrice d'incidence sommets-arcs du graphe orienté précédent est :

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \end{array}$$

3. Programmation linéaire en nombres binaire

La programmation linéaire est un outil très puissant de la recherche opérationnelle, qui peut résoudre un grand nombre de problème.

L'objectif de la programmation linéaire est de trouver la valeur optimale d'une fonction linéaire sous un système d'équations, d'inégalités de contraintes linéaires.

Parmi les méthodes les plus connues pour résoudre les programmes linéaires, nous trouvons la méthode de simplexe.

3.1 Modélisation

Avant de résoudre un problème, il faut bien commencer par sa traduction par des relations mathématiques.

Les étapes pour modéliser un problème sont les suivantes :

- **Etape 1 :** déterminer les variables de décision et les formuler de façon algébrique.
- **Etape 2 :** déterminer les contraintes et les formuler comme des équations ou des inéquations dépendantes des variables de décision.
- **Etapes 3 :** déterminer la fonction objective.

3.1.1 Variables

Lorsque le bus se trouve au point « i », alors pour chaque point « j »($i \neq j$), nous avons deux choix : le bus passe de i à j ou non. D'où la nécessite d'utilise des variables binaires. Notons par :

$$x_{ij} = \begin{cases} 1 \text{ si le bus passe du station «i» au station «j»} \\ 0 \text{ sinon} \end{cases}$$

3.1.2 Fonction objective

La fonction objective contient des coûts associés à chacune des variables.

La distance entre la station « i » et la station « j » est C_{ij}. Donc la distance totale est :

$$z = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} C_{ij} X_{ij}$$

Nous cherchons évidemment à rendre z aussi petit possible. Nous devons donc minimiser z :

$$\min\left(z = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} C_{ij} X_{ij}\right)$$

3.1.3 Contraintes

Les contraintes sont les conditions qui obligent à satisfaire la demande et épuiser la disponibilité.

Dans un problème du transport, il existe une contrainte pour chaque nœud.

Les contraintes sont :

- Lorsque le bus se trouve à la station « i », il doit le quitter vers une seule destination :

  $\sum_{\substack{j=1 \\ j\neq i}}^{n} X_{ij} = 1$, $i = 1, \dots, n$

- *Le bus doit arriver à une station « j » en provenance d'un seul point :*

  $\sum_{\substack{i=1 \\ i\neq j}}^{n} X_{ij} = 1$, $j = 1, \dots, n$

- Contrainte de non négativité et d'intégrité: $X_{ij} = 0 \; ou \; 1 \;\; \forall i, j = 1, \dots, n$

3.1.4 Formulation mathématique

Le modèle de transport deviendra alors :

$$\min(z = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} C_{ij} \; X_{ij})$$
$$\begin{cases} \sum_{j=1}^{n} X_{ij} = 1 & \forall i \epsilon \{1, \dots, n\} \\ \sum_{i=1}^{n} X_{ij} = 1 & \forall j \epsilon \{1, \dots, n\} \\ X_{ij} \in \{0,1\} & \forall i, j \epsilon \{1, \dots, n\} \end{cases}$$

Il s'agit d'un programme linéaire en nombres binaire.

3.2 Formes d'un programme linéaire

3.2.1 Forme générale

$$(\max ou \min)(z = \sum_{j=1}^{n} C_j X_j) \qquad (1)$$
$$\begin{cases} \sum_{j=1}^{n} a_{ij} X_j \quad \leq, = ou \geq b_i & \forall \; i = 1, \dots, m \; (2) \\ X_j \geq 0 & \forall j = 1, \dots, n \qquad (3) \end{cases}$$

(1) : fonction objective

(2) : m contraintes linéaires

(3) : contraintes de positivité

3.2.2 Forme matricielle

Notons par $x = (x_1, \dots, x_n)^T$ le vecteur des variables, $b = (b_1, \dots, b_n)^T$ le second membre des contraintes, $c = (c_1, \dots, c_n)^T$ le vecteur coût associé aux variables et A la matrice m×n des a_{ij}

Forme Standard

$$\begin{cases} max \ (z = cx) \\ Ax = b \\ x \geq 0 \end{cases}$$

La forme standard avec des contraintes égalité s'utilise dans la résolution algébrique.

3.3 Méthode du simplexe en deux phases

L'algorithme du simplexe a été introduit par George Dantzig à partir de 1946. C'est un algorithme de résolution des problèmes d'optimisation linéaire.

Il consiste à minimiser une fonction linéaire de n variables réelles,

$$\boldsymbol{x} = (\boldsymbol{x_1}, \dots, \boldsymbol{x_n}) \in \mathbb{R} \to \boldsymbol{c^T x} = \sum_{i=1}^{n} \boldsymbol{c_i x_i}$$

Où $c = (c_1, \dots, x_n) \in \mathbb{R}$ sur un ensemble défini au moyen des contraintes affines (ou linéaires) d'égalité et d'inégalité.

- <u>**Les deux phases de la méthode de simplexe :**</u>
  - **Phase 1 (Initialisation) :** Trouver une solution de base réalisable.
  - **Phase 2 (Progression) :** On passe d'un sommet à un sommet voisin pour augmenter la fonction objectif.

<u>**Phase 1 :**</u>

PL sous forme standard :

$$(P)\begin{cases} \min(z = \sum_{i=1}^{n} c_i x_i) \\ Ax = b \\ x \geq 0 \end{cases}$$

On ne suppose pas que la matrice $A \in \boldsymbol{M}_{m*n}$ est de rang plein, ni qu'il existe bien des solutions réalisables.

Pour obtenir une solution de base réalisable, on introduit un problème de programmation linéaire auxiliaire pour des variables supplémentaires appelées variables artificielles.

Le programme auxiliaire associé à (PL) s'écrit :

$$(P_a)\begin{cases} \min(z = \sum_{i=1}^{m} t_i) \\ Ax + t = b \\ x \geq 0 \\ t \geq 0 \end{cases}$$

Où $t = (t_1, \dots, t_m)$ sont appelées variables artificielles.
Le modèle linéaire (P_a) permet, soit de conclure que le problème (P) est non réalisable, soit de trouver une solution de base réalisable de (P).

- Si la valeur optimale de (P_a) soit égale à zéro et considérons une solution de base optimale $(x_1, \dots, x_n, t_1, \dots, t_m)$ de (P_a)
  - ✓ Si $t_1, \dots, t_m \neq 0$ (P) non réalisable
  - ✓ Si $t_1, \dots, t_m = 0$ alors$(x_1, \dots, x_n)$ constitue une solution de base réalisable de base réalisable (P). On passe à la phase 2 du simplexe

- Si la valeur optimale de (P_a) est non nul alors (P) est non réalisable.

<u>Phase 2 :</u>

La phase 2 consiste à appliquer l'algorithme du simplexe au problème sous sa forme canonique obtenue à la phase 1

PL sous forme canonique :

$$\begin{cases} \min(z=\sum^{n}_{i=1} c_i x_i) \\ Ax=b \\ x \geq 0 \end{cases}$$

<u>Principe de la méthode du simplexe :</u>

Faire rentrer une variable hors-base dans la nouvelle base (variable entrante) et faire sortir à la place une variable de base (variable sortante).

<u>Algorithme du simplexe :</u>

<u>Etape 1 :</u> Ecrire le problème sous forme canonique en identifiant une solution de base initiale. Donner alors le tableau du simplexe.

<u>Etape 2 :</u> On doit choisir la colonne de pivot.

Pour cela, on choisit l'indice j tel quel : $C_j = \min\{Ci \mid Ci < 0\}$

Si aucun choix est possible, on a atteint la solution optimale et l'algorithme se termine. Sinon, sélectionner la variable hors-base ayant le plus petit coefficient négatif pour entrer la base c'est la variable d'entrée.

Etape 3 : On doit choisir la ligne de pivot.

Pour cela on choisit l'indice i en appliquant la règle du plus petit rapport pour identifier la variable qui quitte la base c'est la variable de sortie.

$$\frac{b_j}{a_{ij}} = \min\left\{\frac{b_k}{a_{kj}} / a_{kj} > 0 \;\; k = 1, \dots \boldsymbol{m}\right\}$$

Où j est la colonne de pivot de l'étape 2.

Etape 4 : On applique la procédure d'élimination de Gauss-Jordan autour du pivot situé à l'intersection de la ligne i et de la colonne j. Ensuite, on divise la ligne i par le pivot pour le mettre égal à 1. Pour avoir le nouveau tableau et la nouvelle solution de base réalisable.

Retourner à l'étape 2

Chapitre 2 : Problème de plus court chemin

Le problème de plus court chemin compte parmi les problèmes les plus classiques de la théorie des graphes et les plus importants dans leurs applications, donc on va essayer dans ce chapitre de définir en premier temps le problème de plus court chemin puis en deuxième temps, on va parler des algorithmes de résolution de ce problème et on va mettre le point sur l'algorithme de Dantzig.

1. Définition d'un problème de plus court chemin

Le problème de plus court chemin peut être posé de la façon suivante :

- étant donnés un graphe valué G=(X, U, L), on associera à chaque arc u= (i, j) un nombre réel l(u) où l_{ij} appelée la longueur de l'arc. Le problème de plus court chemin entre deux sommets « S » (source ou origine) et « d » (destination) du graphe consiste à déterminer parmi tous les chemins allant de « S » à « d » un chemin noté μ^* dont la longueur total : $l(\mu^*) = \sum_{u \in \mu^*} l(u)$ soit minimale.

<u>Condition d'existence de plus court chemin :</u>

Le problème de plus court chemin a une solution si et seulement s'il n'existe pas dans le graphe de circuit de longueur strictement négative pouvant être atteint à partir de l'origine « S ».

2. Algorithmes de résolution

2.1 Généralités

Différents problème peuvent être posés autour de la recherche de plus court chemin.

Un premier problème A consiste à rechercher le plus court chemin entre deux points donnés, c'est –à-dire déterminer le chemin de plus petit longueur qui relie ces points. Pour résoudre ce problème, on résout le problème B. Ce dernier consiste à déterminer, à partir d'un point donné le plus court chemin pour aller à tous les autres nœuds. Enfin le problème C consiste à trouver, pour n'importe quelle paire de nœuds, le chemin le plus court entre eux.

Le problème B peut être résolu de nombreuses manières. Tout dépend des hypothèses émises sur la structure du réseau. Lorsque le réseau est sans circuit, on appliquera l'algorithme de Bellman. Lorsque le réseau n'a que des longueurs positives ou nulles (mais éventuellement avec des circuits), on utilisera la méthode de Dijkstra.

Donc selon les propriétés du graphe traité (orienté/non orienté, avec/sans circuit ou longueur positives/quelconques) et selon le problème considéré (recherche de plus court chemin d'un sommet vers tous les autres/entre tous les couples de sommets) il existe nombreux algorithmes permettant l'obtention d'une solution.

Le tableau suivant présente les différents algorithmes de résolution de ce problème :

Algorithmes	Type du PCC	Propriétés du graphe	
		Type de graphe	Longueur
Dijkstra	**D'un sommet à tous les autres sommets**	**Graphe orienté (et non orienté)**	**Longueur positives**
Bellman		**Graphe orienté sans circuit (sommet d'origine doit être sans prédécesseur)**	**Longueur quelconque (nombre réel)**
Bellman-Ford		**Graphe orienté**	
Floyd	**Entre tous les couples de sommets**	**Graphe orienté sans circuit absorbant**	

Tableau 1 : Algorithmes de résolution de problème de plus court chemin

On S'intéressera ici à l'algorithme de Dantzig (moor-djikstra).

2.2 Algorithme de DANTZIG

L'algorithme de dantzig permet de calculer le plus court chemin d'un sommet « s » à un sommet « d » ou d'un sommet « s » à tous les autres sommets dans un graphe de longueur positive.

Algorithme :

Titre : Dantzig

Entrées : R=(X, U, c) un réseau

Sorties : D une matrice contenant les plus courtes distances

Variables intermédiaires : i, j, k des entiers

Début

Pour i ← 1 à n faire

Pour j ← 1 à n faire

Si i = j alors D(i, j) ← 0 ;

Sinon D(i, j) ← +∞ ;

Fin pour ;

Fin pour ;

Pour k ← 1 à n faire

Pour i ← 1 à k − 1 faire

$D(k,i) \leftarrow \min\{c(x_k,x_j)+D(j,i)/(x_k,x_j) \in U\}$; Fin pour;

Pour i ← 1 à k − 1 faire ;

$D(i, k) \leftarrow \min\{D(i, j) + c(x_j, x_k) / (x_j, x_k)\} \in U\}$;

Fin pour ;

Pour i ← 1 à k − 1 faire Pour j

← 1 à k − 1 faire

$D(i, j) \leftarrow \min\{D(i, j), D(i, k) + D(k, j)\}$;

Fin pour ;

Fin pour ;

Fin pour ;

Fin

Chapitre 3 : Problème du flot maximal

Le problème du flot maximal consiste à trouver, dans un réseau de transport, un flot maximal depuis une source unique et vers un puits unique. Donc on va définir d'une part le problème de flot maximal et d'autre part son algorithme de résolution.

1. Réseau de transport

1.1 Flots dans un réseau de transport

1.1.1 Définition d'un réseau de transport

Un réseau de transport est un graphe sans boucle où chaque arc est associé à un nombre $c(u) > 0$, appelé « capacité de l'arc u ».

En outre, un tel réseau vérifie les hypothèses suivantes :

- Il existe un seul nœud « E » qui n'a pas de prédécesseur, tous les autres ont au moins un prédécesseur. Ce nœud est appelé l'entrée (ou la source) du réseau.
- Il existe également un seul nœud « S » qui n'a pas de successeur, tous les autres ont au moins un successeur. Ce nœud est appelé la sortie (ou le puits) du réseau.

<u>EXEMPLE :</u>

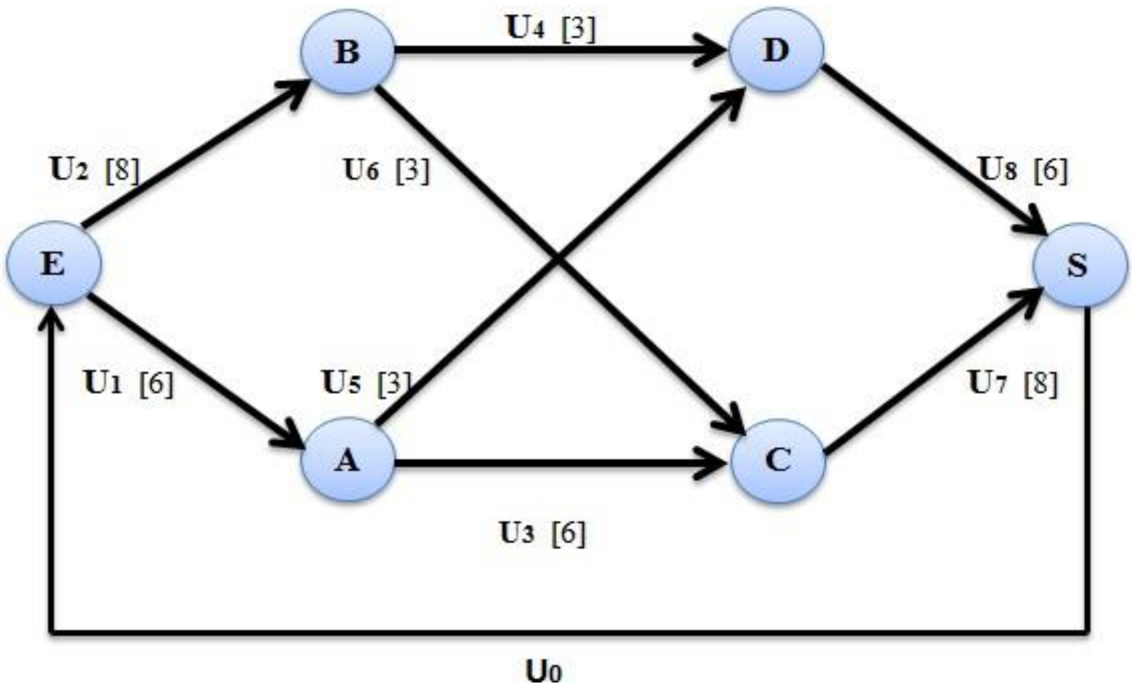


Figure 6 : Réseau de transport avec arc retour

1.1.2 Définition d'un flot

Un flot φ dans un réseau de transport est une fonction qui associe à chaque arc u, $\varphi(u)$ représente la quantité de flot qui passe par cet arc u.

Un flot est vérifié les deux propriétés suivantes :

- $\forall$ u∈U, $0 \leq \varphi(u) \leq c(u)$
- $\forall$ x ∈ X, $\sum_{u \in U^{+}(x)} \varphi(u) = \sum_{u \in U^{-}(x)} \varphi(u)$ « loi de conservation du flot »
  $avec\ U^{-}(x) = \{u \epsilon U / u(y, x), y \epsilon X\}$ Et $U^{+}(x) = \{u \epsilon U / u(x, y), y \epsilon X\}$

1.1.3 Valeur du flot

On ajoute un arc de retour « fictif » depuis « S » vers « E », on définit alors la valeur du flot comme celle du flux qui par cet arc fictif.

1.1.4 Flot compatible

Un flot est compatible si pour tout arc u∈U $0 \leq \varphi(u) \leq c(u)$.Autrement dit, pour chaque arc, le flot qui le traverse ne doit pas dépasser la capacité de l'arc.

1.1.5 Flot complet

Un flot complet est un flot compatible pour lequel tout chemin allant de « E » à « S » possède au moins un arc saturé.

1.1.6 Coupe

On appelle coupe séparant E et S un ensemble d'arc de la forme $\omega^{-}(A)$ ou $A \subset X$ tel que $S \in A$ et $E \notin A$

1.1.7 Capacité d'une coupe

La capacité d'une coupe est la somme des capacités des arcs da la coupe C(A)=$\sum_{u \in \omega^{-}_{A}} C(u)$.

1.2 Saturation d'un réseau de transport :

1.2.1 Arc saturé

Un arc est saturé par un flot φ si la valeur du flot sur l'arc égale à sa capacité

$ie\ \varphi(x, y) = c(x, y)$.

1.2.2 Chemin saturé

Un chemin est saturé si l'un de ses arcs est saturé.

2. Algorithme de résolution

L'algorithme le plus connu pour résoudre ce problème est celui de Ford-Fulkerson.

2.1 Algorithme de flot compatible :

Soit φ un flot quelconque :

1- $d(\varphi) = \sum_{u\epsilon U} d(\varphi_u)$

- Si $d(\varphi) = 0 \rightarrow \varphi$ *est un flot compatible*
- Si $d(\varphi) > 0$ *soit* $u_0 = (x, y) / u_0 \epsilon U\ d(\varphi_{u_0}) > 0$

2- Marquer y

3- Soit u= (i, j), si le sommet i est marqué et le sommet j non marqué,

Marqué le sommet j si :

- Soit il existe un arc u= (i, j) tel que $\varphi_u \leq b_u < C_u$ $(u \neq u_0)$
- Soit il existe un arc u= (i, j) ou u= (j, i) tel que $b_u < \varphi_u < C_u$
- Soit il existe un arc u= (j, i) tel que $b_u < C_u < \varphi_u$

4- Si x est marqué alors il existe un cycle μ passant par l'arc u_0 tel que $(\vec{\mu})_{u_0} = 1$

$$Soit\ \alpha = \min(\alpha_1, \alpha_2), \alpha > 0 \text{ avec } \begin{cases} \alpha_1 = min_{u\epsilon\mu^+}(C_u - \varphi_u) \\ \alpha_2 = min_{u\epsilon\mu^-}(\varphi_u - b_u) \end{cases}$$

Poser $\varphi' = \varphi + \alpha\mu$ *et revenir à l'étape* 1

Si x n'est pas marquer, STOP pas de flot compatible.

<u>Théorème de ford-Fulkerson :</u>

Dans un réseau de transport, la valeur maximale d'un flot est égale à la capacité minimale d'une coupe.

2.2 Algorithme de ford-Fulkerson :

L'algorithme de Ford-Fulkerson se décompose en deux étapes :

- Procédure de marquage : permet de déterminer s'il existe un chemin de E « source » à S « puits »
- Procédure d'augmentation de flot : permet d'augmenter le flot actuel

La procédure d'augmentation ne s'applique que si « S » est marqué à l'issue de la procédure de marquage.

- Soit $E \to \cdots \to x_i \to x_{i+1} \to \cdots \to x_j \leftarrow x_{j+1} \to \cdots \to S$ une chaine de « E » à « S »
- Soit μ^+l'ensemble des arcs de la chaine de type (x_i, x_{i+1}) « arc avant »
- Soit μ^-l'ensemble des arcs de la chaine de type $(x_j,\ x_{j+1})$ « arc arrière »
- Les arcs de μ^+ sont tels qu'il est possible d'augmenter la valeur du flux associe d'une quantité $\alpha^+ \geq 0$
- Les arcs de μ^- sont tels qu'il est possible de diminuer la valeur du flot associe d'une quantité $\alpha^- \geq 0$
- Une chaine est dite améliorant si $\alpha^+ > 0\ et\ \alpha^- > 0$

Algorithme : Recherche d'un flot maximal dans un réseau de transport

1) Partir d'un flot compatible, soit φ_0 sa valeur.
2) Appliquer la procédure du marquage suivant :
   a. Marquer la source « E » du réseau
   b. Soit (i, j) un arc de réseau, si le sommet i est marqué, alors marqué j si :
      * Soit il existe un arc u= (i, j) tel que $\varphi_u < C_u$
      * Soit il existe un arc u= (j, i) tel que $\varphi_u > 0$
3) Si l'on n'arrive pas à marquer le puits « S » du réseau, alors le flot est maximal, FIN
   Si l'on marquer le puits « S » du réseau, ce flot n'est pas maximal, il faut passer à l'étape (4)
4) Amélioration de la valeur du flot φ :
   Choisir une chaine de sommets marqués entre « E » et « S », cette chaine et l'arc u_0 forment un cycle tel que $\overrightarrow{(\mu}_{u_0} = 1$
   Posons $\varphi' = \varphi + \alpha\mu$ avec $\alpha = \min(\alpha_1, \alpha_2)$
   Et $\begin{cases} \alpha_1 = \min\limits_{u\epsilon\mu^+}(C_u - \varphi_u) \\ \alpha_2 = \min\limits_{u\epsilon\mu^-}(\varphi_u) \end{cases}$

Chapitre 4 : Résolution du problème de transport urbain

1. Problématique

Les habitants de la ville de Fès et plus précisément les étudiants universitaires trouvent des difficultés en ce qui concerne les différentes lignes existantes ainsi que les chemins qui relient les différentes stations de cette ville, pour cette raison, on est amené à résoudre ce problème c'est-à-dire :

- Déterminer le plus court chemin entre une station de départ et une autre d'arrivée, et cela pour minimiser le temps.
- Déterminer le flot maximal dans un réseau routier de transport de Fès, c'est-à-dire la saturation du réseau pour éviter tout embouteillage susceptible.

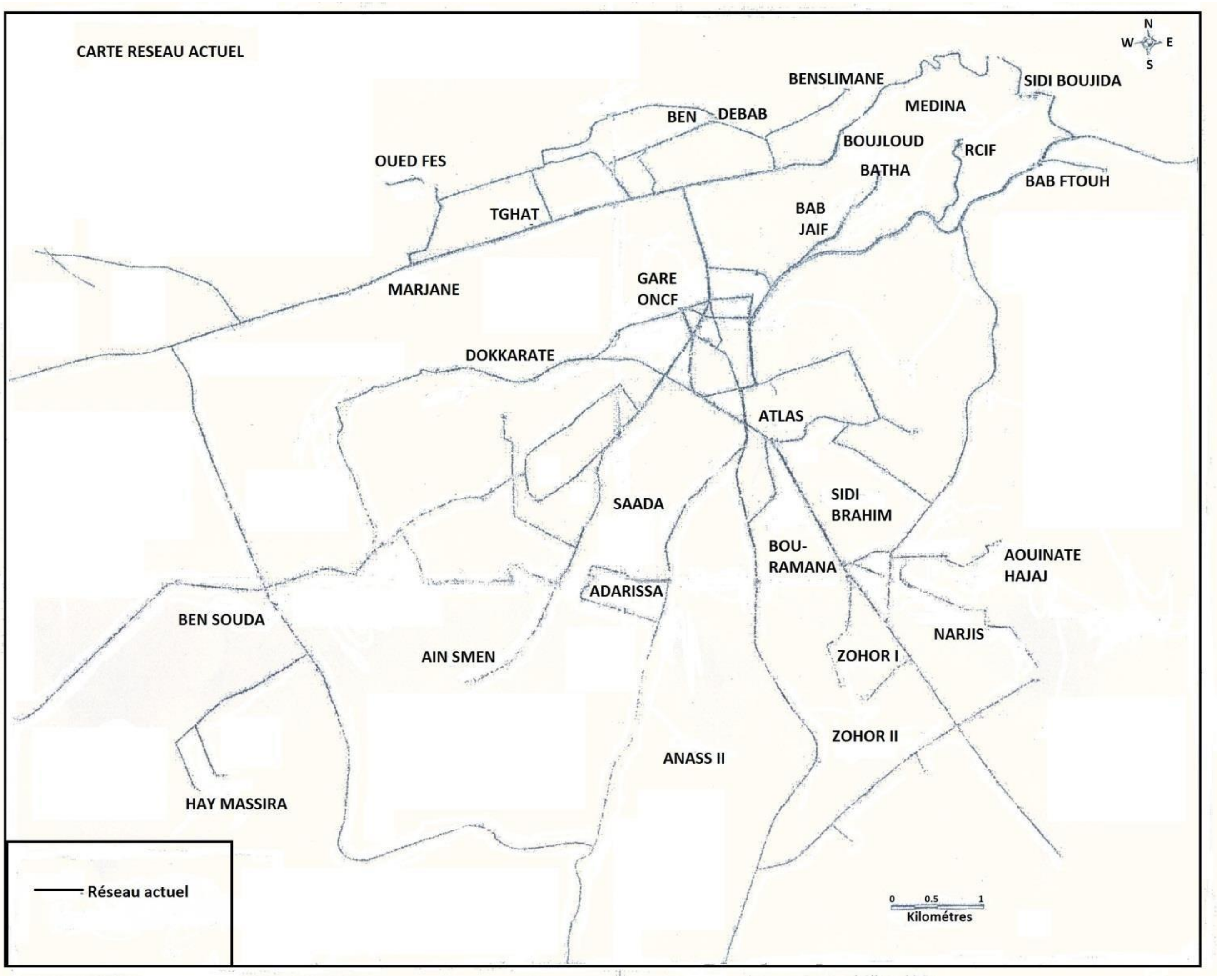


Figure 7 : Carte réseau du transport urbain collectif à Fès

2. Problème de plus court chemin dans un réseau de transport urbain

A partir de la carte réseau du transport urbain collectif à Fès, on a modélisé notre problème sous forme d'un graphe orienté pour lier les stations entre eux. Sachant que Les stations, représentent les sommets et chaque ligne liant deux sommet, représente un arc (chaque arc à une distance).

On va chercher le plus court chemin dans le graphe, en utilisant l'algorithme de Dantzig.

Choisissons la station de départ (Gare O.N.C.F) et la station d'arrivée (Bab Ftouh).

- **<u>Représentation graphique</u> :**

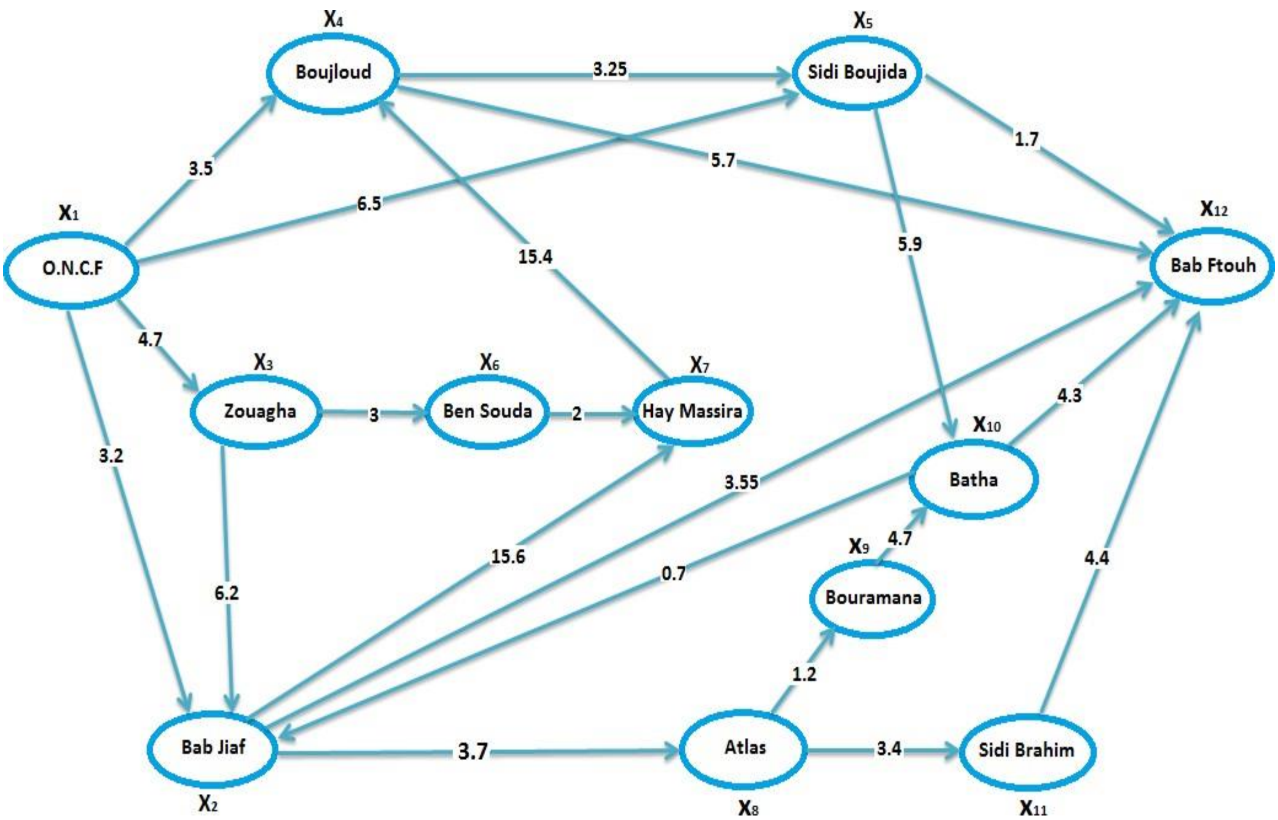


Figure 8 : représentation graphique de Carte réseau du transport urbain collectif à Fès

2.1 Simulation de l'algorithme de DANTZIG

Itération 1 :

- Numéroter les sommets dans un ordre quelconque (voir figure précèdent)
- On pose $\lambda(x_1) = 0 \; et \; A_1 = \{x_1\}$
- $x_1 \in A_1$, donc au sommet x_1 on doit déterminer un sommet x^*_1 tel sorte que :

$$l(x_1, x^*_1) = \min_{\substack{x_1 \in A_1 \\ y_j \notin A_1}} l(x_1, y_j) = min(l(x_1, x_2), l(x_1, x_3), l(x_1, x_4), l(x_1, x_5))$$

$$= min\ (3.2 ; 4.7 ; 3.5 ; 6.5) = 3.2$$

$$Donc\ x_1 \rightarrow x_1^* = x_2$$

- On doit déterminer un couple de sommet (x_q, y_q) tel que :

$$\lambda(x_q) + l(x_q, y_q) = \min_{\substack{x_i \in A_1 \\ y_i \notin A_1}} (\lambda(x_i) + l(x_i, y_i)) = min\ (\lambda(x_1) + l(x_1, x_2)) = 3.2$$

$$\boxed{Donc \quad \lambda(x_2) = \lambda(x_1) + l(x_1, x_2) = 3.2}$$

Itération 2 :

- Soit $\lambda(x_1) = 0$, $\lambda(x_2) = 3.2 \; et \; A_2 = \{x_1, x_2\}$
- On a $x_1, x_2 \epsilon A_2$ donc à chaque sommet $x_i \epsilon A_i$ on doit déterminer un sommet x^*_i tel sorte que : $l(x_i, x^*_i) = \min_{\substack{xi \epsilon A2 \\ yj \notin A2}} l(x_i, y_j)$
  - ✓ Pour le sommet $x_1 \epsilon A_2$ $l(x_1, x^*_1) = min(l(x_1, x_3), l(x_1, x_4), l(x_1, x_5))$

$$= min\ (4.7 ; 3.5 ; 6.5) = 3.5$$

$$Donc\ x_1 \rightarrow x_1^* = x_4$$

  - ✓ Pour le sommet $x_2 \epsilon A_2$ $l(x_2, x^*_2) = min(l(x_2, x_7), l(x_2, x_8), l(x_2, x_{12}))$

$$= min\ (15.6 ; 3.7 ; 3.55) = 3.55$$

$$Donc\ x_2 \rightarrow x_2^* = x_{12}$$

- On doit déterminer un couple de sommet (x_q, y_q) tel que :

$$\lambda(x_q) + l(x_q, y_q) = min(\lambda(x_1) + l(x_1, x_4), \lambda(x_2) + l(x_2, x_{12})) = min(3.5 ; 6.75) = 3.5$$

$$\boxed{Donc \quad \lambda(x_4) = (\lambda(x_1) + l(x_1, x_4)) = 3.5}$$

Itération 3 :

- Soit $\lambda(x_1) = 0$, $\lambda(x_2) = 3.2$, $\lambda(x_4) = 3.5$ *et* $A_3 = \{x_1, x_2, x_4\}$
- On a x_1, x_2, $x_4 \epsilon\ A_3$ donc à chaque sommet $x_i \epsilon A_i$ on doit déterminer un sommet x_i^* tel sorte que : $l(x_i, x_i^*) = \min_{\substack{xi\epsilon A3 \\ yj\notin A3}} l(x_i, y_j)$

  ✓ Pour le sommet $x_1 \epsilon A_3$ $l(x_1, x_1^*) = min(l(x_1, x_3), l(x_1, x_5))$

  $= min\ (4.7\ ;\ 6.5) = 4.7$

  Donc $x_1 \to x_1^* = x_3$

  ✓ Pour le sommet $x_2 \epsilon A_3$ *on a :* $x_2 \to x_2^* = x_{12}$

  ✓ Pour le sommet $x_4 \epsilon A_3$ $l(x_4, x_4^*) = min(l(x_4, x_5), l(x_4, x_{12}))$

  $= min\ (3.25\ ;\ 5.7) = 3.25$

  Donc $x_4 \to x_4^* = x_5$

- On doit déterminer un couple de sommet (x_q, y_q) tel que :

$$\lambda(x_q) + l(x_q, y_q) = min(\lambda(x_1) + l(x_1, x_3), \lambda(x_2) + l(x_2, x_{12}), \lambda(x_4) + l(x_4, x_5))$$

$$= min\ (4.7\ ;\ 6.75\ ;\ 6.75) = 4.7$$

$$\boxed{Donc\ \ \lambda(x_3) = (\lambda(x_1) + l(x_1, x_3)) = 4.7}$$

Itération 4 :

- Soit $\lambda(x_1) = 0$, $\lambda(x_2) = 3.2$, $\lambda(x_3) = 4.7$ $\lambda(x_4) = 3.5$ *et* $A_4 = \{x_1, x_2, x_3, x_4\}$
- On a $x_1, x_2, x_3, x_4 \epsilon\ A_4$ donc à chaque sommet $x_i \epsilon A_i$ on doit déterminer un sommet x_i^* tel sorte que : $l(x_i, x_i^*) = \min_{\substack{xi\epsilon A4 \\ yj\notin A4}} l(x_i, y_j)$

  ✓ Pour le sommet $x_1 \epsilon A_4$ $l(x_1, x_1^*) = min(l(x_1, x_5))$

  $= min\ (6.5) = 6.5$

  Donc $x_1 \to x_1^* = x_5$

  ✓ Pour le sommet $x_2 \epsilon A_4$ on a : $x_2 \to x_2^* = x_{12}$

  ✓ Pour le sommet $x_4 \epsilon A_4$ on a $x_4 \to x_4^* = x_5$

  ✓ Pour le sommet $x_3 \epsilon A_4$ $l(x_3, x_3^*) = min(l(x_3, x_2), l(x_3, x_6))$

  $= min\ (6.2\ ;\ 3) = 3$

  Donc $x_3 \to x_3^* = x_6$

- On doit déterminer un couple de sommet (x_q, y_q) tel que :

$$\lambda(x_q) + l(x_q, y_q) = min(\lambda(x_1) + l(x_1, x_5), \lambda(x_2) + l(x_2, x_{12}), \lambda(x_4) + l(x_4, x_5),$$
$$\lambda(x_3) + l(x_3, x_6)) = min\ (6.5\ ;\ 6.75\ ;\ 6.75\ ;\ 7.7) = 6.5$$

$$\boxed{\text{Donc}\ \ \lambda(x_5) = (\lambda(x_1) + l(x_1, x_5)) = 6.5}$$

Itération 5 :

- Soit $\lambda(x_1) = 0$, $\lambda(x_2) = 3.2$, $\lambda(x_3) = 4.7$, $\lambda(x_4) = 3.5$, $\lambda(x_5) = 6.5$

  $A_5 = \{x_1, x_2, x_3, x_4, x_5\}$

- On a $x_1, x_2, x_3, x_4, x_5 \epsilon A_5$ donc à chaque sommet $x_i \epsilon A_i$ on doit déterminer un sommet x_i^* tel sorte que : $l(x_i, x_i^*) = min_{x_i \epsilon A_5, y_j \notin A_5} l(x_i, y_j)$

  - ✓ Pour le sommet $x_2 \epsilon A_4$ on a : $x_2 \to x_2^* = x_{12}$
  - ✓ Pour le sommet $x_4 \epsilon A_5$ *on a* $x_4 \to x_4^* = x_{12}$
  - ✓ Pour le sommet $x_3 \epsilon A_5$ *on a* $x_3 \to x_3^* = x_6$
  - ✓ Pour le sommet $x_5 \epsilon A_5$ $l(x_5, x_5^*) = min(l(x_5, x_9), l(x_5, x_{12}))$

    $= min\ (5.9 ; 1.7) = 1.7$

    Donc $x_5 \to x_5^* = x_{12}$

- On doit déterminer un couple de sommet (x_q, y_q) tel que :

  $\lambda(x_q) + l(x_q, y_q) = min(\lambda(x_2) + l(x_2, x_{12}), \lambda(x_4) + l(x_4, x_{12}),$

  $\lambda(x_3) + l(x_3, x_6), \lambda(x_5) + l(x_5, x_{12})) = min\ (6.75 ; 9.2 ; 7.7 ; 8.2) = 6.75$

$$\boxed{Donc\ \ \lambda(x_{12}) = (\lambda(x_2) + l(x_2, x_{12})) = 6.75}$$

Résultat :

L'algorithme de Dantzig donne comme chemin minimum :

Gare (O.N.C.F) →Bab Jiaf→Bab Ftouh

La longueur de ce chemin est : 6.75 Km

2.2 Programmation en langage C

➢ **Numérotation :**

0 → Gare(O.N.C.F)	**6** → Hay Massira
1 → Bab Jiaff	**7** → Atlas
2 → Zouagha	**8** → Bouramana
3 → Boujloud	**9** → Batha **11**
4 → Sidi Boujida	**10** → Sidi Brahim
5 → Ben Souda	**11** → Bab Ftouh

➢ L’exécution de programme :

D:\PFE\Nouveau dossier\Dantzig.exe

```
La matrice adjacence est
0       1       1       1       1       0       0       0       0       0
0       0
0       0       0       0       0       0       1       1       0       0
0       1
0       1       0       0       0       1       0       0       0       0
0       0
0       0       0       0       1       0       0       0       0       0
0       1
0       0       0       0       0       0       0       0       0       1
0       1
0       0       0       0       0       0       1       0       0       0
0       0
0       0       0       1       0       0       0       0       0       0
0       0
0       0       0       0       0       0       0       0       1       0
1       0
0       0       0       0       0       0       0       0       0       1
0       0
0       1       0       0       0       0       0       0       0       0
0       1
0       0       0       0       0       0       0       0       0       0
0       1
0       0       0       0       0       0       0       0       0       0
0       0
```

```
La matrice des couts est
        0       1       2       3       4       5       6       7       8
9       10      11
0||     -       3.20    4.70    3.50    6.50    -       -       -       -
-       -       -
1||     -       -       -       -       -       -       15.60   3.70    -
-       -       3.55
2||     -       6.20    -       -       -       3.00    -       -       -
-       -       -
3||     -       -       -       -       3.25    -       -       -       -
-       -       5.70
4||     -       -       -       -       -       -       -       -       -
5.90    -       1.70
5||     -       -       -       -       -       -       2.00    -       -
-       -       -
6||     -       -       -       15.40   -       -       -       -       -
-       -       -
7||     -       -       -       -       -       -       -       -       1.20
-       3.40    -
8||     -       -       -       -       -       -       -       -       -
4.70    -       -
9||     -       0.70    -       -       -       -       -       -       -
-       -       4.30
10||    -       -       -       -       -       -       -       -       -
-       -       4.40
11||    -       -       -       -       -       -       -       -       -
-       -       -
 le chemin le plus court est
  0 ->  1 -> 11
 la longueur de ce chemin est 6.75
Appuyez sur une touche pour continuer...
```

L’algorithme de dantzig donne comme chemin minimum :

Gare (O.N.C.F) →Bab Jiaf→Bab Ftouh

La longueur de ce chemin est : 6.75 Km

2.3 Modélisation d'un problème de plus court chemin

Il faut noter que le problème de recherche de plus court peut résolu à travers les notions de programmation linéaire. Ainsi le problème peut être modélisé sous forme d'un problème de programmation linéaire.

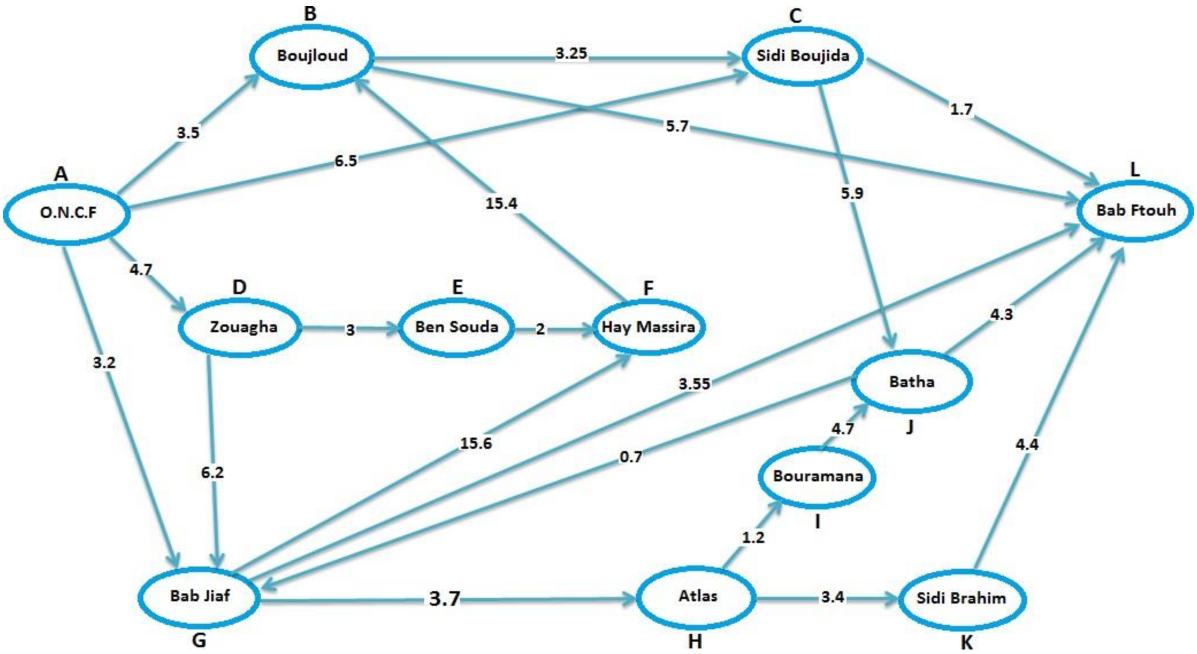


Figure 9 : représentation graphique de Carte réseau du transport urbain collectif à Fès

- **Variables:**

x_{ij}: Le déplacement de la station « i » au « j »

- **Contraintes :**

-Bilan des trajets de la station A : $x_{AB} + x_{AC} + x_{AD} + x_{AG} = 1$

-Bilan des trajets de la station B : $x_{BC} + x_{BL} - x_{AB} - x_{FB} = 0$

-Bilan des trajets de la station C : $x_{CJ} + x_{CL} - x_{AC} - x_{BC} = 0$

-Bilan des trajets de la station D : $x_{DE} + x_{DG} - x_{AD} = 0$

-Bilan des trajets de la station E : $x_{EF} - x_{DE} = 0$

-Bilan des trajets de la station F : $x_{FB} - x_{EF} - x_{GF} = 0$

-Bilan des trajets de la station G : $x_{GF} + x_{GH} + x_{GL} - x_{AG} - x_{DG} - x_{JG} = 0$

-Bilan des trajets de la station H : $x_{HI} + x_{HK} - x_{GH} = 0$

-Bilan des trajets de la station I : $x_{IJ} - x_{HI} = 0$

-Bilan des trajets de la station J : $x_{JG} + x_{JL} - x_{CJ} - x_{IJ} = 0$

-Bilan des trajets de la station K : $x_{KL} - x_{HK} = 0$

-Bilan des trajets de la station L : $-x_{BL} - x_{CL} - x_{GL} - x_{KL} - x_{JL} = -1$

-$x_{ij} \geq 0$

-$x_{ij} \in \{0,1\}$

- **Fonction objectif :**

$$\min z = 3.5x_{AB} + 6.5x_{AC} + 4.7x_{AD} + 3.2x_{AG} + 3.25x_{BC} + 5.7x_{BL} + 5.9x_{CJ} + 1.7x_{CL} + 3x_{DE} + 6.2x_{DG} + 2x_{EF} + 15.4x_{FB} + 15.6x_{GF} + 3.7x_{GH} + 3.55x_{GL} + 1.2x_{HI} + 3.4x_{HK} + 4.7x_{IJ} + 0.7x_{JG} + 4.3x_{JL} + 4.4x_{KL}$$

On effectue les changements suivants :

x_{AB}	x_{AC}	x_{AD}	x_{AG}	x_{BC}	x_{BL}	x_{CJ}	x_{CL}	x_{DE}	x_{DG}	x_{EF}	x_{FB}	x_{GF}	x_{GH}	x_{GL}	x_{HI}	x_{HK}	x_{IJ}	x_{JG}	x_{JL}	x_{KL}
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}

Le modèle mathématique se résume ainsi

$$\textbf{(P)} \quad \begin{cases} \min z = 3.5x_1 + 6.5x_2 + 4.7x_3 + 3.2x_4 + 3.25x_5 + 5.7x_6 + 5.9x_7 + 1.7x_8 + 3x_9 + 6.2x_{10} + 2x_{11} + 15.4x_{12} \\ \qquad + 15.6x_{13} + 3.7x_{14} + 3.55x_{15} + 1.2x_{16} + 3.4x_{17} + 4.7x_{18} + 0.7x_{19} + 4.33x_{20} + 4.4x_{21} \\ \boldsymbol{sc} \quad x_1 + x_2 + x_3 + x_4 = 1 \\ \quad -x_1 + x_5 + x_6 - x_{12} = 0 \\ \quad -x_1 - x_5 + x_7 + x_8 = 0 \\ \quad -x_3 + x_9 + x_{10} = 0 \\ \quad -x_9 + x_{11} = 0 \\ \quad -x_{11} + x_{12} - x_{13} = 0 \\ \quad -x_4 - x_{10} + x_{13} + x_{14} + x_{15} - x_{19} = 0 \\ \quad -x_{14} + x_{16} + x_{17} = 0 \\ \quad -x_{16} + x_{18} = 0 \\ \quad -x_7 - x_{18} + x_{19} + x_{20} = 0 \\ \quad -x_{17} + x_{21} = 0 \\ \quad -x_6 - x_8 - x_{15} - x_{20} - x_{21} - x_{22} = -1 \\ \quad x_i \geq 0 \ \forall i = 1, \ldots, 21 \\ \quad x_i \in \{0,1\} \ 1 \leq i \leq 21 \end{cases}$$

2.4 Simulation de la méthode du simplexe en deux phases

La contrainte 12 est de type (=), et $b_{12} < 0$ donc on multiple la contrainte 12 par -1, il est nécessaire d'ajouter la variable artificielle x_{22}

Les contraintes 1,...,11 sont de type (=) ils sont nécessaire d'ajouter les variable artificielle x_{23}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}, x_{31}, x_{32}; x_{33}

$$
\mathbf{P(a)} \begin{cases}
\min z = 3.5x_1 + 6.5x_2 + 4.7x_3 + 3.2x_4 + 3.25x_5 + 5.7x_6 + 5.9x_7 + 1.7x_8 + 3x_9 + 6.2x_{10} + 2x_{11} + 15.4x_{12} \\
\quad + 15.6x_{13} + 3.7x_{14} + 3.55x_{15} + 1.2x_{16} + 3.4x_{17} + 4.7x_{18} + 0.7x_{19} + 4.33x_{20} + 4.4x_{21} \\
SC \quad x_1 + x_2 + x_3 + x_4 + x_{33} = 1 \\
-x_1 + x_5 + x_6 - x_{12} + x_{32} = 0 \\
-x_2 - x_5 + x_7 + x_8 + x_{31} = 0 \\
-x_3 + x_9 + x_{10} + x_{30} = 0 \\
-x_9 + x_{11} + x_{29} = 0 \\
-x_{11} + x_{12} - x_{13} + x_{28} = 0 \\
-x_4 - x_{10} + x_{13} + x_{14} + x_{15} - x_{19} + x_{27} = 0 \\
-x_{14} + x_{16} + x_{17} + x_{26} = 0 \\
-x_{16} + x_{18} + x_{25} = 0 \\
-x_7 - x_{18} + x_{19} + x_{20} + x_{24} = 0 \\
-x_{17} + x_{21} + x_{23} = 0 \\
x_6 + x_8 + x_{15} + x_{20} + x_{21} + x_{22} = 1 \\
x_i \geq 0 \ \forall i = 1, \dots, 33 \\
x_i \in \{0,1\} \ 1 \leq i \leq 33
\end{cases}
$$

Tableau 1			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**	**P_{22}**	**P_{23}**	**P_{24}**	**P_{25}**	**P_{26}**	**P_{27}**	**P_{28}**	**P_{29}**	**P_{30}**	**P_{31}**	**P_{32}**	**P_{33}**
P_{33}	-1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
P_{32}	-1	0	1	0	0	0	-1	-1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P_{31}	-1	0	1	1	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P_{30}	-1	0	0	0	1	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P_{29}	-1	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P_{28}	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_{27}	-1	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P_{26}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{25}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_{24}	-1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{23}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_{22}	-1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Z		-2	-3	-2	-2	-2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

On a $C_s < 0$ *donc la base n'est pas optimale. On prent* $C_1 = -3$ *donc* P_1 *entre en base. En appliquant la régle de plus petit rapport* $\tilde{x} = \min\{\frac{1}{1}, \frac{0}{1}, \frac{0}{1}\} = 0$ *donc* P_{31} *quitter la base et 1 joue le role de pivot. On a donc le tableau 2 suivant.*

Tableau 2			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**	**P_{22}**	**P_{23}**	**P_{24}**	**P_{25}**	**P_{26}**	**P_{27}**	**P_{28}**	**P_{29}**	**P_{30}**	**P_{31}**	**P_{32}**	**P_{33}**
P_{33}	-1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1
P_{32}	-1	0	0	-1	0	0	-1	-1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0
P_1	0	0	1	1	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
P_{30}	-1	0	0	0	1	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P_{29}	-1	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P_{28}	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_{27}	-1	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P_{26}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{25}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_{24}	-1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{23}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_{22}	-1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Z		-2	0	1	-2	-2	1	0	-3	-3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0	0

La variable qui sort de la base est P_{24}, et celle qui entre est P_7

Tableau 3			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**	**P_{22}**	**P_{23}**	**P_{24}**	**P_{25}**	**P_{26}**	**P_{27}**	**P_{28}**	**P_{29}**	**P_{30}**	**P_{31}**	**P_{32}**	**P_{33}**
P_{33}	-1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	1	1	0	0	0	-1	0	0	0	0	0	0	-1	0	1
P_{32}	-1	0	0	-1	0	0	-1	-1	0	1	0	0	0	1	0	0	0	0	0	-1	1	1	0	0	0	-1	0	0	0	0	0	0	-1	1	0
P_1	0	0	1	1	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	1	0	0
P_{30}	-1	0	0	0	1	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P_{29}	-1	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P_{28}	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_{27}	-1	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P_{26}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{25}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{23}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_{22}	-1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Z		-2	0	1	-2	-2	1	0	0	-3	0	0	0	0	0	0	0	0	0	3	-3	-2	0	0	0	3	0	0	0	0	0	0	3	0	0

La variable qui sort de la base est P_{32}, et celle qui entre est P_8

Tableau 4			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**	**P_{22}**	**P_{23}**	**P_{24}**	**P_{25}**	**P_{26}**	**P_{27}**	**P_{28}**	**P_{29}**	**P_{30}**	**P_{31}**	**P_{32}**	**P_{33}**
P_{33}	-1	1	0	1	1	1	1	1	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1
P_8	0	0	0	-1	0	0	-1	-1	0	1	0	0	0	1	0	0	0	0	0	-1	1	1	0	0	0	-1	0	0	0	0	0	0	-1	1	0
P_1	0	0	1	0	0	0	-1	-1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
P_{30}	-1	0	0	0	1	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P_{29}	-1	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P_{28}	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_{27}	-1	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P_{26}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{25}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{23}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_{22}	-1	1	0	1	0	0	1	2	0	0	0	0	0	-1	0	0	1	0	0	1	-1	-1	1	1	0	1	0	0	0	0	0	0	1	-1	0
Z		-2	0	-2	-2	-2	-2	-3	0	0	0	0	0	3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0

La variable qui sort de la base est P_{22}, et celle qui entre est P_6

Tableau 5			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**	**P_{22}**	**P_{23}**	**P_{24}**	**P_{25}**	**P_{26}**	**P_{27}**	**P_{28}**	**P_{29}**	**P_{30}**	**P_{31}**	**P_{32}**	**P_{33}**
P_{33}	-1	0.5	0	0.5	1	1	0.5	0	0	0	0	0	0	-0.5	0	0	-0.5	0	0	-0.5	0.5	0.5	-0.5	-0.5	0	-0.5	0	0	0	0	0	0	-0.5	-0.5	1
P_8	0	0.5	0	-0.5	0	0	-0.5	0	0	1	0	0	0	0.5	0	0	0.5	0	0	-0.5	0.5	0.5	0.5	0.5	0	-0.5	0	0	0	0	0	0	-0.5	0.5	0
P_1	0	0.5	1	0.5	0	0	-0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	0.5	0
P_{30}	-1	0	0	0	1	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P_{29}	-1	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P_{28}	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_{27}	-1	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P_{26}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{25}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{23}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_6	0	0.5	0	0.5	0	0	0.5	1	0	0	0	0	0	-0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	-0.5	0
Z		-0.5	0	-0.5	-2	-2	-0.5	0	0	0	0	0	0	1.5	0	0	1.5	0	0	1.5	-1.5	-0.5	1.5	1.5	0	1.5	0	0	0	0	0	0	1.5	1.5	0

La variable qui sort de la base est P_{30}, et celle qui entre est P_3

Tableau 6			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**	**P_{22}**	**P_{23}**	**P_{24}**	**P_{25}**	**P_{26}**	**P_{27}**	**P_{28}**	**P_{29}**	**P_{30}**	**P_{31}**	**P_{32}**	**P_{33}**
P_{33}	-1	0.5	0	0.5	0	1	0.5	0	0	0	1	1	0	-0.5	0	0	-0.5	0	0	-0.5	0.5	0.5	-0.5	-0.5	0	-0.5	0	0	0	0	0	-1	-0.5	-0.5	1
P_8	0	0.5	0	-0.5	0	0	-0.5	0	0	1	0	0	0	0.5	0	0	0.5	0	0	-0.5	0.5	0.5	0.5	0.5	0	-0.5	0	0	0	0	0	0	-0.5	0.5	0
P_1	0	0.5	1	0.5	0	0	-0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	0.5	0
P_3	0	0	0	0	1	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P_{29}	-1	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P_{28}	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_{27}	-1	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P_{26}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{25}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{23}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_6	0	0.5	0	0.5	0	0	0.5	1	0	0	0	0	0	-0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	-0.5	0
Z		-0.5	0	-0.5	0	-2	-0.5	0	0	0	-2	-2	0	1.5	0	0	1.5	0	0	1.5	-1.5	-0.5	1.5	1.5	0	1.5	0	0	0	0	0	2	1.5	1.5	0

La variable qui sort de la base est P_{27}, et celle qui entre est P_4

Tableau 7			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**Cb**	**P0**	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**	**P18**	**P19**	**P20**	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**	**P28**	**P29**	**P30**	**P31**	**P32**	**P33**
P33	-1	0.5	0	0.5	0	0	0.5	0	0	0	1	0	0	-0.5	1	1	0.5	0	0	-0.5	-0.5	0.5	-0.5	-0.5	0	-0.5	0	0	-1	0	0	-1	-0.5	-0.5	1
P8	0	0.5	0	-0.5	0	0	-0.5	0	0	1	0	0	0	0.5	0	0	0.5	0	0	-0.5	0.5	0.5	0.5	0.5	0	-0.5	0	0	0	0	0	0	-0.5	0.5	0
P1	0	0.5	1	0.5	0	0	-0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	0.5	0
P3	0	0	0	0	1	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
P29	-1	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P28	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P26	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P25	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P23	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P6	0	0.5	0	0.5	0	0	0.5	1	0	0	0	0	0	-0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	-0.5	0
Z		-0.5	0	-0.5	0	0	-0.5	0	0	0	-2	0	0	1.5	-2	-2	-0.5	0	0	1.5	0.5	-0.5	1.5	1.5	0	1.5	0	0	2	0	0	2	1.5	1.5	0

La variable qui sort de la base est P_{29}, et celle qui entre est P_9

Tableau 8			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**Cb**	**P0**	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**	**P18**	**P19**	**P20**	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**	**P28**	**P29**	**P30**	**P31**	**P32**	**P33**
P33	-1	0.5	0	0.5	0	0	0.5	0	0	0	0	0	1	-0.5	1	1	0.5	0	0	-0.5	-0.5	0.5	-0.5	-0.5	0	-0.5	0	0	-1	0	-1	-1	-0.5	-0.5	1
P8	0	0.5	0	-0.5	0	0	-0.5	0	0	1	0	0	0	0.5	0	0	0.5	0	0	-0.5	0.5	0.5	0.5	0.5	0	-0.5	0	0	0	0	0	0	-0.5	0.5	0
P1	0	0.5	1	0.5	0	0	-0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	0.5	0
P3	0	0	0	0	1	0	0	0	0	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
P9	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
P28	-1	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P26	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P25	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P23	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P6	0	0.5	0	0.5	0	0	0.5	1	0	0	0	0	0	-0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	-0.5	0
Z		-0.5	0	-0.5	0	0	-0.5	0	0	0	0	0	-2	1.5	-2	-2	-0.5	0	0	1.5	0.5	-0.5	1.5	1.5	0	1.5	0	0	2	0	2	2	1.5	1.5	0

La variable qui sort de la base est P_{28}, et celle qui entre est P_{11}

Tableau 9			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	C_b	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}	P_{26}	P_{27}	P_{28}	P_{29}	P_{30}	P_{31}	P_{32}	P_{33}
P_{33}	-1	0.5	0	0.5	0	0	0.5	0	0	0	0	0	0	0.5	0	1	0.5	0	0	-0.5	-0.5	0.5	-0.5	-0.5	0	-0.5	0	0	-1	-1	-1	-1	-0.5	-0.5	1
P_8	0	0.5	0	-0.5	0	0	-0.5	0	0	1	0	0	0	0.5	0	0	0.5	0	0	-0.5	0.5	0.5	0.5	0.5	0	-0.5	0	0	0	0	0	0	-0.5	0.5	0
P_1	0	0.5	1	0.5	0	0	-0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	0.5	0
P_3	0	0	0	0	1	0	0	0	0	0	0	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
P_9	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P_{11}	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
P_{26}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{25}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{23}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_6	0	0.5	0	0.5	0	0	0.5	1	0	0	0	0	0	-0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	-0.5	0
Z		-0.5	0	-0.5	0	0	-0.5	0	0	0	0	0	0	-0.5	0	-2	-0.5	0	0	1.5	0.5	-0.5	1.5	1.5	0	1.5	0	0	2	2	2	2	1.5	1.5	0

La variable qui sort de la base est P_{26}, et celle qui entre est P_{14}

Tableau 10			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	C_b	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}	P_{26}	P_{27}	P_{28}	P_{29}	P_{30}	P_{31}	P_{32}	P_{33}
P_{33}	-1	0.5	0	0.5	0	0	0.5	0	0	0	0	0	0	0.5	0	0	0.5	1	1	-0.5	-0.5	0.5	-0.5	-0.5	0	-0.5	0	-1	-1	-1	-1	-1	-0.5	-0.5	1
P_8	0	0.5	0	-0.5	0	0	-0.5	0	0	1	0	0	0	0.5	0	0	0.5	0	0	-0.5	0.5	0.5	0.5	0.5	0	-0.5	0	0	0	0	0	0	-0.5	0.5	0
P_1	0	0.5	1	0.5	0	0	-0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	0.5	0
P_3	0	0	0	0	1	0	0	0	0	0	0	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
P_9	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P_{11}	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	0	-1	-1	-1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
P_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
P_{25}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{23}	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_6	0	0.5	0	0.5	0	0	0.5	1	0	0	0	0	0	-0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	-0.5	0
Z		-0.5	0	-0.5	0	0	-0.5	0	0	0	0	0	0	-0.5	0	0	-0.5	-2	-2	1.5	0.5	-0.5	1.5	1.5	0	1.5	0	2	2	2	2	2	1.5	1.5	0

La variable qui sort de la base est P_{25}, et celle qui entre est P_{16}

Tableau 11			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**Cb**	**P0**	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**	**P18**	**P19**	**P20**	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**	**P28**	**P29**	**P30**	**P31**	**P32**	**P33**
P33	-1	0.5	0	0.5	0	0	0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	1	0.5	-0.5	0.5	-0.5	-0.5	0	-0.5	-1	-1	-1	-1	-1	-1	-0.5	-0.5	1
P8	0	0.5	0	-0.5	0	0	-0.5	0	0	1	0	0	0	0.5	0	0	0.5	0	0	-0.5	0.5	0.5	0.5	0.5	0	-0.5	0	0	0	0	0	0	-0.5	0.5	0
P1	0	0.5	1	0.5	0	0	-0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	0.5	0
P3	0	0	0	0	1	0	0	0	0	0	0	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
P9	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P11	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	0	-1	0	-1	-1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	-1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P23	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P6	0	0.5	0	0.5	0	0	0.5	1	0	0	0	0	0	-0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	-0.5	0
Z		-0.5	0	-0.5	0	0	-0.5	0	0	0	0	0	0	-0.5	0	0	-0.5	0	-2	-0.5	0.5	-0.5	1.5	1.5	0	1.5	2	2	2	2	2	2	1.5	1.5	0

La variable qui sort de la base est P_{23}, et celle qui entre est P_{17}

Tableau 12			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**Cb**	**P0**	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**	**P18**	**P19**	**P20**	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**	**P28**	**P29**	**P30**	**P31**	**P32**	**P33**
P33	-1	0.5	0	0.5	0	0	0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	0.5	0.5	-0.5	-1	-0.5	-1	-1	-1	-1	-1	-1	-0.5	-0.5	1
P8	0	0.5	0	-0.5	0	0	-0.5	0	0	1	0	0	0	0.5	0	0	0.5	0	0	-0.5	0.5	0.5	0.5	0.5	0	-0.5	0	0	0	0	0	0	-0.5	0.5	0
P1	0	0.5	1	0.5	0	0	-0.5	0	0	0	0	0	0	0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	0.5	0
P3	0	0	0	0	1	0	0	0	0	0	0	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
P9	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P11	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	0	-1	0	0	-1	1	0	-1	0	1	0	1	1	1	0	0	0	0	0	0
P14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	-1	0	1	0	1	1	0	0	0	0	0	0	0
P16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P6	0	0.5	0	0.5	0	0	0.5	1	0	0	0	0	0	-0.5	0	0	0.5	0	0	0.5	-0.5	-0.5	0.5	0.5	0	0.5	0	0	0	0	0	0	0.5	-0.5	0
Z		-0.5	0	-0.5	0	0	-0.5	0	0	0	0	0	0	-0.5	0	0	-0.5	0	0	-0.5	0.5	-0.5	-0.5	1.5	2	1.5	2	2	2	2	2	2	1.5	1.5	0

La variable qui sort de la base est P_{33}, et celle qui entre est P_{2}

Tableau 13			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**	**P_{22}**	**P_{23}**	**P_{24}**	**P_{25}**	**P_{26}**	**P_{27}**	**P_{28}**	**P_{29}**	**P_{30}**	**P_{31}**	**P_{32}**	**P_{33}**
P_2	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	-1	1	1	-1	-2	-1	-2	-2	-2	-2	-2	-2	-1	-1	2
P_8	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	1
P_1	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	1	1	1	1	1	1	1	1	1	1	-1
P_3	0	0	0	0	1	0	0	0	0	0	0	-1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
P_9	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P_{11}	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
P_4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	0	-1	0	0	-1	1	0	-1	0	1	0	1	1	1	0	0	0	0	0	0
P_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	-1	0	1	0	1	1	0	0	0	0	0	0	0
P_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0	0	0	1	0	0	0	0	0	0	0	0	0
P_{17}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0
P_6	0	0	0	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0	0	0	0	-1	0	1	1	1	1	1	1	1	1	1	1	0	-1
Z		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

La solution (0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) est une solution optimale de (Pa) avec une valeur optimale égale à 0. Donc (0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0) constitue une solution de base initiale du problème (P).Ceci termine la Phase I. Donc passer à la Phase II.

On va appliquer l'algorithme de simplexe au problème (P) sous sa forme canonique obtenue à la phase I.

Tableau 1			-3.5	-6.5	-4.7	-3.2	-3.25	-5.7	-5.9	-1.7	-3	-6.2	-2	-15.4	-15.6	-3.7	-3.55	-1.2	-3.4	-4.7	-0.7	-4.3	-4.4
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**
P_2	-6.5	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	-1	1	1
P_8	-1.7	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	1
P_1	-3.5	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0
P_3	-4.7	0	0	0	1	0	0	0	0	0	0	-1	0	-1	1	0	0	0	0	0	0	0	0
P_9	-3	0	0	0	0	0	0	0	0	0	1	0	0	-1	1	0	0	0	0	0	0	0	0
P_{11}	-2	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0
P_4	-3.2	0	0	0	0	1	0	0	0	0	0	1	0	0	-1	0	-1	0	0	-1	1	0	-1
P_{14}	-3.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	-1
P_{16}	-1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0
P_7	-5.9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0
P_{17}	-3.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1
P_6	-5.7	0	0	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0	0	0	0	-1	0
Z		-8.2	0	0	0	0	0.25	0	0	0	0	7.7	0	22.6	9.1	0	-1.45	0	0	0.4	9.9	11.2	6.5

La variable qui sort de la base est P_8, et celle qui entre est P_{15}

Tableau 2			-3.5	-6.5	-4.7	-3.2	-3.25	-5.7	-5.9	-1.7	-3	-6.2	-2	-15.4	-15.6	-3.7	-3.55	-1.2	-3.4	-4.7	-0.7	-4.3	-4.4
Base	**C_b**	**P_0**	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**P_{11}**	**P_{12}**	**P_{13}**	**P_{14}**	**P_{15}**	**P_{16}**	**P_{17}**	**P_{18}**	**P_{19}**	**P_{20}**	**P_{21}**
P_2	-6.5	0	0	1	0	0	1	0	0	-1	0	0	0	0	0	0	0	0	0	1	-1	0	0
P_{15}	-3.55	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	1
P_1	-3.5	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0
P_3	-4.7	0	0	0	1	0	0	0	0	0	0	-1	0	-1	1	0	0	0	0	0	0	0	0
P_9	-3	0	0	0	0	0	0	0	0	0	1	0	0	-1	1	0	0	0	0	0	0	0	0
P_{11}	-2	0	0	0	0	0	0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0
P_4	-3.2	1	0	0	0	1	0	0	0	1	0	1	0	1	-1	0	0	0	0	-1	1	1	0
P_{14}	-3.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0	0	-1
P_{16}	-1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-1	0	0	0
P_7	-5.9	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	-1	-1	0
P_{17}	-3.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1
P_6	-5.7	0	0	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0	0	0	0	-1	0
Z		-6.75	0	0	0	0	0.25	0	0	1.45	0	7.7	0	24.05	9.1	0	0	0	0	0.4	9.9	12.65	7.95

Tous les coûts relatifs sont non négatifs, donc cette base est optimale.

La solution optimale de (P) est (0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0) avec valeur optimale z = 6.75

$on\ a\ x_4 = x_{AG} = 1\ et\ x_{15} = x_{GL} = 1$

Donc le chemin minimum est A ❼ G ❼

L Alors Gare (O.N.C.F) ❼ Bab Jiaf ❼

Bab Ftouh. La longueur de ce chemin est : 6.75

Km

3. Détermination d’un flot maximal dans le réseau de transport urbain

Considérons le réseau de transport de la ville de Fès, les sommets sont composés des différents stations, deux sommets sont en liaison s’il passe au moins un bus entre deux stations, on associe à chaque arc un flot et une capacité afin de déterminer le flot maximal (saturation du réseau de transport).

➢ **<u>Représentation graphique :</u>**

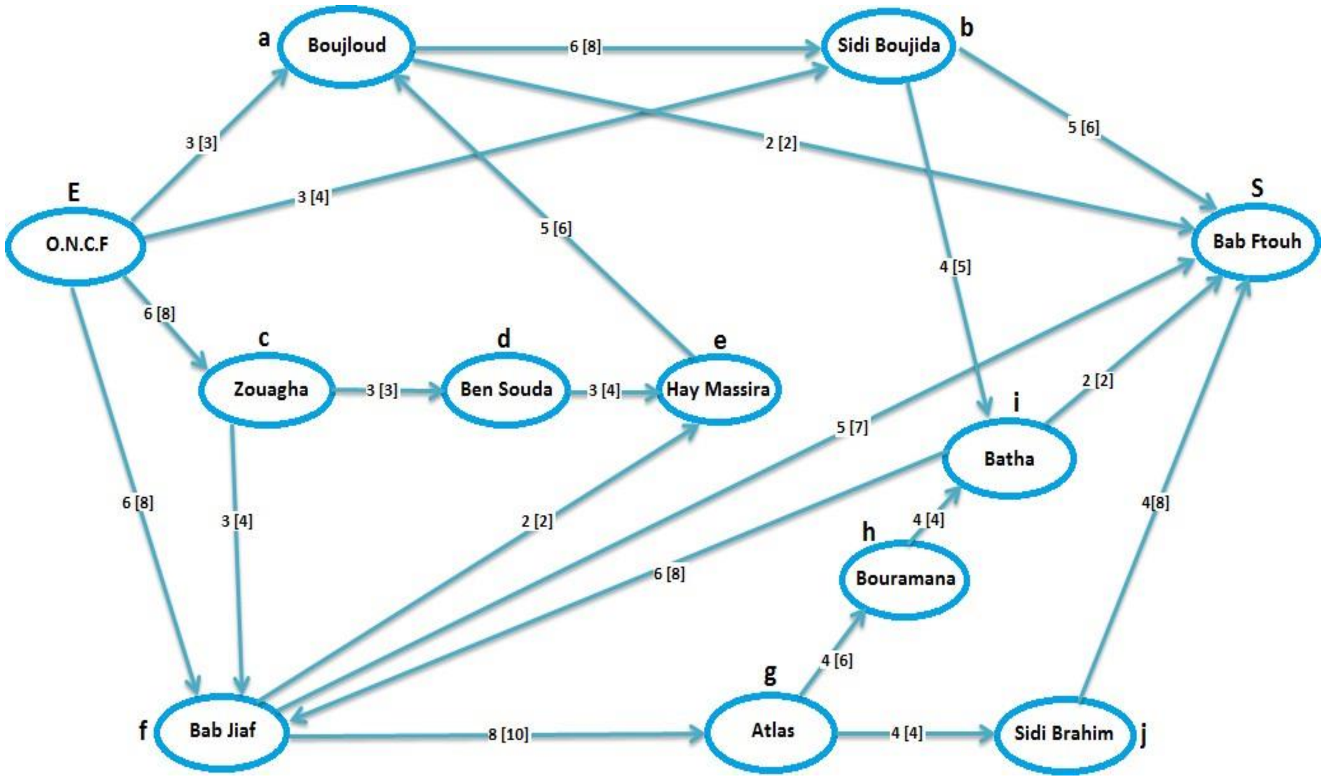


Figure 10 : représentation graphique de Carte réseau du transport urbain collectif à Fès

3.1Simulation de l’algorithme de FORD-FULKERSON (Saturation d’un réseau)

- La loi de conservation est vérifié, donc l’application φ définie bien un flot dans le réseau de transport de valeur $\varphi^0 = 18$
- $\forall u \epsilon U, 0 \leq \varphi(u) \leq c(u) \Leftrightarrow d(\varphi) = \sum_{u\epsilon U}(\varphi(u)) = 0$ Donc le flot φ est compatible.
- Par la procédure de marquage de ford-Fulkerson peut marquer les sommets suivant : **b (+E), c (+E), f (+E), S (+b)**

La sortie S du réseau est marqué, donc il existe une chaine **C= « E b S »** qui ne possède aucun arc saturé en allant de « E » à « S » donc le flot φ n'est pas maximal.

- La chaine C et l'arc u_0 forment un cycle, et donc à travers ce cycle on peut améliorer le flot φ de la manière suivante :

On pose $\varphi = \varphi^0$ et $\varphi^1 = \varphi^0 + \alpha\mu$ avec $\alpha = (\alpha_1, \alpha_2)$

E ▬ 3 (4) ➔ b ▬ 5 (6) ➔ S

α_2 n'existe pas car tous les arcs orienté dans le sens direct.

$\alpha_1 = min_{\mu^+}(c_u - \varphi_u) = min(4 - 3 ; 6 - 5) = min(1 ; 1) = 1$

Donc $\varphi^1 = 18 + 1 = 19$

Itération 2 :

- Par la procédure de marquage de ford-Fulkerson peut marquer les sommets suivants : **c (+E), f (+E), S (+E)**

  La sortie S du réseau est marqué, donc il existe une chaine **C^1= « E f S »** qui ne possède aucun arc saturé en allant de « E » à « S » donc le flot φ^1 n'est pas maximal.
- Soit $\mu^1 = C^1 \cup \{u_0\}$ un cycle, et donc à travers ce cycle on peut améliorer le flot φ de la manière suivant :

  On pose $\varphi^2 = \varphi^1 + \alpha\mu^1$

E ▬ 6 (8) ➔ f ▬ 5 (7) ➔ S

α_2 n'existe pas car tous les arcs orienté dans le sens direct.

$\alpha_1 = min_{\mu^+}(c_u - \varphi_u) = min(8 - 6 ; 7 - 5) = min(2 ; 2) = 2$

Donc $\varphi^2 = 19 + 2 = 21$

Itération 3 :

- Par la procédure de marquage de ford-Fulkerson peut marquer les sommets suivants : **c (+E), f (+E), g (+f), i (-f), h (-i), a (-b), e (a), d (-e), c (-d)**

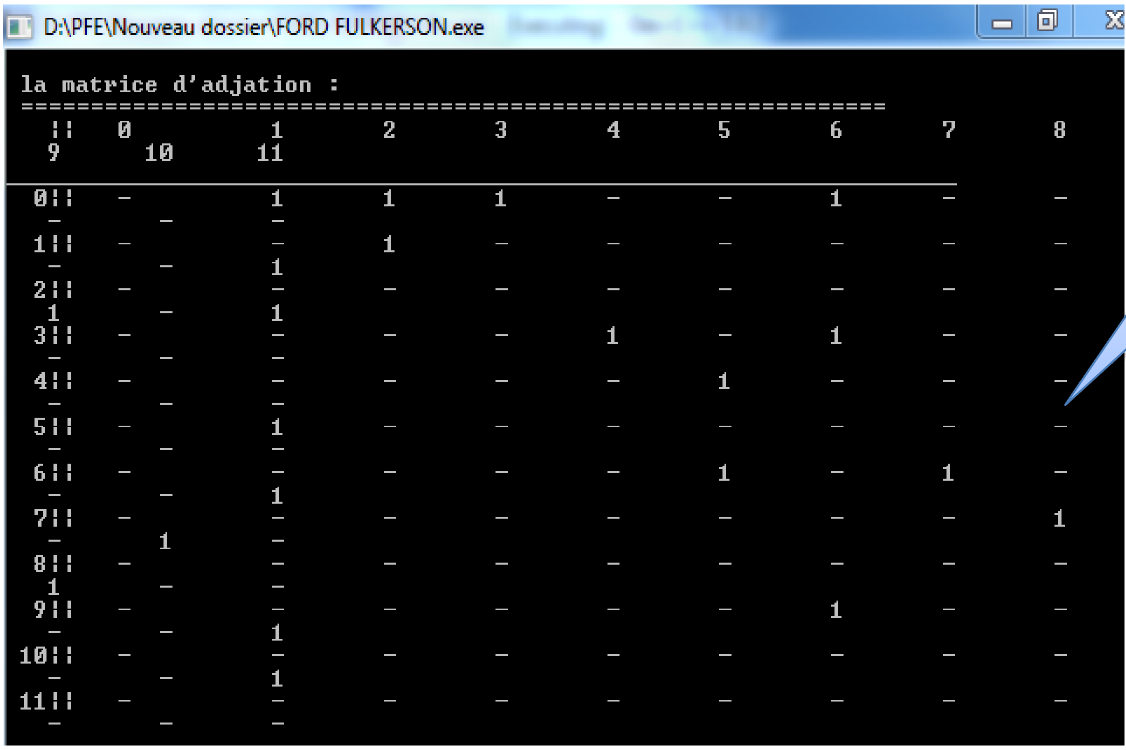
  La sortie S du réseau n'est pas marquée, donc le flot est maximal.

Conclusion :

Le flot maximal dans ce réseau de transport est $\varphi^2 = 21$

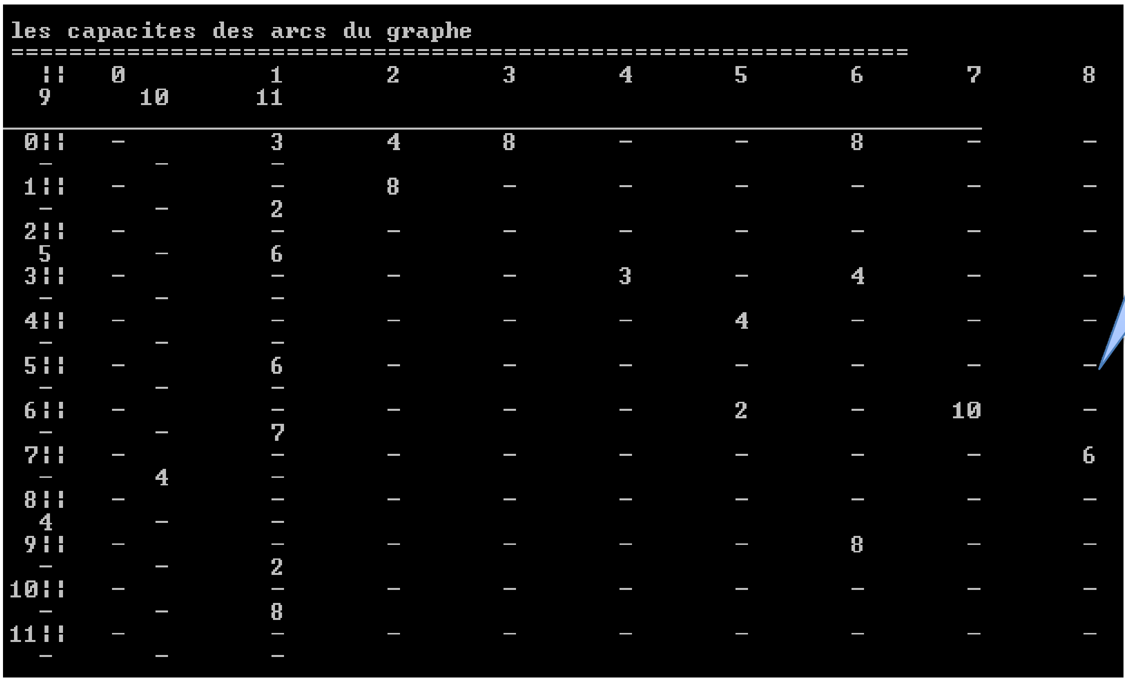
3.2 Programmation en langage C

➢ **<u>L'exécution de programme :</u>**




```
la matrice d'adjation :
==================================================================
  ||   0          1        2        3        4        5        6        7        8
  9       10         11
  0||  -          1        1        1        -        -        1        -        -
  -       -          -
  1||  -          -        1        -        -        -        -        -        -
  -       -          1
  2||  -          -        -        -        -        -        -        -        -
  1       -          1
  3||  -          -        -        -        1        -        1        -        -
  -       -          -
  4||  -          -        -        -        -        1        -        -        -
  -       -          -
  5||  -          1        -        -        -        -        -        -        -
  -       -          -
  6||  -          -        -        -        -        1        -        1        -
  -       -          1
  7||  -          -        -        -        -        -        -        -        1
  -       1          -
  8||  -          -        -        -        -        -        -        -        -
  1       -          -
  9||  -          -        -        -        -        -        1        -        -
  -       -          1
 10||  -          -        -        -        -        -        -        -        -
  -       -          1
 11||  -          -        -        -        -        -        -        -        -
  -       -          -
```

Affichage de la matrice d'adjacence



```
les capacites des arcs du graphe
==================================================================
  ||   0          1        2        3        4        5        6        7        8
  9       10         11
  0||  -          3        4        8        -        -        8        -        -
  -       -          -
  1||  -          -        8        -        -        -        -        -        -
  -       -          2
  2||  -          -        -        -        -        -        -        -        -
  5       -          6
  3||  -          -        -        -        3        -        4        -        -
  -       -          -
  4||  -          -        -        -        -        4        -        -        -
  -       -          -
  5||  -          6        -        -        -        -        -        -        -
  -       -          -
  6||  -          -        -        -        -        2        -        10       -
  -       -          7
  7||  -          -        -        -        -        -        -        -        6
  -       4          -
  8||  -          -        -        -        -        -        -        -        -
  4       -          -
  9||  -          -        -        -        -        -        8        -        -
  -       -          2
 10||  -          -        -        -        -        -        -        -        -
  -       -          8
 11||  -          -        -        -        -        -        -        -        -
  -       -          -
```

Affichage des capacités des arcs du graphe les flux des arcs du graphe :

	0	1	2	3	4	5	6	7	8	9	10	11
0	-	3	3	6	-	-	6	-	-	-	-	-
1	-	-	6	-	-	-	-	-	-	-	-	2
2	-	-	-	-	-	-	-	-	-	4	-	5
3	-	-	-	-	3	-	3	-	-	-	-	-
4	-	-	-	-	-	3	-	-	-	-	-	-
5	-	5	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	2	-	8	-	-	-	5
7	-	-	-	-	-	-	-	-	4	-	4	-
8	-	-	-	-	-	-	-	-	-	4	-	-
9	-	-	-	-	-	-	6	-	-	-	-	2
10	-	-	-	-	-	-	-	-	-	-	-	4
11	-	-	-	-	-	-	-	-	-	-	-	-

Affichage des flux des arcs du graphe

la loi de kirchoff est verifie

	0	1	2	3	4	5	6	7	8	9	10	11
0	-	-	1	1	-	-	1	-	-	-	-	-
1	-1	-	1	-	-	-1	-	-	-	-	-	-
2	-1	-1	-	-	-	-	-	-	-	1	-	1
3	-1	-	-	-	-	-	1	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-
6	-1	-	-	-1	-	-	-	1	-	-1	-	1
7	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-1	-	-	-	1	-	-1	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-

Affichage des chaines possibles pour aller de l'entrée vers la sortie.

1 : signifie qu'il existe un arc avant entre deux sommets

-1 : signifie qu'il existe un arc arrière entre deux

	0	1	2	3	4	5	6	7	8	9	10	11
0	-	-	1	1	-	-	1	-	-	-	-	-
1	-1	-	1	-	-	-1	-	-	-	-	-	-
2	-1	-1	-	-	-	-	-	-	-	1	-	1
3	-1	-	-	-	-	-	-	[illegible]	[illegible]	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-
6	-1	-	-	-1	-	-	-	1	-	-1	-	1
7	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-1	-	-	-	1	-	-1	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-

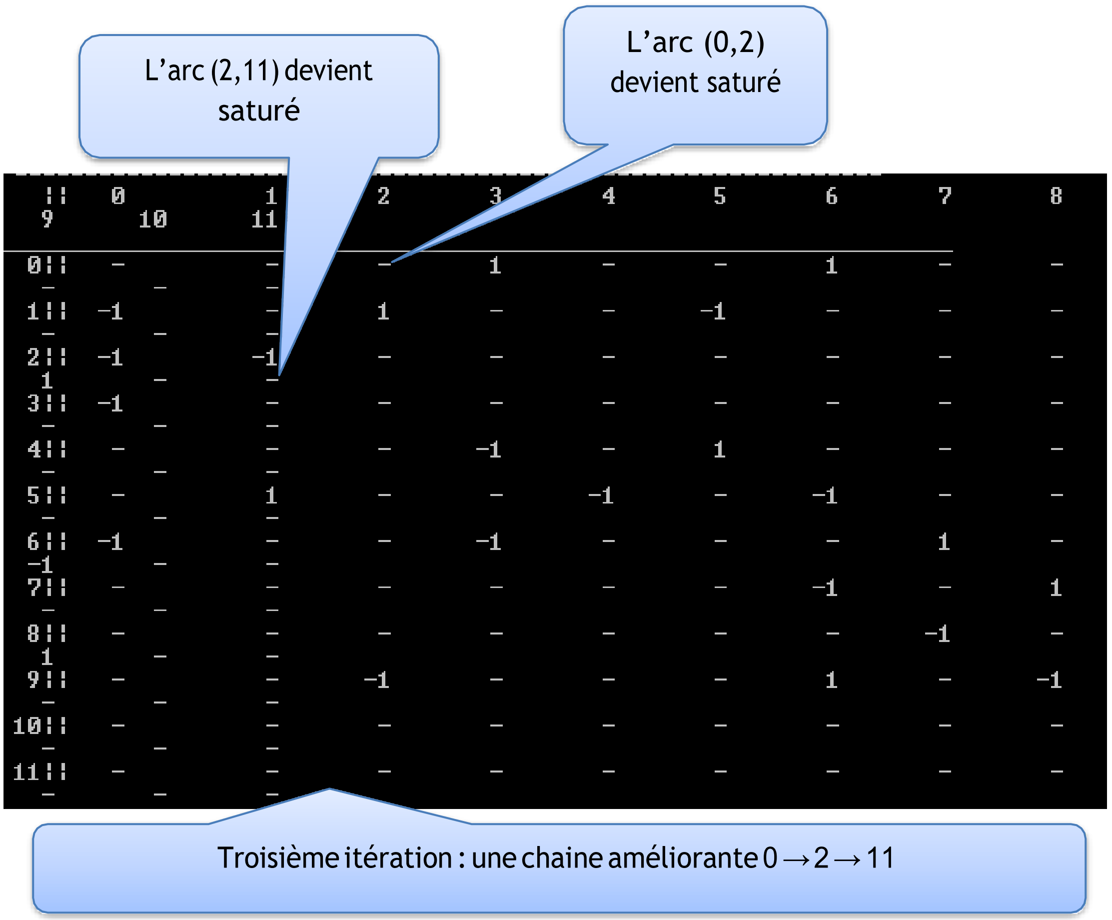
L'arc (3,6) devient saturé

Première itération : une chaine améliorante 0→3→6→11

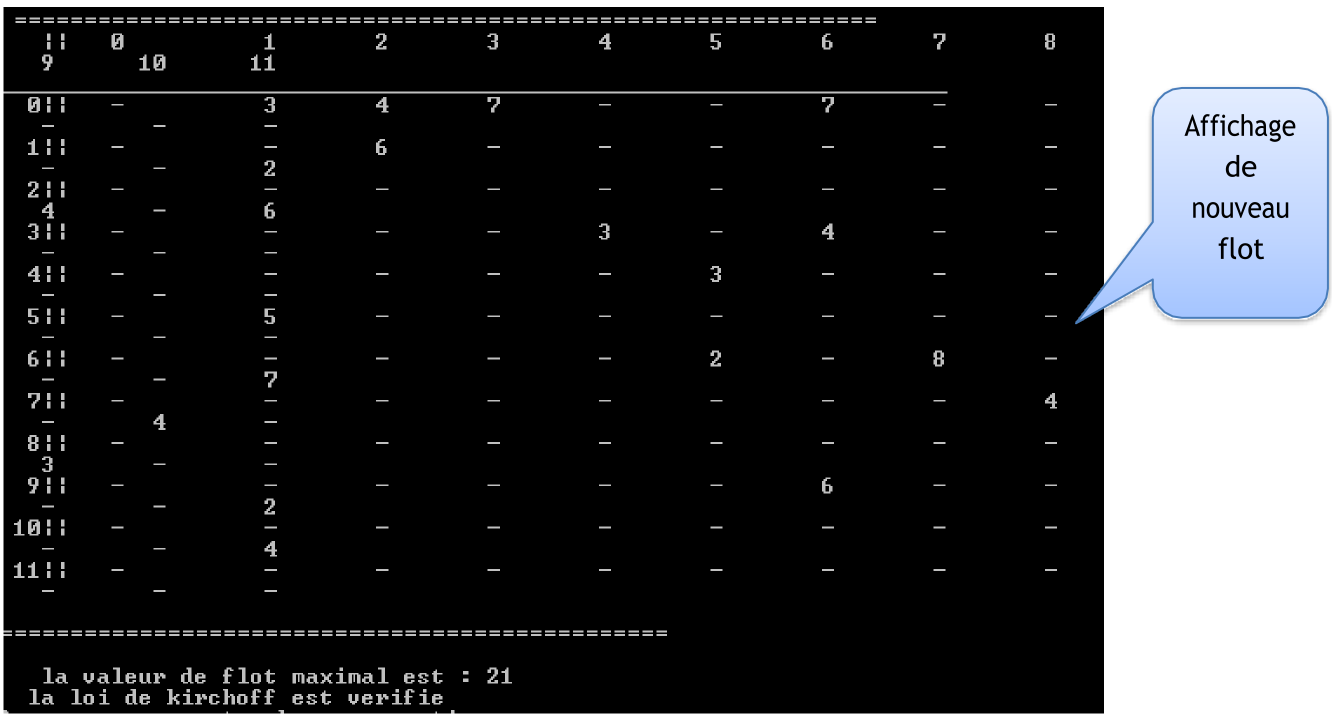
	0	1	2	3	4	5	6	7	8	9	10	11
0	-	-	1	1	-	-	1	-	-	-	-	-
1	-1	-	1	-	-	-1	-	-	-	-	-	-
2	-1	-1	-	-	-	-	-	-	-	1	-	1
3	-1	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-
6	-1	-	-	-1	-	-	-	1	-	-1	-	-
7	-	-	-	-	-	[illegible]	[illegible]	[illegible]	[illegible]	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-1	-	-	-	1	-	-1	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-

L'arc (6,11) devient saturé

Deuxième itération : une chaine améliorante 0→6→11



	0	1	2	3	4	5	6	7	8	9	10	11
0	-	-	-	1	-	-	1	-	-	-	-	-
1	-1	-	1	-	-	-1	-	-	-	-	-	-
2	-1	-1	-	-	-	-	-	-	-	1	-	-
3	-1	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-1	-	1	-	-	-	-	-	-
5	-	1	-	-	-1	-	-1	-	-	-	-	-
6	-1	-	-	-1	-	-	-	1	-	-1	-	-
7	-	-	-	-	-	-	-1	-	1	-	-	-
8	-	-	-	-	-	-	-	-1	-	1	-	-
9	-	-	-1	-	-	-	1	-	-1	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-



	0	1	2	3	4	5	6	7	8	9	10	11
0	-	3	4	7	-	-	7	-	-	-	-	-
1	-	-	6	-	-	-	-	-	-	-	-	2
2	-	-	-	-	-	-	-	-	-	4	-	6
3	-	-	-	-	3	-	4	-	-	-	-	-
4	-	-	-	-	-	3	-	-	-	-	-	-
5	-	5	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	2	-	8	-	-	-	7
7	-	-	-	-	-	-	-	-	4	-	4	-
8	-	-	-	-	-	-	-	-	-	3	-	-
9	-	-	-	-	-	-	6	-	-	-	-	2
10	-	-	-	-	-	-	-	-	-	-	-	4
11	-	-	-	-	-	-	-	-	-	-	-	-

```
la valeur de flot maximal est : 21
la loi de kirchoff est verifie
```

Conclusion :

En conclut que les résultats théoriques et les résultats obtenus à travers le programme en langage C sont identiques.

Conclusion générale :

Ce projet de fin d'étude, m'a donné l'opportunité de me familiariser au domaine de la recherche opérationnelle, qui est la discipline des méthodes scientifiques pour aider à mieux décider et traiter les problèmes stratégiques et économiques.
Dans ce rapport, on s'est intéressé d'avantage à la modélisation et la résolution du problème de plus court chemin et problème du flot maximal par les méthodes de DANTZIG et FORD-FULKERSON afin d'obtenir des solutions bien déterminés, ensuite on a travaillé sur un problème de transport urbain dans le but de faciliter la tâche de déplacements aux habitants de la ville de Fès « en appliquant les deux méthodes cités précédemment ».
Et enfin, on a essayé de faire une comparaison entre les résultats obtenue par ces méthodes théoriquement et leurs programmes en langage C qui ont donné un résultat identique.

Références :

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Riverst and Clifford Stein, Introduction to Algorithms, 2001.

[2] Andrew V. Goldberg and Robert E. Tarjan, A new approach to the maximum-flow problème, 1988.

[3] Robert. Faure, Bernard. Lemaire and Christophe. Picouleau, Précis de recherche opérationnelle, 1992.

Sites web

http://www.phpsimplex.com/fr/theorie_modelisation_problemes.htm

http://www.lamsade.dauphine.fr/~mayag/LECTURENOTES.pdf

https://fr.slideshare.net/sanaaroussi3/chapitre-2-problme-de-plus-court- chemin

http://www.lgi.ecp.fr/~mousseau/Cours/S4/pmwiki/uploads/Main/Flots.pdf