REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L’ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE





UNIVERSITE MOHAMED BOUDIAF - M’SILA

FACULTE DES MATHEMATIQUES ET

DE L’INFORMATIQUE

**Département d’Informatique**

**MEMOIRE de fin d’études**

**Présenté pour l’obtention du diplôme de MASTER**

**Domaine** : Mathématiques et Informatique

**Filière** : Informatique

**Spécialité** : Informatique Décisionnel et Optimisation

**Par : Kheireddine yamina**

**Thème**

|  |
| --- |
| *Quelques applications réelles du problème de flot* |

**Soutenu le : / /2020 devant lejury composé de :**

**AbderrahimGuerna Université de M’sila Président**

**Adel Moussaoui Université de M’sila Rapporteur**

**Nasereddine.Amroune Université de M’sila Examinateur**

**Promotion : 2019 /2020**

# Remerciement

***Je tiens à exprimer toute ma reconnaissance à mon directeur de mémoire, Dr.Adel Moussaoui,je le remercie de m’avoir encadré orienté aidé et conseillé.***

***J’adresse mes sincères remerciements à tous les professeurs intervenants et toutes les personnes qui par leurs paroles leurs écrits leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.***

***J’exprime toute ma gratitude aux membres du jurypour le temps qu’ils ont consacré à la lecture de cemodeste travail et pour l’honneur qu’ils me font d’évaluer ce travail.***

***Et enfin un merci chaleureux est adressé à mes collègues les étudiants enmaster 2 IDO.***

# Dédicace

***Je dédiée ce modeste travail à ma mère qui était toujours à mes cotes et à mon chère papa.***

***A mes sœurs et à mes frères.***

***A mes grands-parents et ceux qui ont partagé avec moi tous les moments d’émotion lors de la réalisation de ce travail ils m’ont chaleureusement supporté et encouragé tout au long de mon parcours.***

***A ma famille mes proches et à ceux qui me donnent de l’amour et de la vivacité.***

***A tous mes amis qui m’ont toujours encouragé et à qui je souhaite plus de succès.***

***A tous ceux que j’aime***

***Merci***

Table des matières

[Remerciement I](#_Toc50750280)

[Dédicace III](#_Toc50750281)

[Liste de figure VII](#_Toc50750282)

[Résume: VIII](#_Toc50750283)

[Introduction générale 3](#_Toc50750284)

[CHAPITRE 1 :NOTIONS DE BASES SUR LA THÉORIE DES GRAPHES 3](#_Toc50750285)

[1.1 Introduction 4](#_Toc50750286)

[1.2 Qu’est-ce qu’un graphe : 4](#_Toc50750287)

[1.2.1.Graphe non-orienté 4](#_Toc50750288)

[1.2.2.Graphe orienté 5](#_Toc50750289)

[1.2.3.Graphe simple/multiple : 6](#_Toc50750290)

[1.2.4.p-graphe : 6](#_Toc50750291)

[1.2.5.Graphe partiel  6](#_Toc50750292)

[*1.2.6.Sous-graphe* 6](#_Toc50750293)

[1.3. Degré d’un sommet 6](#_Toc50750294)

[Graphe non orienté 6](#_Toc50750295)

[Graphe orienté 6](#_Toc50750296)

[1.4. Différents type de graphes :  6](#_Toc50750297)

[1.4.1 Graphe complet  6](#_Toc50750297)

[1.4.2 Graphe simple 7](#_Toc50750298)

[1.4.3 Graphe biparti 7](#_Toc50750299)

[1.5 Connexité 8](#_Toc50750300)

[1.5.1 Chaînes et chemins : 8](#_Toc50750301)

[1.5.2 Cycle : 8](#_Toc50750302)

[1.5.3 Circuit : 9](#_Toc50750303)

[1.5.4 Graphe Connexe 9](#_Toc50750304)

[1.5.5 Composante connexe: 9](#_Toc50750305)

[1.5.6 graphe fortement connexe : 10](#_Toc50750306)

[1.5.7 Composante fortement connexe : 11](#_Toc50750307)

[1.6 Représentation Matricielle des graphes 11](#_Toc50750308)

[1.6.1 Matrice d’adjacence (sommet-sommet) : 11](#_Toc50750309)

[1.6.2 Matrice d’incidence (sommet-arc) : 13](#_Toc50750310)

[Remarques: 13](#_Toc50750311)

[Chapitre 02 : Notion de Flot 3](#_Toc50750312)

[2.1 Introduction 15](#_Toc50750313)

[2.2 Flots: définitions et propriétés 15](#_Toc50750314)

[2.2.1 Définition : 15](#_Toc50750315)

[2.2.2Définition algébrique des flots 15](#_Toc50750316)

[2.2.3Opérations sur les flots : 16](#_Toc50750318)

[2.3 Tensions : définitions et propriétés 16](#_Toc50750319)

[2.3.1 Définition : 16](#_Toc50750320)

[2.3.2 Propriétés sur les tensions: 17](#_Toc50750321)

[2.4 Le problème de flot de valeur maximum ou de cout minimum 17](#_Toc50750322)

[2.4.1 Flot de valeur maximale 17](#_Toc50750323)

[2.4.2 Recherche d’un flot complet : 18](#_Toc50750324)

[2.4.3 Position du problème 21](#_Toc50750325)

[2.4.4 Algorithme de Ford et Fulkerson 22](#_Toc50750326)

[2.4.5 Problème du flot compatible: 23](#_Toc50750327)

[2.4.6 Flot de cout minimum ; 26](#_Toc50750328)

[Chapitre 3 : Problème de réseau électrique 27](#_Toc50750329)

[3.1Introduction 28](#_Toc50750330)

[3.2. Le réseau électrique 28](#_Toc50750331)

[3.2.1 Disjoint path problem 28](#_Toc50750332)

[3.3 Le problème de réseau électrique de la wilaya de M’sila 28](#_Toc50750333)

[3.3.1 Le réseau considéré 28](#_Toc50750334)

[3.3.2 Modélisation du problème 29](#_Toc50750335)

[3.4 La résolution des problèmes et interprétation des résultats 30](#_Toc50750336)

[3.4.1 Résolution 30](#_Toc50750337)

[Conclusion 33](#_Toc50750338)

[Chapitre 4 : Problème de Concours de recrutement. 33](#_Toc50750339)

[4.1 Introduction 34](#_Toc50750340)

[4.2 Problème de concours de recrutement 34](#_Toc50750341)

[4.2.1 Maximum Bipartite Matching 34](#_Toc50750342)

[4.2.2 Maximum Bipartite Matching and Max Flow Problem 34](#_Toc50750343)

[4.3 Le problème de Concours de recrutement d’hôpital ben khnatha en Ben Srour 35](#_Toc50750344)

[4.3.1 Modélisation du problème 35](#_Toc50750345)

[Concours d’hôpitalsous forme d’un graphe bipartite 35](#_Toc50750346)

[4.3 La résolution des problèmes et interprétation des résultats 36](#_Toc50750347)

[4.3.1 Résolution 36](#_Toc50750348)

[Conclusion 37](#_Toc50750349)

[Conclusion générale 39](#_Toc50750350)

# Liste de figure

1.1 Graphe non orienté . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ………………..5

1.2 Graphe orienté . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . …………………….5

1.3Graphe complet et non complet . . . . . . . . . . . . . . …. . . . . . . . . . . . . . . . . . . . ... 7

1.4Graphe biparti.. . . . . . . . . . . . . . . . . . . . . . . ………………………. . . . . . . . . . ….7

1.5Graphe est fortement connexe . . . . . . . . . . . . . . . . . . . . . . . . . . ………….. . . 10

1.6Graphe 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ……………………………….. 11

1.7Graphe 2. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . …………………….. . . . . . . . 11

1.8Exemple de sommet . . . . . . . . . . . . . . . . . . . . . . ……... . . . . . . . . . . . . . . . . ...12

1.9Exemple de sommet . . . . . . . . . . . . . . . . . . . . . ……… . . . . . . . . . . . . . . . . ......12

1.10 Sommet d’une matrice d’incidence . . . . . . . . . . . . ….. . . . . . . . . . . . . . . . . …13

2.1 Exemple d’un flot complet…………………………………………………………… 18

2.2 Etape1………………………………………………………………………………… .19

2.3 Etape 2………………………………………………………………………………….19

2.4 Etape3…………………………………………………………………………………...20

2.5. Algorithme de Ford et Fulkerson…………………………………………………….23

2.6 Organigramme de l’algorithme…………………………………………………… ....26

3.1 Résolution de problème ………………………………………………………………30

3.2 L’accueil……………………………………………………………………………… 31

3.3 Résultat…………………………………………………………………………………32

4.1Resolutin de problème……………………………………………………………… 36

4.2.L’acceuil……………………………………………………………………………… 37

4.3 Résultat…………………………………………………………………………………38

# Résume:

**L’objectif de ce travail est d’étudier quelques applications réelles du problème de flot et de son importance en théories de graphes. Nous avons appliqué la théorie du flot aux réseaux électriques de la wilaya de M’sila et au concours de recrutement de l’hôpital de Ben Srour.**

**ملخص**

**الهدف من هذا العمل هو دراسة بعض التطبيقات الحقيقية للتدفقات واهميتها في نظرية الرسم البياني. تم تطبيق هذه المفاهيم على مشكلة الشبكة الكهربائية لولاية المسيلة ومسابقة التوظيف لمستشفى بن سرور كمثال.**

**Abstract:**

**The objective of the work is to study some real applications of the flow problem and its importance in graph theories. We applied the theory of the flow to the electricity grid of M’sila and the recruitment competition of the hospital of Ben Srour.**

# Introduction générale

La théorie des graphes est un très vaste domaine, en évolution constante tant du point de vue des recherches fondamentales que de celui des applications. Ces dernières sont très nombreuses.

Les problèmes de flot et leurs dérivés : problèmes de transport, d’affectation réseau électrique constituent un très important domaine d’application de la théorie des graphes. Sous leur forme la plus simple, ils consistent organiser de façon optimale, sous diverses contraintes, les mouvements de certaines quantités d’un bien dans un réseau.

Les caractéristiques de flux élémentaires sont supposées suffisantes pour définir n’importe quelle politique de flot. Elles dépendent, bien sûr, du problème étudié : on distingue des caractéristiques physiques telles que les capacités de débit, d’écoulement et de réception, et des caractéristiques économiques telles que les coûts unitaires de mouvement ou d’attente.

Dans ce mémoire nous portant un intérêt particulier aux problèmes de flots. Nous essayerons dans un premier temps de rappeler quelques définitions et résultats sur la théorie des graphes. Par la suit nous donnerons un aperçu sur les notions de flot et tension.

Dans le premier chapitre, nous donnerons les concepts de base de la théorie des graphes. Dans le deuxième chapitre nous présenterons les notions de flots et tensions. Dans le troisième chapitre nous avons étudié le problème de réseau électrique. Le quatrième chapitre est dédié au concours de recrutement.

# CHAPITRE 1 :NOTIONS DE BASES SUR LA THÉORIE DESGRAPHES

## Introduction

Les graphes permettent de modéliser certaines situations dans lesquelles il y a des interactions entre les objets. Les techniques utilisées en théorie des graphes permettent de répondre à beaucoup de problèmes algorithmiques posés. En effet, étudier les propriétés de ces problèmes revient à étudier les propriétés structurelles de leurs topologies représentées par des graphes. Et ce premier chapitre correspond aux préliminaires nécessaires à la lecture de ce manuscrit. Il introduit les notions utilisées en Théorie des Graphes.

## Qu’est-ce qu’un graphe :

Un graphe G est un schéma constitué par deux ensembles finis, un ensemble V non vide de points { v1,v2,….,vn} appelés sommets, et par un ensemble E (qui peut être vide) de segments(flèches) {u1 , u2,….. um} appelés arcs. On note par e = (v1, v2) une arête e ayant v1 et v2 comme extrémités. [1]

▪ **Le terme réseau:** utilisé pour désigner des systèmes réels (réseau routier, Réseau électrique, réseau informatique …..etc.).

▪ **Le terme Graphe :**est utilisé pour désigner une représentation mathématique d'un réseau.

Mais, généralement : “*Réseau*” ≡ “*Graphe*”[2]

1.2.1.Graphe non-orienté :

Un graphe non-orienté est un graphe où le couple (x ,y) est représenté graphiquement par x — y .

On dit que les sommets x et y sont adjacents s’ils sont reliés par une arête ou un arc.

L’ensemble des sommets adjacents au sommet x V est noté Γ (x) [3]

###### **Figure1 -1 graphe non-orienté**

1.2.2.Graphe orienté :

Un graphe orienté (digraphe) est un graphe où le couple (x , y) est représenté graphiquement par x y, On appelle x l’extrémité initiale ou origine , et y est l’extrémité terminale .

-L’arc u = (x , y) est dit sortant de x et incident en y, et y est un successeur de x noté par Succ(x), tandis que x est un prédécesseur de y noté par préd(x)[3].

###### **Figure 1-2 : graphe orienté**

### **1.2.3.Graphe simple/multiple** **:**

Un graphe G est dit simple s’il ne comporte pas de boucle , et si chaque paire de sommets x et y sont relié par au plus un arc (arête). Dans le cas contraire, il sera dit multiple.[4]

### **1.2.4.p-graphe** **:**

Un p-graphe est un graphe orienté comporte au plus p arcs entre deux sommets quelconques de G. Le plus souvent, on étudiera des 1-graphes .[5]

1.2.5.Graphe partiel :Soit G(X, U) un graphe, on appelle graphe partiel de G, le graphe G'(X', U') tel que: X'=X, U' U (c'est un graphe ayant le même ensemble X de sommets, mais un ensemble diffèrent des arcs U' U). [2]

#### ***1.2.6.Sous-graphe*** :

Soit G(X, U) un graphe

* *G'(X', U')* est un sous graphe de *G* si *X' X*
* U'={uU/ les deux extrémités de u appartiennent à X'}

(c.à.d., on supprime des sommets et des arcs)[2]

1.3. Degré d’un sommet :

### Graphe non orienté

Degré : le nombre des arêtes qui relient les sommets

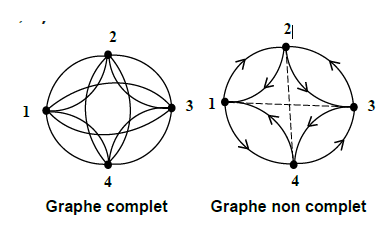
### Graphe orienté

Dans un graphe orienté, on définit des degrés intérieurs des degrés extérieurs. Le degré total d'un sommet est la somme du degré intérieur et le degré extérieur.

**1.4.**Différents types de graphes :

1.4.1Graphe complet :

Un graphe G(X, U) est dit complet si seulement si: *x, yX, (x, y) U et (y, x) U[2] .*



###### **Figure 1-3 : graphe complet et non- complet**

1.4.2 Graphe simple : Un graphe G(X, U) est dit simple si seulement si:

1. Ne contient aucune boucle.
2. ∀ x, y X, ∃ au plus u=(x, y)U

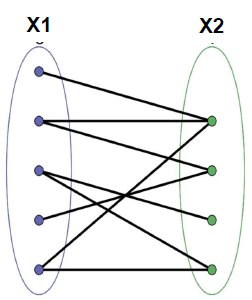
1.4.3 Graphe biparti : Un graphe biparti (ou un bi-graphe) est un graphe dont l'ensemble de ses sommets peuvent être divisé en deux sous-ensembles disjoints de sommets X1 et X2 tels que deux sommets du même sous-ensemble ne soient pas adjacents (c.à.d. tout arc de G a une extrémité dans X1 et l'autre dans X2).

Et on note:

G(X, U) avec: X = X1X2 et X1 X2= 

Et si u=(x1, x2) et x1  X 1 ===> x2X2

x1X2 ===> x2X1[2]



###### **Figure1-4 : graphe biparti**

## 1.5 Connexité

### 1.5.1 Chaînes et chemins :

#### 1.5.1.1*chemins*

On appelle un chemin de G(X, U) la suite de sommets a1, a2, …, ap (sans coupure) tel que:

▪ (ai, ai+1)U avec i=1, 2, …, p-1. ▪ ai est l'extrémité initiale du chemin ▪ ap est l'extrémité terminale du chemin

\*chemin simple :a1, a2, …, ap est un chemin simple tel que:  i≠j, (ai, ai+1)≠(ai, ai+1) (c.à.d. un arc est pris une seule fois).

\*chemin élémentaire : est un chemin qui ne prend les sommets qu'une seule fois.

\*. Un chemin hamiltonien: est un chemin élémentaire qui passe par tous les sommets du graphe.

\*. Un chemin eulérien : est un chemin simple qui passe par tous les arcs du graphe une et une seule fois.

1.5.1.2chaînes :

Une chaîne d'un graphe non orienté (ou orienté mais l'orientation n'est pas prise en considération) G(X, U) est une suite de sommets a1, a2, …, ap (sans coupure) tel que: (ai, ai+1)U ou (ai+1, ai)U avec i=1, 2, …, p-1.

* chaîne simple : C’est une chaîne ou les (arcs/arêtes) sont pris(es) une seule fois.
* chaîne élémentaire : C'est une chaîne dont tous les sommets du graphe sont pris une seule fois.
* Une chaîne hamiltonienne: est une chaîne élémentaire qui passe par tous les sommets du graphe.
* Une chaîne eulérienne: est une chaîne qui passe par tous les arcs du graphe une et une seule fois.

### 1.5.2 Cycle :

Un cycle est une chaîne a1a2a3 ….ap tel que: a1=ap

c.à.d. l'extrémité initiale et l'extrémité terminale sont confondues

\* Un cycle hamiltonien: est un cycle élémentaire qui passe par tous les sommets du graphe.

\* Un cycle eulérien: est une chaîne simple et fermée qui passe par tous les arcs du graphe une et une seule fois.

### 1.5.3 Circuit :

Un circuit est un chemin a1a2a3 ….ap tel que: a1=aP c.à.d. les deux extrémités initiale et terminale sont confondues (un chemin fermé).

\* Un circuit hamiltonien: est un circuit élémentaire qui passe par tous les sommets du graphe.

\*Un circuit eulérien: est un chemin simple et fermé qui passe par tous les arcs du graphe une et une seule fois. [2]

1.5.4 Graphe Connexe :Un graphe G est dit connexe s’il existe une chaîne qui relie tous les sommets du graphe deux à deux, si G n’est pas connexe alors il se décompose en plusieurs sous-graphes appelés composantes connexes.

[6] .

### **1.5.5 Composante connexe**:

On appelle composantes connexes un ensemble de sommets, qui ont deux à deux la relation de connexité, de plus tout sommet en dehors de la composante n’a pas de relation de connexité avec les sommets de cette composante.

Les composantes connexes d’un graphe se déterminent en utilisant un algorithme de marquage simple.

#### Algorithme de marquage

L’algorithme de marquage simple est un algorithme qui permet de déterminer les composantes connexe d’un graphe.

Principe **:**

Soit G = (V;E) un graphe orienté (digraphe) . L’idée de cet algorithme est la suivante : Pour un sommet quelconque v  G, il s’agit de trouver toutes les chaînes reliant ce sommet aux autres sommets de graphe, ainsi les sommets reliés au sommet v par une chaîne forment la composante connexe qui contient le sommet v [4]

1.5.6 graphe fortement connexe : Un graphe G(X, U) est dit fortement connexe si: ∀ x, y ∈X, il existe un chemin de x vers y et un autre de y vers x.

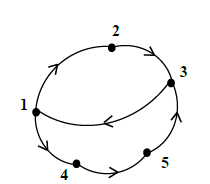
Exemple:

(1, 2) ∃ un chemin (1, 2)

(2, 1) ∃ un chemin (2, 3), (3, 1)

(2, 3) ∃ un chemin (2, 3)

(3, 2) ∃ un chemin (3, 1), (1, 2)

(3, 1) ∃ un chemin (3, 1)

(1, 3) ∃ un chemin (1, 2), (2, 3)

(1, 4), (4, 5), (5, 3)

(3, 4) ∃ un chemin (3, 1), (1, 4)

(4, 3) ∃ un chemin (4, 5), (5, 3)

(4, 5) ∃ un chemin (4, 5)

(5, 4) ∃ un chemin (5, 3), (3, 1), (1, 4)

(5, 3) ∃ un chemin (5, 3) **figure1-5 :G est fortement connexe**

(3, 5) ∃ un chemin (3, 1), (1, 4), (4, 5)

(1, 4) ∃ un chemin (1, 4)

(4, 1) ∃ un chemin (4, 5), (5, 3), (3, 1)

(1, 5) ∃ un chemin (1, 4), (4, 5)

(5, 1) ∃ un chemin (5, 3), (3, 1)

(2, 4) ∃ un chemin (2, 3), (3, 1), (1, 4)

(4, 2) ∃ un chemin (4, 5), (5, 3), (3, 1), (1, 2)

(2, 5) ∃ Un chemin (2, 3), (3, 1), (1, 4), (4, 5)

(5, 2) ∃ un chemin (5, 3), (3, 1), (1, 2).

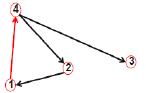
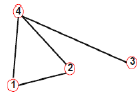
On constate que ∀x, y∈X On peut trouver un chemin de x vers y et un autre de y vers x ===> le graphe G(X, U) est un graphe fortement connexe. [2]

### 1.5.7Composante fortement connexe :

On appelle une composante fortement connexe d'un graphe G(X, U) un sous graphe G'(X', U') fortement connexe.[2]

## 1.6 Représentation Matricielle des graphes

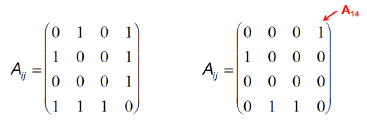
### 1.6.1 Matrice d’adjacence (sommet-sommet) :



**Figure1-6 :Graph G1 figure1-7 : Graph G2**

{𝑨𝒊𝒋 = 𝟏𝒔′𝒊𝒍 𝒆𝒙𝒊𝒔𝒕𝒆 𝒖𝒏 𝒍𝒊𝒆𝒏 𝒆𝒏𝒕r𝒆 𝒍𝒆 𝒔𝒔𝒐𝒎𝒎𝒆𝒕𝒔 𝒊 𝒆𝒕 𝒋

𝑨𝒊𝒋 = 𝟎 𝒔𝒊 𝒍𝒆𝒔 𝒅𝒆𝒖𝒙 𝒔𝒐𝒎𝒎𝒆𝒕𝒔 𝒊 𝒆𝒕 𝒋 𝒏𝒆 𝒔𝒐𝒏𝒕 𝒑𝒂𝒔 𝒄𝒐𝒏𝒏𝒆𝒄𝒕é𝒔}



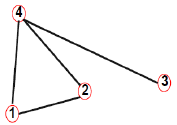
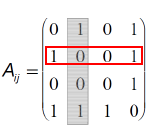
**A(N,N) A(N,N)**



**N.B:** Notons que pour un graphe orienté (à droite) la matrice n'est pas symétrique.

#### Matrice d'adjacence et degrés de sommets :

1. **Graphe non orienté**

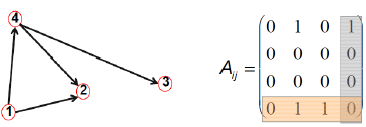


###### **figure1-8 :example de degrés de sommet**

{𝑨𝒊𝒋 = 𝑨𝒋𝒊

𝑨𝒊𝒊 = 𝟎}

1. **Graphe orienté**



matrice d’adjacence de grphe orienté

###### **Figure 1-9 : Example de degré de sommet** :

{𝑨𝒊𝒋 ≠ 𝑨𝒋𝒊

𝑨𝒊𝒊 = 𝟎

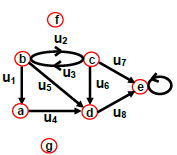
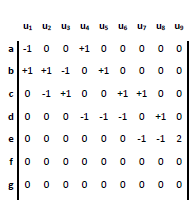
**L=**

### 1.6.2 Matrice d’incidence (sommet-arc) :

Soit le graphe G(X, U) suivant:

X = {a, b, c, d, e, f, g}

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



**A(N, M) figure1-10 :sommet matrice**

## Remarques:

* Chaque colonne dans la matrice contient un seul (+1) correspond à l'extrémité initiale de l'arc, et un seul (-1) correspond à son extrémité terminale
* Le nombre des (+1) sur la ligne donne le 1/2 degré extérieur du sommet, bien que le nombre des (-1) donne le 1/2 degré intérieur du même sommet.

# Chapitre02 : Notion de Flot

2.1 Introduction : Le flot est une notion très importante en théorie des graphes puisqu’elle permet de représenter des flux. De nombreux problèmes autour de ce concept ont été modélisés et étudiés, et par conséquent de nombreuses méthodes de résolution et d’importants résultats théoriques sont disponibles. Nous donnons quelques définitions et propriétés élémentaires sur les flots et tensions.[8]

## 2.2 Flots: définitions et propriétés

### 2.2.1 Définition :

Soit G = (X; U) un graphe connexe. X = {x1, x2,…..,xn} et U = {u1,u2,….,um} .

Un flot sur un graphe G est un vecteur ligne ϕ= (ϕ1, ϕ2,… ϕm) ∈ Rm à m composantes

qui affecte à chaque arc u un nombre ϕu tel que: en tout sommet i ∈ X de G, la première loi de Kirchhoff (loi de conservation aux nœuds) est vérifiée, c’est-à-dire :

**(1)**

Pour u ∈ U , la composante u du vecteur ϕ est appelée quantité de flot ou flux sur l’arc u.

**Remarque 2.2.1** La relation (1) exprime simplement que la somme des flux entrant en un sommet est égale à la somme des flux sortant de ce sommet.

* + 1. Définition algébrique des flots : Soit A = (aiuj)i=1…,n

J=1…,m

La matrice d’incidence sommet-arcs du graphe G = (X; U).

A chaque sommet i ∈X correspond la ligne i de A, on a :

**w + (i) = {u ∈ U/aiu = +1}**

**w-(i) = {u ∈ U/aiu = - 1}**

Les lois de conservation aux nœuds (1) peuvent donc se mettre sous la forme matricielle équivalente:

**∀i =1,…..,n ;ϕ=0⇔A.ϕ=0. (2)**

Si on considère l’application linéaire g associée à la matrice A : →, l’ensemble Φ des flots sur G constitue, d’après (2), le noyau de g. C’est donc un sous-espace vectoriel de , et sa dimension vérifie :

**dim (Φ)+rang(A)=m.**

Si G possède p composants connexes, on a :

**rang (A) = n – p**

Et par suite, la dimension de Φ est :

**dim(Φ)=m-n+p**

Qui n’est autre que le nombre cyclomatique de G noté ν(G) :

### Opérations sur les flots :

Soient ϕ,ϕ1,ϕ2des .flots sur G et k ∈R.

**Lemme 2.2.1 [9]**

* k.ϕ est un flot sur G.
* ϕ1 + ϕ2 est un flot sur G.
* ϕ1 - ϕ2 est un flot sur G.

**Théorème 2.2.1 :** Un flot sur un arbre est nécessairement identiquement nul. [10]

## 2.3 Tensions : définitions et propriétés

### 2.3.1 Définition :

Soit G = (X,U) un graphe orienté où X = {x1, x2,…. xn} et U = {u1,u2,…. um}.

Une tension sur un graphe G est un vecteur ϕ = (ϕ1,ϕ2,…, ϕm) ∈ Rm à m composantes tel que, pour tout cycle élémentaire μ , on a:

Cette égalité s’exprime aussi par le produit scalaire

**<u,θ>= .**

### 2.3.2 Propriétés sur les tensions:

**Définition 2.3.1** **:** On appelle potentiel dans un graphe G = (X,U) tout vecteur de Rn

C’est-à-dire toute fonction qui à chaque sommet associé un nombre réel.

**Théorème 2.3.1 :** Dans un graphe G = (X;U) un vecteur θ (θ1, θ2,…., θm) de Rmest une tension si et seulement s’il existe une fonction t ∈Rn tel que pout tout arc

u = (xi, xj) ∈U on ait:

**θu = t(xj) - t(xi)**

La fonction t est par définition un potentiel attaché à la tension θ.

**Opérations sur les tensions**

**Soient θ, θ1, θ2 ∈Φ où Φ désigne l’ensemble des tensions et λ∈ R:**

**Lemme 3.3.1 ( voir [9] )**

* **λ. θ est une tension.**
* **θ1 + θ2 est une tension.**

## 2.4 Le problème de flot de valeur maximum ou de cout minimum

### 2.4.1 Flot de valeur maximale

#### 2.4.1.1 Définitions

**Définition 1** : Un graphe fortement connexe, sans boucle et ayant plus d’un sommet est appelé un réseau.

**Définition2** :Soit G = (X,U) un 1-graphe orienté antisymétrique comportant n

sommets. Nous supposons qu’il existe dans X deux sommets particuliers x1 et xn tels que Ƭ-(x1) = θ et Ƭ+(xn) = θ, avec x1 est appelé sommet-entrée ou source noté x1 = s et xn est appelé sommet-sortie ou destination noté xn = t (t ≠ s). De plus, nous affectons chaque arc u = (xi, xj) ∈ U d’une quantité cu≥ 0 qui représente la capacité de cet arc.

Un tel graphe est appelé réseau avec capacité, il est défini par le triplet R = (X,U,C), où C = {cij , (xi; xj) ∈U}: On considère le graphe G0 = (X; U0) déduit de G en rajoutant un arc (t; s) dont les extrémités initiale et terminale sont respectivement t et s. L’arc (t,s) est appelé l’arc de retour du flot, et on convient de lui attribuer le numéro 0. Les arcs de G0 sont donc numérotés 0, 1,…,m.

**Définition3 :**  Pour un flot ϕ dans un réseau de transport R = (X,U,C), on dit qu’un

arc est saturé si on a:

**ϕu = cu.**

**Définition 4 :** Le flot est dit complet si tout chemin allant de s à t contient au moins

un arc saturé. (8)

### 2.4.2 Recherche d’un flot complet :

Soit R = (X; U;C) un réseau de transport, un flot ϕsur R est dit complet si pour tout chemin μde s à t, il existe un arc u ∈μtel que ϕu = cu.

* Autrement dit, tout chemin allant de s à t possède au moins un arc saturé par le flot.
* Un flot complet n’est pas nécessairement maximum.

**Construction d’un flot complet :**

* Poserϕ= 0.
* Répéter
  + Choisir un chemin μ tel que u ∈μ on a ϕu < Cu:
  + Si un tel chemin n’existe pas alors fin.
  + Sinon pour tout u ∈μ faire α = ,ϕ0= ϕ0+α

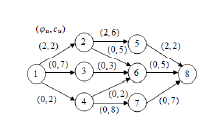
**Exemple 2.4.2**Déterminons un flot complet de 1 à 8 dans le réseau R = (X, U, C) ci-dessous. On commence par le flot nul ϕ= (0,0, 0, 0,0,0,0,0) de valeur ϕ0 = 0

###### 

###### **Figure2.1 : Exemple d’un flot complet**

**Etape 1:**

* μ = ((1, 2), (2, 5) , (5 ,8)) est un chemin dans R.
* α =
* ϕ12 = ϕ25 = ϕ58 = 2, ϕij = 0 ailleurs
* ϕ0= 2



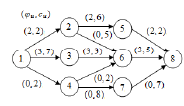
###### **Figure2.2 : étape 1**

**Etape 2:**

* μ = ((1, 3), (3,6), (6, 8)) est un chemin dans R.
* α = min(7, 3,5)=3

u∈μ

* ϕ13 = ϕ36 = ϕ68 = 3 , ϕ12 = ϕ25 = ϕ58 = 2, ϕij = 0 ailleurs.
* ϕ0 = 5.



###### **Figure2.3 :étape 2**

**Etape 3:**

* μ= ((1,4), (4,6), (6, 8)) est un chemin dans R.
* α= min(2,2,5)=2

u∈μ

* ϕ13 = ϕ36 = 3, ϕ68 = 5,ϕ12 = ϕ25 = ϕ58 = 2 , ϕij= 0 ailleurs.
* ϕ0 = 7.

###### 

###### **Figure2.4 : étape 3**

**Etape 4:**

* Il n y a pas de chemin de 1 à 8 dans R. Donc le flot ϕà l’étape 3 est complet et sa valeur est ϕ0 = 7. [8]

### 2.4.3 Position du problème

2.4.3.1 Définition:

On appelle Réseau de transport un graphe G = [X, U] ou chaque arc u∈U est muni d'un nombre Cu ≥ 0 appelé la capacité de l'arc u. Lorsqu'on fait circuler un flot sur G, ce nombre indique la limite supérieure du flux admissible sur l'arc u.

Dans la suite, et sauf mention contraire, les capacités Cu seront supposées être des nombres entiers ≥0.

Etant donnés deux sommets particuliers S ∈X (source) et t 2 X (puits) (S ≠ t)de G, on considéré le graphe G0 = [X, U0] déduit de G en rajoutant un arc (t, S) dont les extrémités initiale et terminale sont respectivement t et S. L'arc (t ,S) est appelé l'arc de retour du flot, et on convient de lui attribuer le numéro 0. Les arcs de G0 sont donc numérotés 0, 1, …., M.

On dit que le vecteur ϕ= [ϕ1, ϕ2 ,….ϕM ] T est un flot de S à t dans G, si et seulement si les lois de conservation aux nœuds sont vérifiées en tous les sommets de G sauf aux sommets S et t ou on a :

**= =ϕ0**

La quantité ϕ0 est appelée la valeur du flot.

On remarque que si ϕ = [ϕ1, ϕ2,….. ϕM] T est un flot de S à t dans G et de valeur ϕ0,

alors ϕ0 = [ϕ0, ϕ1, ….. ϕM] T est simplement un flot dans G0.

Le problème du flot maximum de S à t dans G muni des capacités Cu(u ∈ U), revient alors à déterminer un flot ϕ0 = [ϕ0,ϕ1,…. ϕM] dans G0 vérifiant les contraintes de capacité :

0 ≤ ϕu ≤ Cu (u = 1,…., M) et tel que la composante ϕ0 sur l'arc de retour (valeur du

flot) soit maximale. (12)

#### 2.4.3.2 Définition des coupes capacité d'une coupe:

On appelle coupe séparant S et t un ensemble d'arcs de la forme : W+(A) ou A X est un sous-ensemble de sommets tel que S ∈A et t ∉A.

On définit la capacité de la coupe W+(A) comme la somme des capacités des arcs qui la constituent :  **(12)**

2.4.3.3Le théorème du flot maximum et de la coupe minimale :

La valeur maximal d'un flot de s à t dans G = [X, U] muni des capacités Cu (u ∈ U) est égal à la capacité d'une coupe de capacité minimale séparant S et t. (13)

### 2.4.4 Algorithme de Ford et Fulkerson

#### 2.4.4.1 Principe de l’algorithme :

Une itération de l’algorithme de Ford et Fulkerson revient à choisir une chaîne joignant s et t, constituée d’arcs (x1, .) ,…., (xi, xj) ,…., (.., xn) tels que:

1. (xi , xj)∈U , avec rij=cij - ϕij > 0

ou

1. (xj , xi)∈U ,avec rij = ϕji > 0.

On peut associer au flot ϕ0 existant au début de chaque étape un graphe

G(ϕ) = (X, U(ϕ)), dit graphe d’écart, dont les arcs sont définis comme suit:

Pour tout (xi , xj) ∈ U

1. (xi,xj) ∈ U (ϕ) , si cij-ϕij >0 ;
2. (xj,xi) ∈ U (ϕ) , si ϕij >0 ;

**(8)**

#### 2.4.4.2 Organigramme de l’algorithme de Ford et Fulkerson :

R( X , U , C)

Flot réalisable initial ϕ de valeur ϕ0

Construire le graphe d’ecart

G (ϕ)-(X,U(ϕ)) ;(i,j) ∈U(ϕ) si :

1. (i,j)∈ U et Cij-ϕi >0
2. (j,i) ∈ U et ϕji > 0

Il existe un chemin de s à t dans G (ϕ)

Stop,le flot ϕ obtenu est un flot de valeur maximale

Non

Construire un tel chemin μ

rij =

α= min

ϕ0= ϕ0+α

ϕij=

###### **Figure 2.5 : algorithme de Ford et Fulkerson.**

### 2.4.5 Problème **du flot compatible**:

Définition ; Soit un graphe G = (X,U) connexe. A chaque arc u ∈ U de G on

affecte deux nombres bu et cu tels que: bu ≤ cu: Le problème est de trouver un flot ϕ dans G compatible avec les contraintes:

**bu ≤ ϕu ≤ cu ,u ∈U.**

Remarque : le flot nul n’est pas nécessairement compatible. (8)

#### 2.4.5.1 Une condition nécessaire d’existence :

Si ϕ est un flot dans G, alors pour tout sous-ensemble de sommets A  X

Si ϕ est un flot compatible c’est-à-dire bu ≤ ϕu ≤ cu, u ∈ U alors:

**(8)**

#### 2.4.5.2 Algorithme de recherche d’un flot compatible :

**Principe de l’algorithme :**

Soit ϕ un flot quelconque dans G, compatible ou non.

On appelle distance de ϕij à l’intervalle [bij , cij ] le nombre rij défini par:

Donc, pour un flot quelconque ϕ, on a:

**r= ≥0**

L’idée est de minimaliser r sur l’ensemble ϕ de tous les flots.

#### 2.4.5.3 Organigramme de l’algorithme :

R(X , U , C)

Flot réalisable initiale de valeur ϕ0

Construire le graphe d’écart

G(ϕ) = (X , U (ϕ)) , (i,j) ∈ U (ϕ) si :

bij ≤ϕij ≤cij

Stop,ϕ est un flot compatible

oui

Non

Il existe un arc u =(i ,j)

ϕij < b ij

Cif <ϕij

Il existe un chemin u de i à j dans G(ϕ)

Stop, il n’existe pas de flot compatible dans G ( ϕ)

Il existe un chemin u de j à i dans G (ϕ)

Non Non

oui

α= min ( )

ϕ=

α= min ( )

ϕ=

i

**figure2.6 : organigramme de l’algorithme.**

### 2.4.6 Flot de cout minimum ;

#### 2.4.6.1 Formulation du probléme :

Soit R = (X,U,C) un réseau avec capacité. Supposons qu’à chaque arc (i, j) ∈ U est associé un nombre réel fij qui représente le coût unitaire de mouvement de la matière le long de cet arc. Le coût total d’un flot réalisable ϕ, noté f(ϕ) est:

f(ϕ) =

Considérons l’ensemble des flots réalisables ϕ de même valeur Φ0, 0 ≤ Φ0 ≤ΦM où

ΦM = max(ϕ0 , ϕ∈Φ). Un flot de cet ensemble est dit de coût minimal s’il minimise le coût total f(ϕ). Nous présentons ci-dessous un algorithme permettant l’obtention d’un tel flot, pour une valeur Φ0 donnée. Cet algorithme est une extension directe de l’algorithme de Ford et Fulkerson.(8)

#### 2.4.6.2 Algorithme d’obtention d’un flot de cout minimum :

**Principe de l’algorithme :**

Rappelons que chaque étape de l’algorithme de Ford et Fulkerson consiste à rechercher un chemin μ de s à t dans le graphe d’écart G(ϕ): Pour minimiser les coûts de circulation de matière, il suffit donc de déterminer, à chaque itération un chemin de coût minimum.

En assimilant le coût unitaire fij sur l’arc (i, j) à une longueur Lij ; le problème revient à obtenir, à chaque étape, un chemin de longueur minimale, et cela jusqu’à ce que la valeur du flot ainsi construit atteigne le niveau Φ0: Remarquons qu.il convient généralement de choisir ici comme flot réalisable initial le flot de valeur nulle.(8)

# Chapitre 3 : Problème de réseau électrique

## 3.1Introduction

Le problème de réseau électrique constitue un champ d’application classique de la théorie des flots.

Ce problème peut être résolu par les méthodes générales exposées au chapitre précédant. L’objet de ce chapitre est de présenter un algorithme qui permet de résoudre très facilement un tel problème.

## 3.2. Le réseau électrique

Un réseau électrique est un ensemble d'infrastructures énergétiques plus ou moins disponibles permettant d'acheminer l'énergie électrique des centres de production vers les consommateurs d'électricité.

Un réseau électrique est un graphe dont les sommets sont des stations (générateurs,Transformateurs) électriques et deux stations sont liées si une ligne de haute tension

On a appliqué pour c'est réseau électrique le **problème de “Disjoint path problème »**

### 3.2.1 Disjoint path problème

Étant donné un digraphe G = (V, E) (le réseau électrique) et deux nœuds s et t (des stations), trouvez le maximum nombre de chemins s-t disjoints

**Théorème** : Le maximum flow est égal au nombre maximum de edge-disjoint paths.

On a Exécutez l'algorithme Ford-Fulkerson pour trouver le maximum flot de la source au puits.

## 3.3 Le problème de réseau électrique de la wilaya de M’sila

### 3.3.1 Le réseau considéré

**Données :**

Selon les données qu’on a récupéré de l’Internet concernant l’agence Algérie de réseau électrique et gaz de la wilaya de m’sila, on va prendre une partie du réseau électrique-m'sila.

### 3.3.2 Modélisation du problème

Le problème considéré consiste à la recherche des nombre maximum de réseau électrique entre les villes. Nous avons utilisé le réseau électrique de la wilaya de M’sila et nous pouvons modéliser le réseau sous forme d’un graphe orienté.

On va modéliser ce problème de réseaux sous forme d’un graphe orienté tandis que :

Les sommets sont : les villes de réseaux

Les arcs sont : le réseau entres les deux villes ou plus

Distance : lorsque on veut placer un réseau dans une zone, on choisit la région la plus proche et amène le réseau à partir de là.

**Les villes :**

1. M'sila

2. Awlademansore

3. Awlademadii

4. El mtarfa

5. El swamaa

6. Khatitisadeeljire

7. Termonte

8. El chlal

9. Msife

10. Awladedaraje

11. El maadhide

12. Sidi hajrese

13. Ben zouhe

14. Hamamedhalaa

15. El hiwamed

16. El khabana

17. Awladeaadi

18. Aiinhjel

19. Sidiaissa

20. Sidiamer

21. Wanora

22. Waltame

23. Ben srour

24. El maarife

25. El dhahna

26. Barhome

27. Aiinkhadra

28. Botisaihe

29. Bniyelmane

30. El zarzour

31. Awladeslaymen

32. Mouhamedboudiafe

33. Awladesidibrahime

34. Bousaada

35. Magra

36. Aiinfaresse

37. Sidimhamed

38. Tamsa

39. El hamel

40. Belaiba

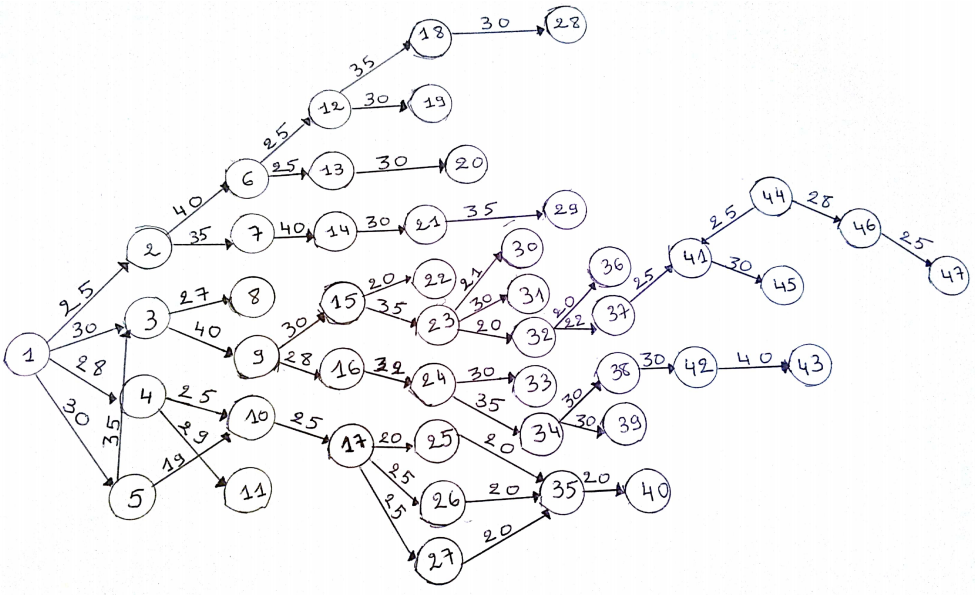
41. Aiinmelhe

42. Mannea

43. Mjadel

44. Jbalemsaade

45. Aiin riche



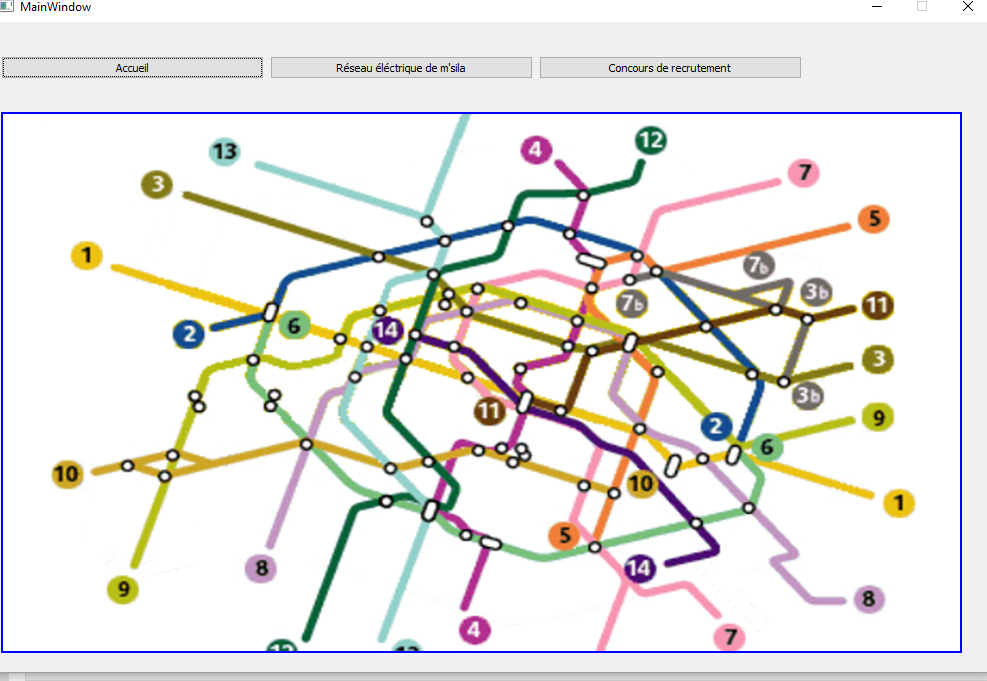
###### **Modélisation du réseau électrique sous forme d’un graphe orienté**

## 3.4 La résolution des problèmes et interprétation des résultats

### 3.4.1 Résolution

Dans le but de bien réaliser notre application qui intègre les algorithmes précédents, mais aussi de bien les implémenter on avait besoin d’utiliser un langage à la fois simple à manipuler, c’est le langage orienté objet python, il peut effectuer toutes les tâches d’un langage de haut niveau (graphique, bases des données, environnement de développement). On a opté pour la version python 3.8 et la distribution anaconda.

L’application que nous avons réalisée est composée d’une interface principale, illustrée dans la figure ci-dessous. Cette dernière est dotée de sous interfaces qui permettent de manipuler et de réaliser des opérations sur les graphes. Nous allons expliquer, à l’aide d’un exemple simple (de réseau électrique de la wilaya de M’sila et ensuite, le problème de Concours de recrutement)

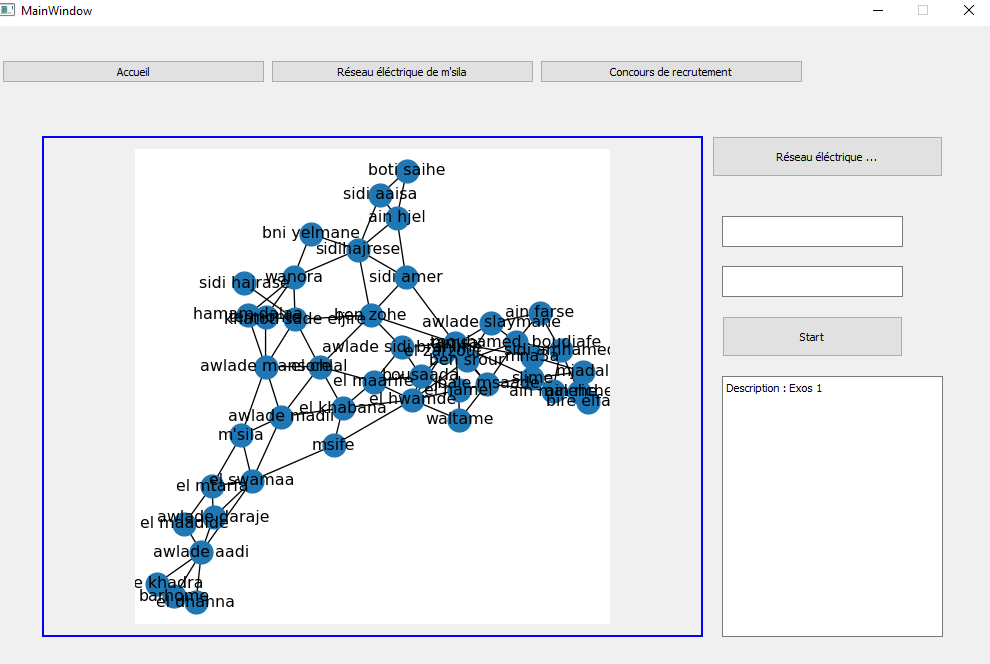


###### **Fig3.1 : Résolution du problème de réseau électrique de la wilaya de M’sila**

#### L’introduction des données :

Premièrement, on exécute MainWindow (Interface python)

Deuxièmement,on clique sur Réseau électrique de M’sila

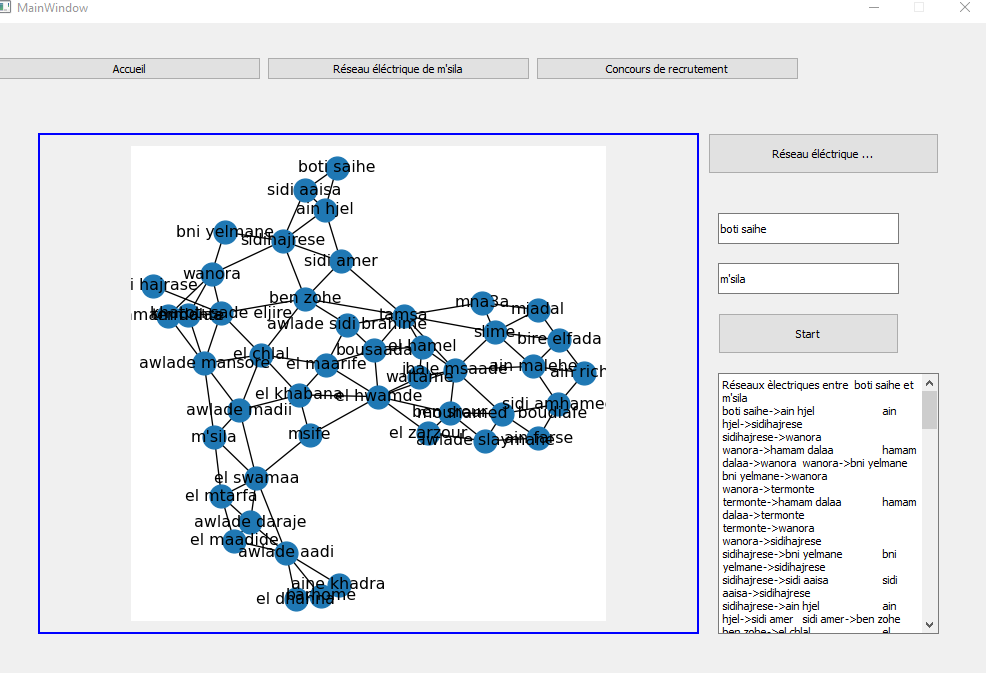


###### **Fig3.2 : l’accueil**

On entre dans les deux zones entre les quelles on veut rechercher des réseaux

On clique sur Start

Au final, tous les réseaux et leur numéro sont affichés



###### **Fig3.3 : Le résultat de la recherche de réseau électrique de la wilaya de M’sila entre Deux villes**

# Conclusion

Dans ce chapitre nous avons présenté le problème de réseau électrique de wilaya de m’sila nous avons modélisé ainsi que leurs résolutions à l’aide des méthodes vu dans le chapitre précédent, et pour faciliter la recherche des réseaux on a donné un programme sous le logiciel python qui nous permette de donner les solutions plus rapidement.

# Chapitre4 : Problème de Concours de recrutement.

## 4.1Introduction

On cherche toujours la réalisation d’une étude d’un système réel dans un environnement opérationnel. Cependant, on est amené à substituer le système par un modèle qui représentera son fonctionnement d’une manière plus au moins précise. Pour cela, on fait appel à des outils descriptifs, mathématiques ou autres soient-ils pour permettre de cerner au mieux le comportement du système. La conversion du système en un modèle se nomme modélisation. Selon les objectifs de l’étude, différentes représentations du système sont possibles. Dans ce contexte, la modélisation est un processus à priori, qui ne conduit que très rarement à un résultat unique. Dès à présent, nous soulignons le fait que l’outil de modélisation, de par sa structure, sera plus au moins bien adapté à un tel ou tel comportement. Dans ce chapitre, on va présenter situation concrète pour modéliser le problème sous forme d’un graphe afin d’optimiser certains paramètres en utilisant les algorithmes cités précédemment.

## 4.2 Problème de concours de recrutement

Dans le problème de concours de recrutement, Il y a un ensemble de candidats au concours et un ensemble d’emplois. Chaque candidat a un sous-ensemble d'emplois qui l'intéressent. Chaque ouverture de poste ne peut accepter qu'un seul candidat et un candidat peut être nommé pour un seul emploi. Nous affectation les emploisaux candidats de manière à ce que le plus de candidats possible obtenir des emplois qui peuvent être formés en tant que correspondance bipartite (Maximum Bipartite Matching)

### 4.2.1 Maximum Bipartite Matching

Un matching dans un graphe bipartite est un ensemble d'arêtes choisies de telle sorte qu'aucune deux arêtes ne partage une extrémité une maximum matching est une matching de taille maximum(nombre maximum d'arêtes).Dans une maximum matching, si une arête y est ajoutée, ce n'est plus une correspondance. Il peut y avoir plus d'une maximum matchings pour un graphe bipartite donné

### 4.2.2 Maximum Bipartite Matching and Max Flow Problem

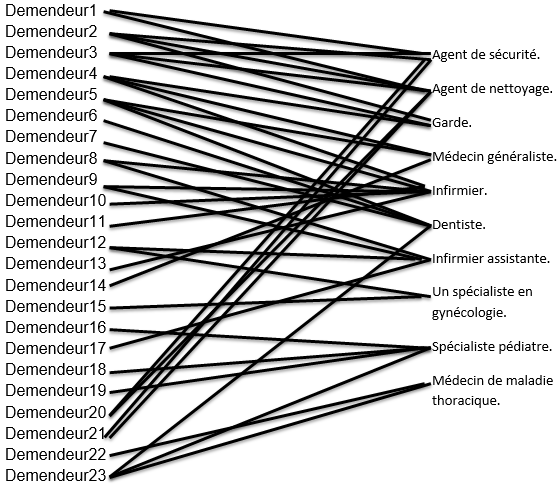
Le problème Maximum Bipartite Matching (MBP) peut être résolu en le convertissant en un réseau de flux. Nous utilisons l'algorithme de Ford-Fulkerson pour trouver le débit maximum Le débit maximum est en fait le MBP que nous recherchons.

## 4.3 Le problème de Concours de recrutement d’hôpital ben khnatha en Ben Srour

### 4.3.1 Modélisation du problème

Le problème considéré consiste à la recherche des d'emplois en hôpital de ben khnatha pour les candidats de manière à ce que le plus de candidats possible obtenir des emplois. On a modéliser c'est problème sous forme d’un graphe bipartite.

Le problème considéré consiste à la recherche de d'emplois en hôpital de ben khnatha pour les candidats de manière à ce que le plus de candidats possible obtenir des emplois. On a modélisé ce problème sous forme d’un graphe bipartite.

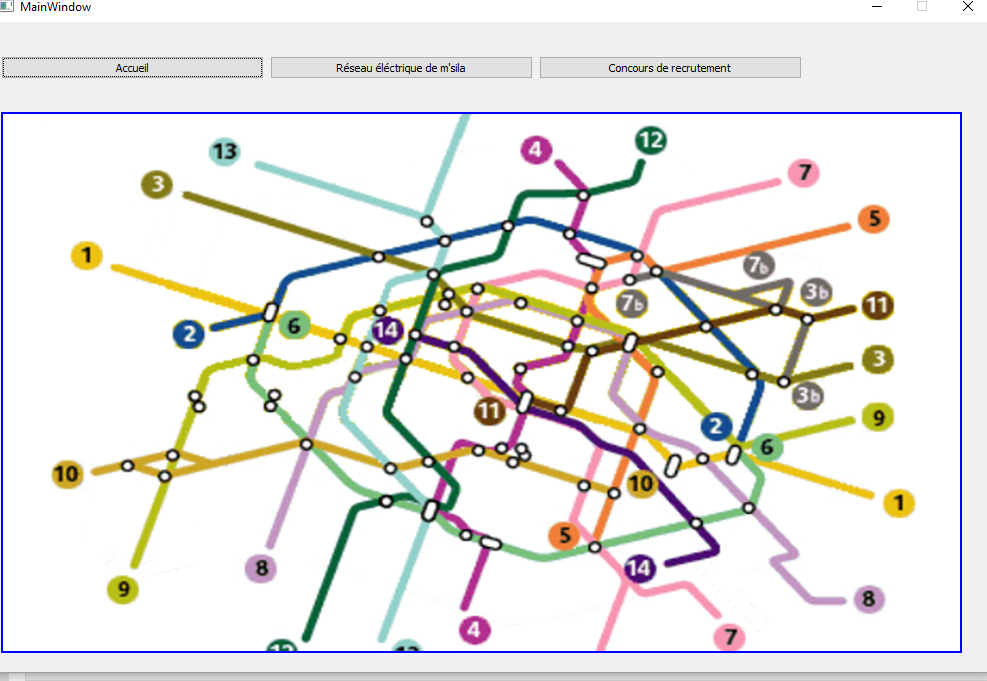


## Concours d’hôpital sous forme d’un graphe bipartite

## 4.3 La résolution des problèmes et interprétation des résultats

### 4.3.1 Résolution

L’application que nous avons réalisée est composée d’une interface principale, illustrée dans la figure ci-dessous. Cette dernière est dotée de sous interfaces qui permettent de manipuler et de réaliser des opérations sur les graphes. Nous allons expliquer, à l’aide d’un exemple simple (problème de Concours de recrutement)

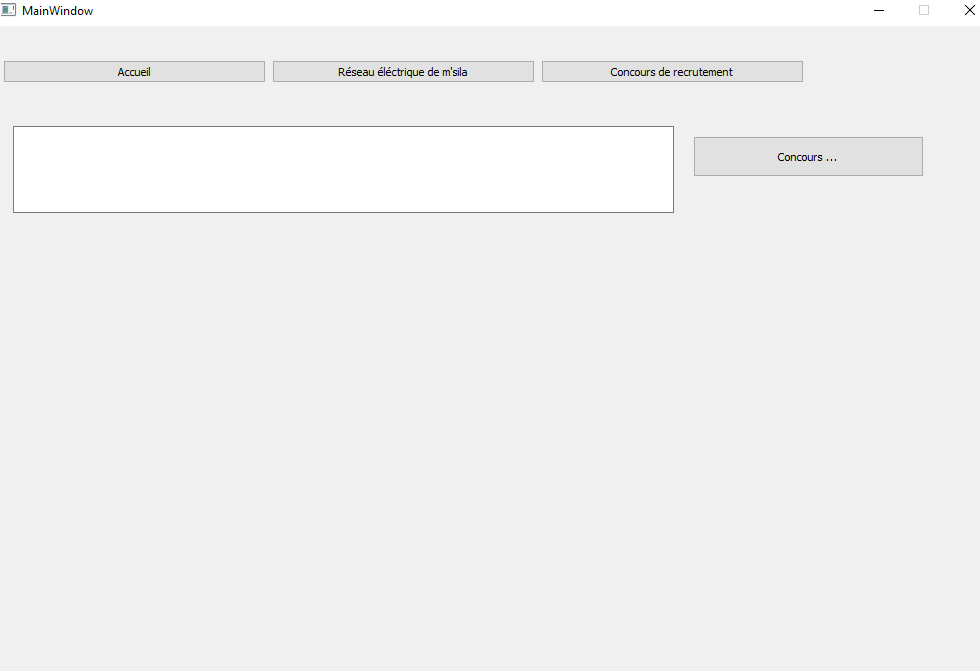


###### **Fig.4.1 : Résolution du problème de Concours de recrutement de hôpital ben khnatha en Ben Srour**

L’introduction des données :

Premièrement, on exécute MainWindow (Interface python)

Deuxièmement, nous cliquons sur Concours de recrutement



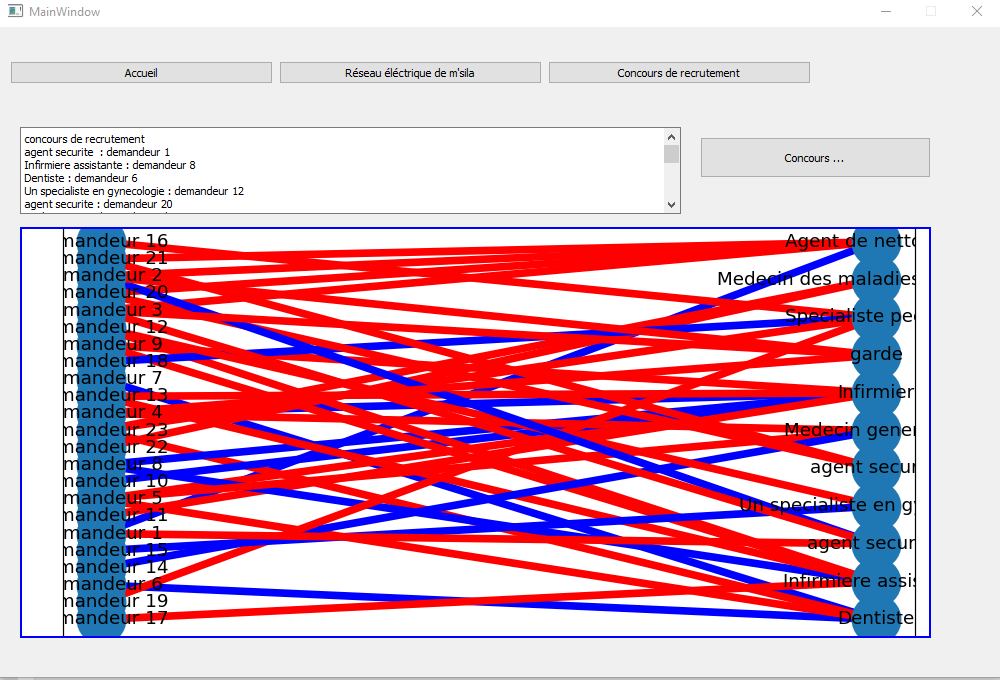
###### **Fig4.2 :l’accueil**

Apparaître cette interface sur laquelle nous cliquons sur Concours.

Apparaître Le résultat de cette Concours

## Conclusion

Dans ce chapitre nous avons présenté le problème du réseau électrique de problème de Concours de recrutement dans l’hôpital ben khnatha de Ben Srour. on a modélisé ainsi que leurs résolutions à l’aide des méthodes vu dans le chapitre précédent, et pour faciliter la recherche des résultats de ce concours on a donné un programme sous le logiciel python qui nous permette de donner les solutions plus rapidement.



###### **Fig4.3 : résultat**

# Conclusion générale

La théorie des graphes est un domaine très vaste ayant la fois un intérêt théorique et pratique au vu de ses applications dans des domaines aussi divers que variés. Dans ce mémoire nous nous somme intéressé au problème de flot et ses applications. Après un rappel des résultats principaux de la théorie des graphes, nous avons abordé le problème. Par la suite, nous nous somme intéressé au problème de flot maximum. Comme nous somme intéressé aussi au problème de flot de valeur donnée et de coût minimal. Nous avons terminé ce mémoire par deux applications des problèmes de flots coût minimal :Le problème de réseau électrique et un problème de Concours recrutement. Bien que le problème de flot soit très classique, il peut fournir des solutions efficaces a pas mal de problèmes en Algérie comme : l’assainissement, l’irrigation en agronomie, le problème de l’encombrement dans les villes, l’orientation des étudiants.

**LISTE DE REFERENCE**

[1] Bondy.A, et U.S.R. Murty , «Graph theory». 2008.

[2] kadri Saïd .les fondement de la théorie des graphes .chp1 :concepts de bases. université Mohammed boudiaf.msila,2017 /2018 .

[3] Bretto.A, A.Faisant, F.Hennecart. Éléments de théorie des graphes. 15/05/2012 .

[4] Amrouni.L,Azzegag.K. La recherche des points d’articulation dans un réseau. Master en recherche opérationnelle. Université Abderrahmane Mira Bejaia.2016-2017.79 Pages .

[5]Berge. Claude., « Graphes et hypergraphes », Dunod Editions, Paris 1970.

[6] Philippe. Compoint, "Les Graphes en recherche opérationnelle", Dunod Editions, Paris 1972.

[7] kadri Saïd .les fondement de la théorie des graphes .chp2 : représentation de graphes. université Mohammed boudiaf .m’sila,2017 /2018 .24pages .

[8] Khalfaoui.F.Ouchene.S. Problème de flot : application a un réseau de transport. Master en mathématique. Université Abderrahmane Mira.béjaia.2013.111pages.

[9] C. Berge, Graphe et hypergraphes, Bordas 1973.

[10]M. Gondran et M. Minoux, Graphes et algorithmes, Eyrolles, 1985.

[11] G. Royer, Graphes et applications, IUP MIAGE, 2003.

[12] Chougar.C.Elbey.F .Flots et tensions dans un réseau. Master en recherche opérationnelle .université Bejaia .2016-2017.65Pages.

[13 ] M. Michel Gondran, Michel Minoux. Graphes et algorithmes.61, Bd Saint-Germain Paris 5, 1985.