République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté de Mathématiques

Département de Recherche Opérationnelle

Rapport

Licence Recherche Opérationnelle

Thème

Problème du transversal minimum dans les graphes bipartis

Présenté par :

- BENHADDAD Sabrina
- HADJI Mohamed Oussama

Encadré par : Mr. KANTOUR Nedjmeddine

Année universitaire : 2020/2021

Remerciements

Nous tenons tout d'abord à diriger nos remerciements les plus sincères à tout enseignant du département RO pour nous avoir accompagné le long de notre cursus universitaire, en particulier en cette période difficile que nous traversons.

Nous souhaitons adresser nos remerciements dans un premier temps à notre encadreur Mr Nedjmeddine Kantour pour avoir toujours été là pour nous conseiller, orienter et assister. Son écoute ainsi que ses réponses à nos nombreuses interrogations que ce soit lors des sessions d'encadrements ou par mails, nous ont effectivement énormément aidé, sans oublier ses remarques nous poussant à présenter un travail toujours aussi qualitatif que abouti.
Nous remercions également Mr Yacine Chaiblaine, notre chargé de TP, qui n'a pas cessé de nous aider en langage Python, nous permettant de concevoir au mieux l'implémentation de notre problème.
Sans oublier notre chef de département Mr Amine Belkhir pour avoir toujours été à l'écoute, et fait preuve d'autant de soutien à de maintes occasions.

Et par la même occasion, une pensée et un grand merci à nos chers amis et familles respectives pour leur soutien inconditionnel, leur précieuse aide et leurs encouragements constants.

Table des matières

Table des figures

Introduction

La Recherche Opérationnelle se définit comme étant un ensemble de méthodes mathématiques, algorithmiques et de synthèse, permettant de prendre les meilleures décisions. Elle fournit par ailleurs des outils pour de simulation, rationalisation et enfin d'optimisation du fonctionnement des systèmes industriels et économiques [**2**]. Elle propose par conséquent, des modèles afin d'analyser des situations assez complexes et permet par la suite aux décideurs, de prendre une décision efficace.

Les apports de la Recherche Opérationnelle sont visibles dans plusieurs secteurs que ce soit au niveau de l'organisation des lignes de production, la planification des missions, l'optimisation des portefeuilles, mais aussi au niveau de la couverture des réseaux par exemple. Ses applications sont quant à elles, tournées autour de l'économie, la planification d'entreprise, la finance et enfin le marketing. On constate toutefois, un élargissement de son champ vers des domaines d'intérêts publics tels que la gestion des systèmes de santé ou de l'éducation [**1**].

L'ingénieur en Recherche Opérationnelle doit donc exploiter des connaissances provenant de sources variées telles que les mathématiques appliquées, l'informatique mais aussi l'économie et la sociologie en étudiant notamment le comportement d'une population.

Les mathématiques appliquées, étant la base d'une étude en Recherche Opérationelle, englobent plusieurs méthodes fondamentales de résolution à l'image de la programmation linéaire ou encore de la théorie des graphes.

La programmation linéaire est intimement liée au développement de la Recherche Opérationnelle, grâce à la création de la première méthode de résolution efficace : l'algorithme du simplexe en 1947 par G.Dantzig.
Son principe consiste à maximiser ou à minimiser une fonction linéaire sous des contraintes elles-aussi linéaires permettant en pratique, de modéliser un important nombre de problèmes réels.

L'autre méthode très importante aux côtés de la programmation linéaire est la théorie des graphes. Cette science élargit les possibilités de modélisation, puisqu'elle assure une représentation graphique de problèmes courants à l'instar du plus court chemin mais aussi, du flux de transport etc.

Le présent mémoire traitera un parfait exemple de la Recherche Opérationnelle alliant la théorie des graphes à la programmation linéaire. Il s'agit du problème transversal minimum dans les graphes bipartis.
Ce problème fera l'objet d'une étude complète à travers trois chapitres principaux :

Le premier chapitre portera sur les notions fondamentales de la programmation linéaire. Elles permettront par la suite d'introduire et de détailler davantage la méthode de résolution adoptée, à savoir : l'algorithme du simplexe.

Le deuxième chapitre quant à lui, traitera dans un premier temps de quelques bases de la théorie des graphes avant de se concentrer sur le problème, sa modélisation et enfin sa résolution.

Le dernier chapitre sera consacré pour finir, à l'implémentation du problème. L'objectif étant d'apporter un logiciel pratique de résolution, tout en explicitant son fonctionnement grâce à une application numérique détaillée.

Chapitre 1

Principes de la programmation linéaire et algorithme du simplexe

La programmation ou optimisation linéaire, est un domaine mathématique permettant de modéliser et résoudre un problème de minimisation ou de maximisation, tout en respectant un ensemble de contraintes.
L'objectif étant de déterminer une solution optimale à l'aide de méthodes ou algorithmes de résolution.

Un programme linéaire est donc un programme mathématique composé de variables de décision, où la fonction objectif ainsi que les contraintes sont linéaires.

1.1 Modélisation d'un problème en programme linéaire

La modélisation consiste à traduire un problème de la vie courante en un programme linéaire, tout en restant fidèle aux exigences imposées. Cette représentation repose donc sur l'identification des éléments suivants :

Les variables de décisions :

Ce sont les inconnus du PL dont on cherche à déterminer leur valeur, on les note comme suit :

$$x_i \ avec \ i = 1, ..., n$$

Les contraintes :

Les contraintes traduisent les exigences imposées, et déterminent l'ensemble des solutions réalisables. Elles prennent la forme d'inégalités ou d'égalités et sont toujours linéaires.

La fonction objectif :

C'est une fonction linéaire à optimiser, elle détermine la valeur de la solution optimale du problème.

1.2 Formulations d'un programme linéaire

Soient i, j, m et n des entiers.
Pour A une $(m \times n)$ matrice, b un vecteur de taille m et c un vecteur de taille n, considérons (P) un programme linéaire.

Forme générale d'un PL :

La forme générale consiste à optimiser la fonction objectif Z à l'aide de variables de décisions réelles $x = (x_1, x_2, ..., x_n)^t \in \mathbb{R}^n$. Les contraintes sont quant à elles de type $\geq, \leq$ *ou* $=$.

Forme canonique d'un PL :

Elle dépend de variables de décisions positives et de contraintes de type $\geq$ en cas de minimisaton de la fonction objectif (elles sont de type $\leq$ en cas de maximisation).

Forme standard d'un PL :

La forme standard consiste à optimiser une fonction objectif Z à l'aide de variables de décisions positives et des contraintes de type $=$.
Son écriture matricielle est comme suit :

$$(P_S) \begin{cases} Ax = b \\ Z(Opt) = cx \\ x \geq 0 \end{cases}$$

1.3 Notions de base et solution de base

Considérons un programme linéaire sous sa forme standard et A supposée de plein rang :

$$(P) : \begin{cases} Ax = b,\ x \geq 0 \\ Z(Opt) = cx \end{cases}$$

Base :

Une base de (P) ou du système linéarie $Ax = b$ est un ensemble d'indices $J \subseteq \{1, ..., n\}$ des colonnes de A, tel que A_J soit carrée régulière.
Les indices j (resp. les colonnes, les variables x_j) sont de base si $j \in J$ et sont hors base si $j \notin J$, autrement dit $j \in \overline{J}$.

Base réalisable et solution de base réalisable :

La solution x du système $Ax = b$ obtenue en posant $x_{\overline{J}} = 0$ est appelée solution de base associée à la base J.
Cette solution de base x_J est donc déterminée comme suit :

$$\begin{cases} x_{\overline{J}} = 0 \\ x_J = (A^J)^{-1}b \end{cases}$$

Une base J de (P) est dite réalisable si la solution de base associée est telle que $x_J \geq 0$.
Cette solution est dite réalisable.

Solution de base optimale :

Soit J une base réalisable de (P). Si le vecteur coût relatif à J est positif ou nul, alors la base est appelée base optimale et la solution de base associée à J est une solution de base optimale de (P).

1.3.1 Écriture sous forme canonique par rapport à une base :

On dit qu'un programme linéaire (P) est écrit sous forme canonique par rapport à une base J, si les deux conditions suivantes sont vérifiées :

— A^J est à une permutation près des colonnes, la matrice identité.

— $c_J = 0$ où c_J est appelé vecteur coût relatif à la base J [**4**].

1.4 L'algorithme du Simplexe

La méthode du simplexe est la plus célèbre méthode pour la résolution numérique des programmes linéaires.
Elle repose essentiellement sur deux phases distinctes :

1. La phase I : elle permet de déterminer une première solution de base réalisable (l'initialisation).
2. La phase II : c'est une procédure itérative qui consiste à trouver à partir d'une solution de base réalisable, la solution optimale grâce à l'algorithme du simplexe.

Notre étude se concentrera sur l'algorithme du simplexe présenté ci-dessous :

Algorithme du Simplexe

1. **Initialement :**

$$(P)\begin{cases} Ax = b \\ cx = min(z) - \alpha^*,\ x \geq 0 \end{cases}$$

Le PL est écrit SFC% à une base réalisable J.

Application col : $\{1, .., m\} \longrightarrow J$ donnée. Matrice des coefficients : M

$$\underbrace{x_{col(i)} = b_i\ (\forall i = \overline{1,m})\ et\ x_{\overline{J}} = 0}_{solution\,de\,base\,réalisable\,associée\,\grave{a}\,J\,(initiale)} \quad \underbrace{z(x) = \alpha *}_{\acute{e}valutation\,initiale}$$

2. **Choisir s tel que $c^s < 0$:**

— si s n'existe pas. Terminer. J base optimale. Poser $x^* = x$
x^* solution de base optimale associée à J ; $z(x) = \alpha^*$ évaluation optimale.

— sinon, aller à 3.

3. **Soit** $I = \{i | A_i^s > 0\}$:

— si $I = \emptyset$. Terminer. (P) n'a pas de solution optimale.

— sinon, aller à 4.

4. **Soit** $L = \left\{ l | \frac{b_l}{A_l^s} = \underset{i \in I}{Min} \left\{ \frac{b_i}{A_i^s} \right\} \right\}$; Choisir $r \in L$
effectuer $Pivotage(m + 1, n + 1, r, s; M) = M'$; $J' = (J \cup \{s\}) \setminus \{col(r)\}$ nouvelle base réalisable

$$x'_s = \frac{b_r}{A_r^s};\ x'_{col(i)} = b_i - A_i^s x'_s;\ x'_{\overline{J}} = 0;\ \alpha = \alpha^* + c^s \frac{b_r}{A_r^s} \text{ nouvelle évalution}$$

Poser $J = J'$, $col(r) = s$, $x = x'$, $\alpha^* = \alpha$; $M = M'$. Aller à 2.

Fin de l'algorithme

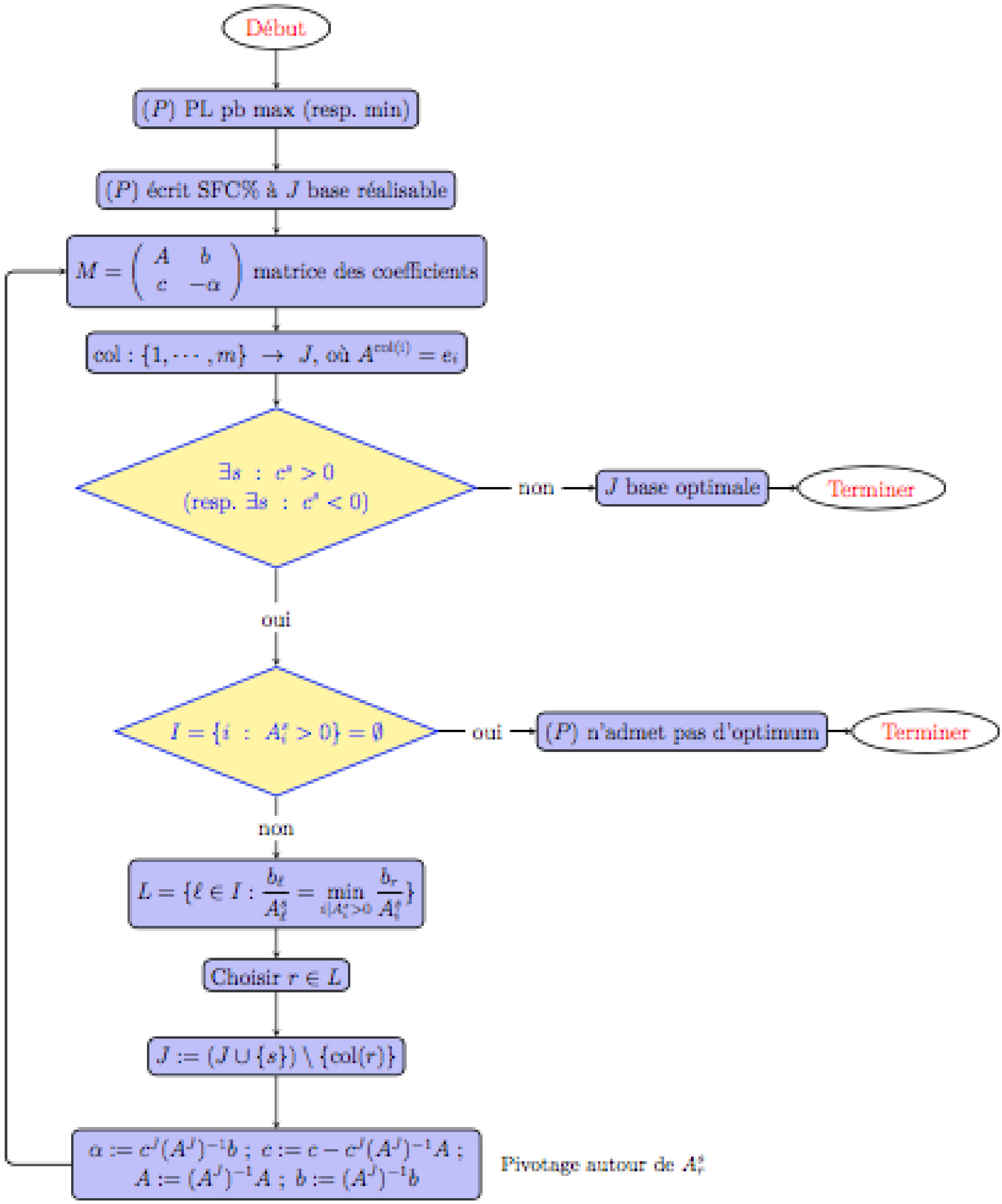

Figure 1.1 – Organigramme de l'algorithme du simplexe

Chapitre 2

Problème du transversal minimum dans les graphes bipartis

Le problème du transversal minimum, également nommé problème de couverture minimum par sommets ou encore "Minimum Vertex Cover Problem" en anglais, est un problème connu de la théorie des graphes et de l'optimisation combinatoire. On le retrouve ainsi dans de nombreux domaines d'application tels que la physique, la biochimie ou encore la mécanique.

Faisant parti des 21 problèmes NP-Complets de Karp, le problème du transversal minimum dans un graphe quelconque est plutôt difficile à résoudre. Nous nous intéresserons par conséquent, à une certaine catégorie de graphes facilitant ainsi, la résolution du problème à l'aide de l'algorithme du simplexe.
Il s'agit du problème du transversal minimum dans les graphes bipartis.

2.1 Concepts fondamentaux de la théorie des graphes

La théorie des graphes est un domaine des mathématiques introduit en 1736 quand Euler démontra qu'il était impossible de traverser chacun des sept ponts de Königsberg une seule et unique fois et de revenir au point de départ.
Elle représente un outil efficace de résolution des problèmes en Recherche Opérationnelle, grâce à sa modélisation à l'aide de graphes.

2.1.1 Généralités sur les graphes

Nous nous intéresserons dans tout ce qui suit, aux concepts non orientés.

Graphe

Un graphe non orienté $G = (X, E)$ est caractérisé par :

— Un ensemble de sommets $X = \{x_1, ..., x_n\}$ et un ensemble d'arêtes $E = \{e_1, ..., e_m\}$.

— $|X| = n$ est appelé ordre de G (i.e nombre de sommets).

— m représente la taille de G (i.e nombre des arêtes).

— Si $e = xy$ une arête de E, alors les sommets x et y sont ses extrémités et sont dites incidentes à e.

Adjacence

- Deux sommets x et y sont dit adjacents ou voisins, s'ils sont joints par une arête.
- Deux arêtes sont dites adjacentes si elles ont au moins une extrémité commune.

Graphe biparti

Un graphe est dit biparti si l'ensemble des sommets X peut être partitionné en deux sous-ensembles X_1 et X_2 tel que :

1. $X_1 \cap X_2 = \emptyset$
2. $X_1 \cup X_2 = X$

De sorte que tous deux sommets dans X_1 ou X_2 ne soient pas adjacents.

On note $G = (X_1 \cup X_2, E)$.

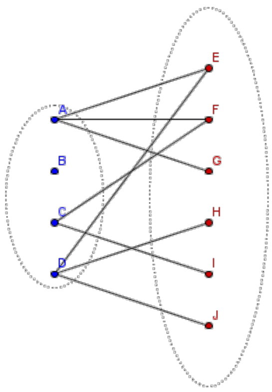

FIGURE 2.1 – Graphe biparti

2.1.2 Matrice d'incidence sommets-arêtes

Soit $G = (X, E)$ un graphe simple non orienté d'ordre n et de taille m.
On définit la matrice d'incidence sommets-arêtes $M = \underset{\forall i=1,..,n, \forall j=1,..,m}{(m_{i,j})}$ dont les lignes et les colonnes sont indicées respectivement par les sommets et les arêtes.

$$m_{i,j} = \begin{cases} 1 \ si \ e_j \ incidente \text{ à } x_i \\ 0 \ sinon \end{cases}$$

Matrice totalement unimodulaire

Une matrice $(n \times m)$ est dite totalement unimodulaire si et seulement si toutes les sous-matrices carrées extraites de la matrice M ont pour déterminant : $+1, -1$ ou 0

2.2 Transversal minimum dans les graphes bipartis : modélisation et résolution du problème

Nous considérons dans ce qui suit un graphe simple non orienté $G = (V, E)$.

Transversal

Le transversal du graphe G (ou recouvrement des arêtes par les sommets) est un sous-ensemble de sommets $T \subseteq V$ tel que toute arête de G ait au moins une de ses extrémités dans T, autrement dit :
Si $e = uv$ une arête de G avec u et v dans V, on distingue alors l'un de cas suivants :

— $u \in T$

— $v \in T$

— Les deux sommets sont dans T

Transversal minimum

Le transversal minimum est le plus petit transversal en terme de cardinalité, autrement dit :

$$T_0 \text{ transversal minimum} \Leftrightarrow |\mathrm{T}_0| = min\{|T|,\, T\ transversal\}$$

L'objectif de notre mémoire est de trouver donc, un sous-ensemble de sommets T_0 de cardinalité minimum, qui couvre la totalité des arêtes du graphe G, en passant par sa formulation en programme linéaire.

2.2.1 Modélisation du problème

Soit $G = (V, E)$ un graphe simple non orienté d'ordre n et de taille m, tel que :

— $V = \{v_1, v_2, ..., v_n\}$ ensemble de sommets de G.

— $E = \{e_1, e_2, ..., e_m\}$ l'ensemble des arêtes de G.

Variables de décisions :

Associons à chaque sommet v_i une variable de décision binaire x_i, $\forall i = 1, .., n$.

$$x_i = \begin{cases} 1\ si\ le\ sommet\ v_i\ appartient\ au\ transversal\ T_0 \\ 0\ sinon \end{cases}$$

Contraintes du PL :

D'après la définition d'un transversal, ce dernier doit obligatoirement couvrir toutes les arêtes $e = x_i x_j$ du graphe G par au moins une de ses extrémités, autrement dit soit l'une d'entre elles soit les deux seront saturées.
Les variables de décision étant binaires, nous distinguerons 4 cas :

1. $x_i = x_j = 0 \Rightarrow x_i + x_j = 0$:
   Alors l'arête $x_i x_j$ ne sera pas couverte, ce qui contredit la définition du transversal, ce cas est donc impossible.
2. $x_i = 1, x_j = 0 \Rightarrow x_i + x_j = 1$:
   Alors l'arête $x_i x_j$ sera couverte par un seul sommet x_i, donc $x_i \in T_0$.
3. $x_i = 0, x_j = 1 \Rightarrow x_i + x_j = 1$:
   Alors l'arête $x_i x_j$ sera couverte par un seul sommet x_j, donc $x_j \in T_0$.
4. $x_i = x_j = 1 \Rightarrow x_i + x_j = 2$:
   Alors l'arête $x_i x_j$ sera couverte par ses deux extrémités, donc x_i et x_j sont dans T_0.

Nous pouvons enfin déduire que la contrainte de couverture des arêtes s'écrit comme suit :

$$x_i + x_j \geq 1,\ \forall e = x_i x_j \in E$$

Fonction objectif :

L'objectif est de trouver le transversal minimum du graphe G, autrement dit minimiser le nombre de sommets T_0.
La fonction objectif s'écrit donc comme suit :

$$W(min) = \sum_{i=1}^{n} x_i$$

Formulation du PL :

Nous pouvons écrire notre PL comme suit :

$$(PL) \begin{cases} W(min) = \sum_{i=1}^{n} x_i \\ x_i + x_j \geq 1,\ \forall e = x_i x_j \in E \\ x_i \in \{0, 1\} \end{cases}$$

2.2.2 Écriture matricielle du PL

Soit la matrice d'incidence sommets-arêtes M du graphe G, tel que chaque ligne et colonne de la matrice par définition, représente respectivement un sommet et une arête.

Sa matrice transposée M^t va naturellement représenter chaque sommet (resp. arête) au niveau des colonnes (resp. lignes). Ce qui modélise parfaitement la contrainte de couverture des arêtes d'où l'écriture suivante :

$$M^t X \geq e$$

Où :

— $X = (x_1, x_2, ..., x_n)^t$ représente le vecteur des variables de décisions.

— $e = {}^t(1, 1, ..., 1)$

Remarques :

Nous étudions le problème du transversal minimum dans un graphe biparti, sa matrice d'incidence sommets-arêtes est alors, totalement unimodulaire.
De plus, on sait que :

— M totalement unimodulaire $\Leftrightarrow$ Sa transposée M^t est totalement unimodulaire.

Nous pouvons donc appliquer le théorème suivant :

Théorème de Hoffman-Kruskal :

Si la matrice des contraintes d'un PL est totalement unimodulaire et le second membre b (dans notre cas e) est à composantes entières, alors les solutions de base de ce PL, sont à composantes entières. Nous pouvons donc, écrire les variables de décisions sous la forme $0 \leq x_i \leq 1$. Ce qui justifie pour finir, l'application de l'algorithme du simplexe pour la résolution du problème.
L'écriture matricielle du PL est donc de cette forme :

$$(P) \begin{cases} W(min) = \sum_{i=1}^{n} x_i \\ M^t X \geq e \\ 0 \leq x_i \leq 1 \ , i = 1, .., n \end{cases}$$

Pour appliquer l'algorithme du simplexe, nous considérons la matrice A représentant toutes les contraintes du PL, autrement dit :

— La contrainte de couverture des arêtes exprimée par M^t

— la nouvelle contrainte : $x_i \leq 1 \ , i = 1, .., n$

2.3 Exemple d'application

Soit le graphe simple non orienté et biparti $G = (X, E)$, d'ordre 7 et de taille 8 avec :

— $X = \{A, B, C, D, E, F, G\}$

— $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$

Représenté comme suit :

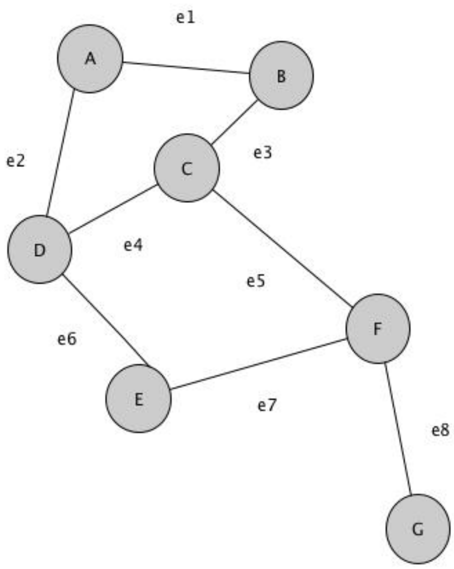

FIGURE 2.2 – Graphe associé à l'exemple

La recherche du transversal minimum T_0 dans G, passera par les étapes suivantes.

2.3.1 Modélisation de l'exemple

Variables de décisions :

Associons à chaque sommet dans X, la variable de décision binaire, tel que : $\forall i = 1, .., 7$

$$x_i = \begin{cases} 1 \; si \; le \; sommet \; i \; appartient \; au \; transversal \; T_0 \\ 0 \; sinon \end{cases}$$

Avec :

— $i = 1$ représente le sommet A.

— $i = 2$ représente le sommet B.

— $i = 3$ représente le sommet C.

— $i = 4$ représente le sommet D.

— $i = 5$ représente le sommet E.

— $i = 6$ représente le sommet F.

— $i = 7$ représente le sommet G.

Contraintes :

Les contraintes de notre exemple s'écrivent comme suit :

$$\begin{cases} x_1 + x_2 \geq 1 \\ x_1 + x_4 \geq 1 \\ x_2 + x_3 \geq 1 \\ x_3 + x_4 \geq 1 \\ x_3 + x_6 \geq 1 \\ x_4 + x_5 \geq 1 \\ x_5 + x_6 \geq 1 \\ x_6 + x_7 \geq 1 \end{cases}$$

Fonction objectif :

$$W(min) = \sum_{i=1}^{7} x_i$$

2.3.2 Résolution de l'exemple

Le PL peut s'écrire, d'après les sections précédentes, comme suit :

$$(PL) \begin{cases} W(min) = \sum_{i=1}^{7} x_i \\ M^t X \geq e \\ 0 \leq x_i \leq 1 \, , i = 1, .., 7 \end{cases}$$

Avec M la matrice d'incidence sommets-arêtes suivante :

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

Puisque le graphe G est biparti (il ne contient aucun cycle impair), alors M est totalement unimodulaire et donc sa transposée M^t l'est aussi, tel que :

$$M^t = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Puisque la matrice associée aux contraintes de couverture des arêtes M^t est totalement unimodulaire alors l'algorithme du simplexe s'applique (en considérant la matrice A de plein rang, englobant toutes les contraintes).
Après avoir résolu le PL par un solver de l'algorithme du simplexe (PHPSimplex), on obtient le vecteur :

$$X^* = (0, 1, 0, 1, 0, 1, 0)^t$$

Ce qui correspond à la solution optimale du problème où $W^* = 3$ est la valeur optimale de la fonction objectif.

Interprétation des résultats :

Nous pouvons facilement déduire d'après les résultats trouvés plus haut, les conclusions suivantes :

— $W^* = 3 \Rightarrow |T| = 3$: le transversal minimum est de cardinalité 3.

— ${}^tX^* = (0, 1, 0, 1, 0, 1, 0) \Rightarrow T_0 = \{B, D, F\}$: ce qui veut dire que les sommets B, D et F appartiennent au transversal minimum.

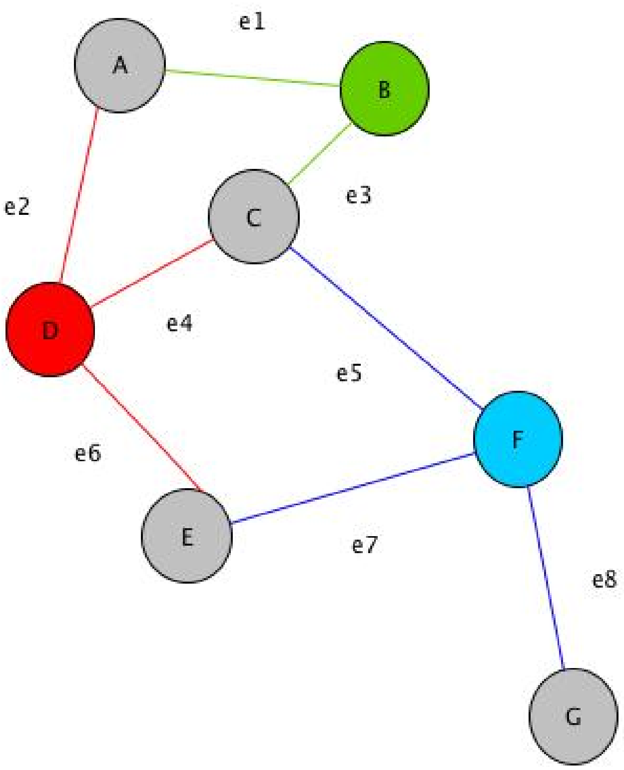

Figure 2.3 – Transversal minimum associé à l'exemple

Chapitre 3

Implémentation du problème du transversal minimum

Les langages de programmation sont un excellent intermédiaire entre le langage humain et le langage ordinateur. Ces derniers assurent effectivement, une bonne communication avec les machines grâce aux formulations algorithmiques, composées d'un ensemble d'instructions destinées à réaliser une succession de tâches précises.
Les langages de programmation sont donc indispensables dans le développement informatique, ce qui explique la conception de nombreux langages à l'image de Python, C++ ou encore Java.

Ce chapitre sera donc axé sur l'approche informatique de la résolution du transversal minimum dans les graphes bipartis, grâce à son implémentation en langage Python.

3.1 Présentation générale des outils de conception et de l'application

3.1.1 Le Langage Python

Python est le langage de programmation open source le plus employé par les professionnels en informatique et développement. Faisant parti des langages les plus faciles en pratique, développer du code avec Python est beaucoup plus rapide et surtout accessible à tous les niveaux y compris les débutants.
Les utilisations de Python sont nombreuses et très variées, il est effectivement possible grâce aux bibiliothèques disponibles gratuitement de :

— Programmer des applications.

— Créer des services Web.

— Analyser des données.

— Développer toute sorte de logiciels.

Mais la praticité de Python ne s'arrête pas là, puisque'il est également possible de créer des interfaces graphiques à l'aide de son module "Tkinter", qui nous sera d'une grande aide quant à l'implémentation de notre problème.

3.1.2 Introduction à l'application

L'application, conçue sous Python 3.8, consiste à résoudre le problème du transversal minimum dans un graphe biparti. Elle se compose d'une fenêtre divisée en deux parties essentielles pour la résolution, à savoir : L'insertion des données et l'affichage de la solution.

La saisie des données

La première partie de l'application consiste à introduire toutes les informations relatives au graphe traité. L'utilisateur doit donc insérer le nombre de sommets de ce graphe, le nombre de ses arêtes et enfin, sa matrice d'incidence sommets-arêtes.

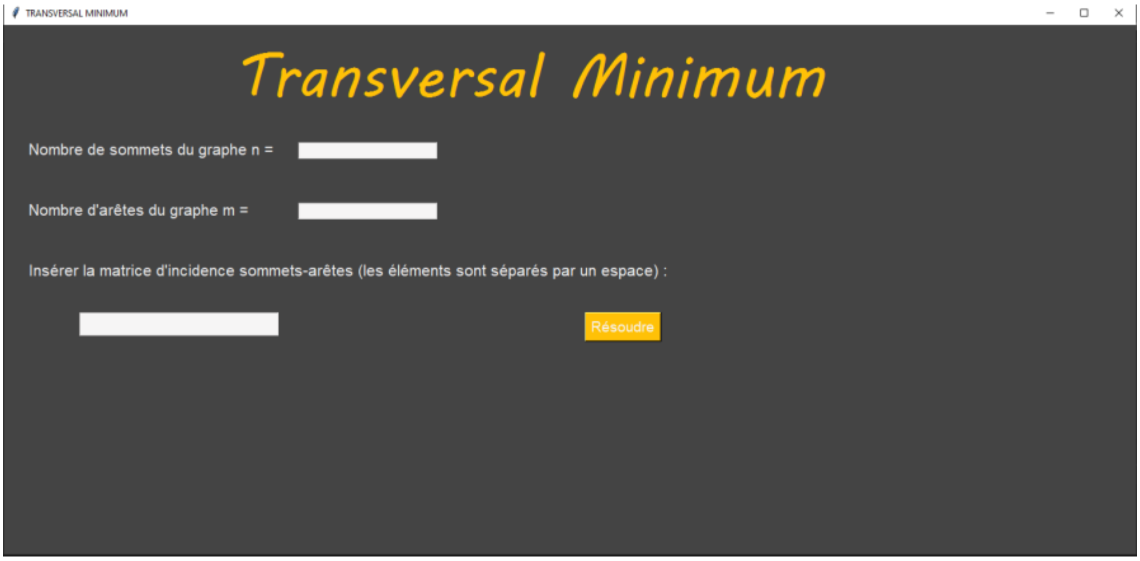

FIGURE 3.1 – Introduction à l'application : saisie des données

L'application va par la suite, vérifier toutes les données insérées avant de passer à la deuxième partie de l'implémentation.

L'affichage de la solution

La deuxième partie de la résolution du problème consiste à afficher la solution. Pour ce faire, il suffit d'appuyer sur la touche "Résoudre" pour obtenir par la suite, l'affichage des sommets appartenant au transversal minimum, son cardinal et enfin, le graphe associé à la solution.

3.2 Manuel d'utilisation de l'application

Le présent manuel, permet d'avoir une vue d'ensemble sur l'application ainsi que sur ses fonctionnalités. Ce dernier guidera les utilisateurs pas à pas, à aboutir à la solution de leur problème : le transversal minimum dans un graphe biparti.
Les seuls pré-dispositions à avoir sont :

- Un ordinateur (son système d'exploitation n'est pas important).
- La bonne version du langage Python : ici le 3.8 au minimum.

— L'installation de bibiliothèques importantes lors de la résolution du problème à l'image de PULP.

Nous allons dérouler dans tout ce qui suit, l'exemple vu dans le deuxième chapitre afin d'expliquer davantage la procédure à suivre.

3.2.1 Manipulation de l'application

La fenêtre de démarrage constitue une plateforme pratique et simple dans laquelle l'utilisateur doit introduire un nombre de données. (Voir Figure 3.1)

Insertion du nombre de sommets et arêtes :

L'utilisateur doit dans un premier temps, introduire le nombre de sommets n de son graphe, ainsi que le nombre de ses arêtes m à l'image de l'insertion de l'exemple du chapitre 2 suivante :

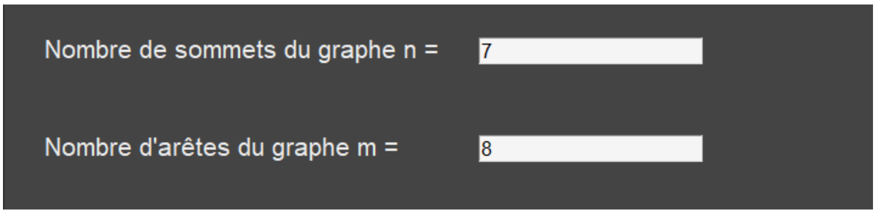

FIGURE 3.2 – Champ de saisie n°1

Un message d'erreur s'affichera en cas d'une mauvaise insertion comme suit :

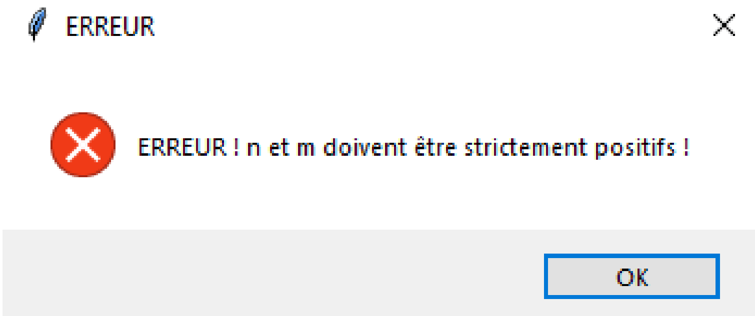

FIGURE 3.3 – Message d'erreur n°1

Insertion de la matrice d'incidence sommets-arêtes :

L'utilisateur doit par la suite introduire la matrice d'incidence sommets-arêtes ligne par ligne. Ses éléments doivent être séparés par des espaces.
La matrice d'incidence sommets-arêtes représentée dans notre cas, sera celle proposée dans le chapitre 2 :

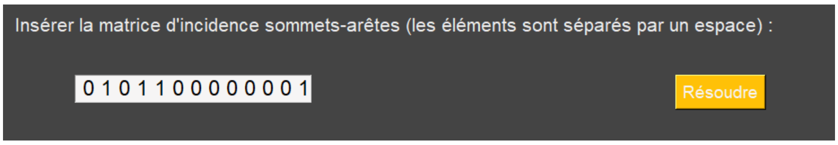

FIGURE 3.4 – Champ de saisie n°2

Une insertion non adaptée de cette matrice, empêchera la résolution du problème et affichera un message selon le type d'erreur :

— En cas d'une insertion ne respectant pas les tailles introduites :

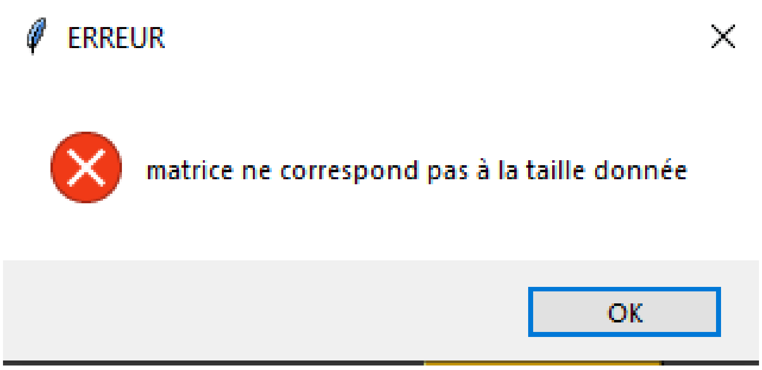

FIGURE 3.5 – Message d'erreur n°2

— En cas d'une insertion ne respectant pas la définition d'une matrice d'incidence sommets-arêtes :

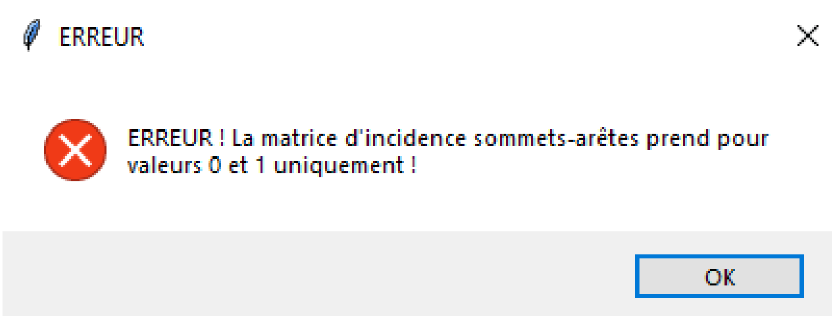

FIGURE 3.6 – Message d'erreur n°3

Dans le cas d'une insertion correcte, l'utilisateur n'a plus qu'à appuyer sur le bouton "Résoudre" pour afficher la solution.

Affichage de la solution :

Une fois l'intégralité des données insérées, l'application passera au traitement de ces informations pour distinguer si oui ou non le problème admet une solution.
Nous distinguerons donc, deux possibilités :
- Si la matrice n'est pas totalement unimodulaire (G n'est donc pas biparti), alors l'application informera l'utilisateur que le problème ne se résoud pas comme suit :

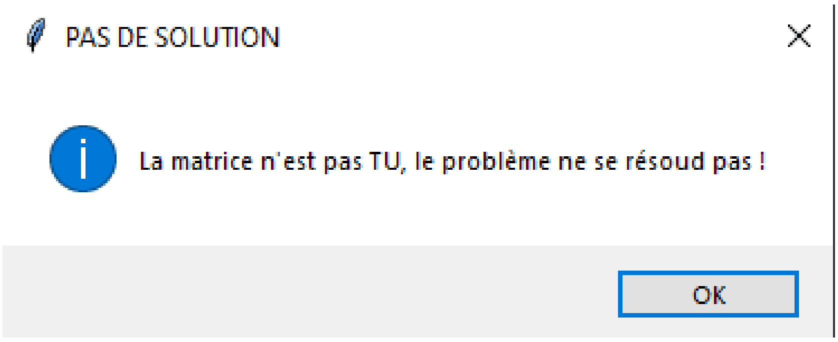

FIGURE 3.7 – Affichage en cas de matrice non TU

- Dans le cas contraire, l'application affichera le transversal minimum trouvé, son cardinal et enfin, son graphe associé comme suit :

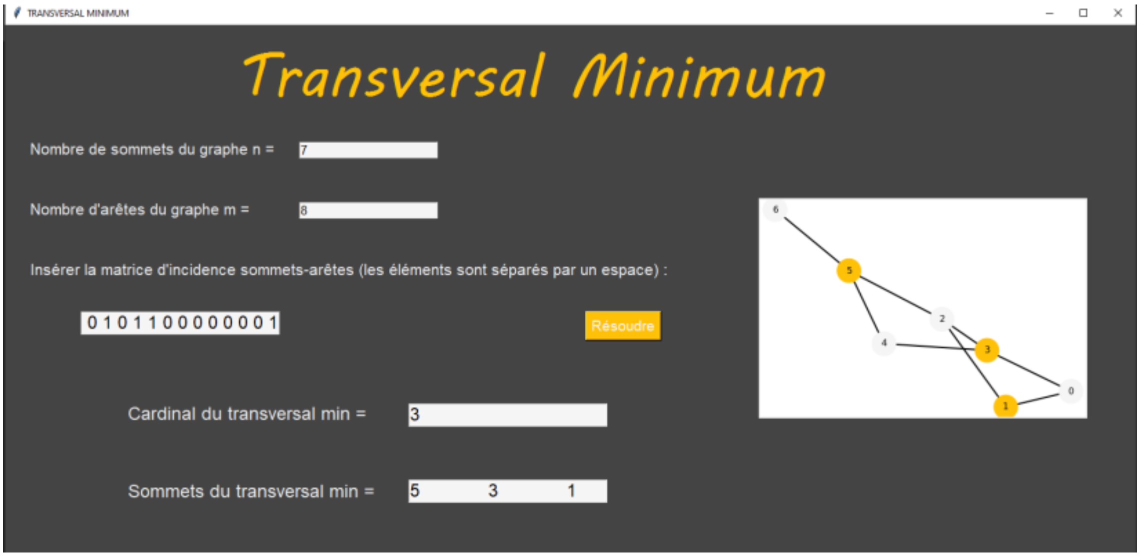

FIGURE 3.8 – Affichage de la solution associée à l'exemple vu au chapitre 2

Conclusion générale

Nous nous sommes intéressés dans ce mémoire, à un problème d'optimisation dans les graphes : le tranversal minimum dans les graphes bipartis.

L'objectif de ce travail portait sur la résolution de ce problème à l'aide de la méthode du simplexe, l'algorithme de résolution des problèmes d'optimisation linéaire le plus uti- lisé et le plus efficace, cela en passant par la théorie des graphes.

Dans ce sens, nous avons dans un premier temps, exposé quelques concepts de base de la programmation linéaire, et en particulier la méthode du Simplexe.
Nous avons ensuite traité ce problème par les deux outils que sont la programmation linéaire et les graphes.
Nous avons enfin, présenté une application conçue en langage Python, permettant la résolution de ce problème.

Ce travail nous aura montré qu'un problème d'optimisation dans les graphes, pour être résolu, peut être transformé en problème de programmation linéaire.
Ce qui nous laisse confirmer que la théorie des graphes, au vu de ses applications dans des domaines aussi divers que variés, est bel et bien un très vaste domaine ayant à la fois un intérêt théorique et pratique.

Bibliographie

[1] F.Meunier "Introduction à la recherche opérationnelle", Université Paris Est, CERMICS, Ecole des Ponts Paristech, 2016.

[2] ROADEF "Le LIVRE BLANC de la recherche opérationnelle en France", 2019.

[3] R.Faure, B.Lemaire et C.Picouleau : " Précis de recherche opérationnelle, méthodes et exercices d'application", 7ème édition, DUNOD, 2014.

[4] M. Sakarovitch, Optimisation combinatoire - Graphes et programmation lineaire, Hermann, 1984.

[5] ORF363 lectures notes. http ://www.aaa.princeton.edu/orf363. Accessed : 2016-02-26.

[6] P.Lopez "Cours de GRAPHES", LAAS–CNRS, 2 avril 2008.

[7] K.Biswas, S.Harun "Constraint Minimum Vertex Cover in K-Partite Graph : Approximation Algorithm and Complexity Analysis", 2009.

[8] D.Sonika "A new approximation algorithm for vertex cover problem", USICT, 2013.