

TSP et programmation linéaire

J.F. Scheid

2 novembre 2024

Table des matières

1	Introduction	1
2	TSP asymétrique	1
2.1	Formulation MTZ	2
2.1.1	Introduction	2
2.1.2	Variante	4
2.1.3	Mise sous formes matricielles	5
2.2	Formulation DFJ	7
2.2.1	Forme matricielle	8
2.3	Formulation par flot	9
2.3.1	Formes matricielles pour GG.	10
2.4	Résultats numériques avec données non-symétriques	11
3	TSP symétrique	11
3.1	Formulation	11
3.2	Résultats numériques : comparaison des formulations avec données symétriques	12
	Références	13

1 Introduction

Dans cette courte note, on présente et compare différentes formulations de programmation linéaire en nombres entiers pour le problème du voyageur de commerce (TSP). Les deux versions *asymétrique* et *symétrique* du TSP sont considérées. Pour le TSP asymétrique, on s'intéresse à 3 formulations : DFJ (élimination itérative des sous-circuits), TMZ et formulation par flot (GG). Pour la version symétrique du TSP, on considère une version symétrique de DFJ, cette dernière se révélant particulièrement efficace pour le TSP symétrique.

2 TSP asymétrique

Les distances entre les villes ne sont pas symétriques : la distance pour aller de la ville i à la ville j n'est pas la même que la distance pour aller de j à i . Il faut donc tenir compte du sens de parcours (a priori dans la formulation). Soit n villes (qu'on appellera aussi sommets pour faire référence au graphe associé au TSP) et on note d_{ij} la distance séparant la ville i de la ville j . A priori, on a ici $d_{ij} \neq d_{ji}$. On introduit les variables d'affectation x_{ij} pour $i, j \in \llbracket 1, n \rrbracket$, $i \neq j$, telles que

$$x_{ij} = \begin{cases} 1 & \text{si on va de la ville } i \text{ à la ville } j \\ 0 & \text{sinon.} \end{cases}$$

L'objectif est de minimiser la distance totale parcourue en passant une et une seule fois par toutes les villes (chemin hamiltonien) :

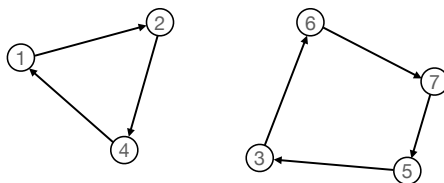
$$\min \sum_{\substack{i,j=1 \\ i \neq j}}^n d_{ij} x_{ij} \quad (1)$$

Pour traduire le fait que chaque ville possède un seul trajet *sortant* et *rentrant* (on doit passer une et une seule fois par chaque ville), on écrit les contraintes :

$$\sum_{\substack{j \\ j \neq i}} x_{ij} = 1, \quad \forall i \in \llbracket 1, n \rrbracket \quad (2)$$

$$\sum_{\substack{i \\ i \neq j}} x_{ij} = 1, \quad \forall j \in \llbracket 1, n \rrbracket \quad (3)$$

Ces contraintes ne sont pas suffisantes pour empêcher les sous-circuits, comme le montre l'exemple suivant.



Dans cet exemple, les contraintes (2),(3) sont bien réalisées mais la solution réalisable présente deux sous-circuits. Les ensembles des sommets (villes) du graphe constituant ces sous-circuits forment des composantes connexes (strictes) du chemin hamiltonien. On évite ces sous-circuits, soit en introduisant des contraintes supplémentaires sur les variables x_{ij} , soit en ajoutant des variables supplémentaires et des contraintes associées.

2.1 Formulation MTZ

2.1.1 Introduction

Pour éviter les sous-circuits, on introduit les variables supplémentaires u_i pour $1 \leq i \leq n$, représentant la position de la ville i dans la tournée du voyageur (dans le chemin hamiltonien). On part toujours de la ville 1, on a

$$u_1 = 1 \quad \text{et} \quad u_i \in \{2, \dots, n\}, \quad i \in \llbracket 2, n \rrbracket.$$

Par exemple, avec $n = 6$, pour le trajet $(1, 4, 6, 3, 5, 2, 1)$, on a $u_2 = 6, u_3 = 4, \dots, u_6 = 3$.

Le problème de voyageur de commerce (TSP) dans sa formulation **MTZ** (appelée par la suite **MTZ1** par rapport à ses variantes) s'écrit de la façon suivante :

$$\min \sum_{\substack{i,j=1 \\ i \neq j}}^n d_{ij} x_{ij} \quad (4)$$

$$\sum_{\substack{j \\ j \neq i}} x_{ij} = 1, \quad \forall i \quad (5)$$

$$\sum_{\substack{i \\ i \neq j}} x_{ij} = 1, \quad \forall j \quad (6)$$

$$u_i - u_j + 1 \leq (n - 1)(1 - x_{ij}), \quad \forall i, j, \quad i \neq 1, \quad j \neq 1, \quad i \neq j \quad (7)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \llbracket 1, n \rrbracket, \quad i \neq j \quad (8)$$

$$u_i \in \{2, \dots, n\}, \quad \forall i \in \llbracket 2, n \rrbracket. \quad (9)$$

Remarques.

- Les inconnues sont les variables binaires/entières x_{ij} ($i, j \in \llbracket 1, n \rrbracket$, $i \neq j$) et u_i ($i \in \llbracket 2, n \rrbracket$). Il y a donc au total $n(n-1) + n - 1 = n^2 - 1$ inconnues.
- Il y a $2n + (n-1)(n-2)$ contraintes.
- Les données "diagonales" d_{ii} n'interviennent pas.
- La contrainte (7) est obtenue de la façon suivante. Si $x_{ij} = 1$ alors nécessairement $u_i < u_j$. Cette condition se traduit par

$$u_i - u_j + 1 \leq \alpha(1 - x_{ij}) \quad (10)$$

où α est un réel. Pour déterminer α , on considère le cas où $x_{ij} = 0$. Dans ce cas, on ne sait pas si $u_i < u_j$ ou si $u_i > u_j$. Ce que l'on sait, compte tenu de (9), c'est que $2 - n + 1 \leq u_i - u_j + 1 \leq n - 2 + 1 = n - 1$. Pour que (10) soit vraie dans tous les cas (i.e. pour $x_{ij} = 1$ et $x_{ij} = 0$), on choisit $\alpha = n - 1$, d'où la condition (7).

L'introduction des variables u_i empêche bien les *sous-circuits* (appelés aussi *sous-tours*) dans la tournée. Un sous-circuit noté $\mathcal{S} = (i_1, i_2, \dots, i_k, i_1)$ est un chemin partant de la ville i_1 , passant par les villes i_2, \dots, i_k et revenant en i_1 (on a donc $x_{i_1, i_2} = x_{i_2, i_3} = \dots = x_{i_{k-1}, i_k} = x_{i_k, i_1} = 1$) avec $k < n$ i.e. on ne passe pas par toutes les villes.

Proposition 1 *Il ne peut exister qu'un seul circuit partant de la ville 1, passant par toutes les villes et revenant à la ville 1 de départ.*

Preuve. Il suffit de montrer que s'il existe un sous-circuit alors nécessairement celui-ci doit passer par la ville de départ n°1¹. On suppose donc qu'il existe un sous-circuit $\mathcal{S} = (i_1, i_2, \dots, i_k, i_1)$ qui ne passe pas par la ville n°1 i.e. avec $i_j \neq 1$ pour tout $j \in \llbracket 1, k \rrbracket$. Pour tous ces indices, on peut alors utiliser la contrainte (7) et compte tenu du fait que $x_{i_1, i_2} = x_{i_2, i_3} = \dots = x_{i_{k-1}, i_k} = x_{i_k, i_1} = 1$, on obtient :

$$\begin{aligned} u_{i_1} - u_{i_2} + 1 &\leq 0 \\ u_{i_2} - u_{i_3} + 1 &\leq 0 \\ &\dots \\ u_{i_{k-1}} - u_{i_k} + 1 &\leq 0 \\ u_{i_k} - u_{i_1} + 1 &\leq 0 \end{aligned}$$

En faisant la somme de toutes ces inégalités, on obtient $k \leq 0$, ce qui est impossible. □

On a aussi une autre propriété.

Lemme 1 *Toute solution réalisable satisfaisant les contraintes (5)–(9) vérifie*

$$x_{ij} + x_{ji} \leq 1, \quad \forall i \neq 1, j \neq 1, i \neq j.$$

Preuve. Pour tous $i \neq 1, j \neq 1, i \neq j$, on a d'après (7),

$$\begin{aligned} u_i - u_j + (n-1)x_{ij} &\leq n-2, \\ u_j - u_i + (n-1)x_{ji} &\leq n-2. \end{aligned}$$

En faisant la somme de ces deux inégalités, on obtient $(n-1)(x_{ij} + x_{ji}) \leq 2(n-2)$. On en déduit nécessairement que $x_{ij} + x_{ji} \leq 1$ (supposer $x_{ij} + x_{ji} \geq 2$ conduit à une contradiction). □

La propriété du Lemme 1 assure que si $x_{ij} = 1$ i.e. l'arête (i, j) appartient au trajet, alors nécessairement $x_{ji} = 0$ i.e. l'arête inverse (j, i) ne peut appartenir au trajet.

1. Ce sous-circuit passera nécessairement par toutes les villes, car sinon il y aurait un autre sous-circuit ne passant pas par la ville 1, d'où la contradiction.

2.1.2 Variante

On peut écrire le problème (4)–(9) en incluant les termes pour $i = j$. On dira que les exclusions du type $i \neq j$ sont relaxées. On dispose alors des variables x_{ij} pour $i, j \in \llbracket 1, n \rrbracket$ et toujours les u_i pour $i \in \llbracket 1, n \rrbracket$:

$$\min \sum_{i,j=1}^n d_{ij}x_{ij} \quad (11)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad \forall i \quad (12)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j \quad (13)$$

$$u_i - u_j + 1 \leq (n-1)(1-x_{ij}), \quad \forall i, j, i \neq 1, j \neq 1, \quad (14)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \llbracket 1, n \rrbracket, \quad (15)$$

$$u_i \in \{2, \dots, n\}, \quad \forall i \in \llbracket 2, n \rrbracket. \quad (16)$$

Il y a donc $n^2 + n - 1$ variables (au lieu de $n^2 + 1$ pour la formulation avec $i \neq j$). L'intérêt d'inclure le cas $i = j$ est d'avoir une formulation matricielle un peu plus simple (cf. section suivante).

Même en relaxant les exclusions, les données "diagonales" d_{ii} n'interviennent pas car on peut montrer que $x_{ii} = 0$ pour tout i . Ceci vient du fait d'avoir écrit la contrainte (7) aussi pour $i = j$. On a les propriétés suivantes :

Lemme 2 *Toute solution réalisable satisfaisant les contraintes (12)–(16) vérifie*

$$(a) \quad x_{ij} + x_{ji} \leq 1, \quad \forall i \neq 1, j \neq 1.$$

$$(b) \quad x_{ii} = 0 \text{ pour tout } i \in \llbracket 1, n \rrbracket.$$

Preuve. (a) Démonstration identique au Lemme 1.

(b) Cette propriété vient du fait qu'on a écrit la contrainte (14) aussi pour $i = j$. Pour tout $i \neq 1$, on prend $j = i$ dans (a), ce qui donne $2x_{ii} \leq 1$ d'où on déduit $x_{ii} = 0$.

La preuve pour montrer aussi que $x_{11} = 0$ est moins directe. Commençons par rappeler que si $x_{ij} = 1$ alors $u_i < u_j$ pour $i \neq 1, j \neq 1$ d'après (14). On raisonne à présent par l'absurde en supposant que $x_{11} = 1$. Alors compte tenu de (12) et (13), on a $x_{1j} = 0$ pour tout $j \neq 1$ et $x_{i1} = 0$ pour tout $i \neq 1$. Ainsi, on a la propriété

$$\forall i \neq 1, \exists j \neq 1, j \neq i \text{ tel que } x_{ij} = 1 \text{ et donc } u_i < u_j. \quad (17)$$

Soit $k_1 \neq 1$ (on peut prendre $k_1 = 2$). D'après (17) :

$$\begin{aligned} &\exists k_2 \in \llbracket 2, n \rrbracket, k_2 \neq k_1 \text{ tel que } u_{k_1} < u_{k_2} \\ &\exists k_3 \in \llbracket 2, n \rrbracket, k_3 \neq k_2 \text{ tel que } u_{k_2} < u_{k_3} \\ &\quad \vdots \\ &\exists k_n \in \llbracket 2, n \rrbracket, k_n \neq k_{n-1} \text{ tel que } u_{k_{n-1}} < u_{k_n} \end{aligned} \quad (18)$$

On a donc n entiers k_1, k_2, \dots, k_n qui appartiennent tous à $\llbracket 2, n \rrbracket$ donc parmi ces n entiers il y en a au moins 2 qui sont égaux : $\exists i, j \in \llbracket 1, n \rrbracket, i < j$ tels que $k_i = k_j$. Ainsi, d'après (18)

$$u_{k_i} < \dots < u_{k_j} = u_{k_i}$$

d'où la contradiction. □

2.1.3 Mise sous formes matricielles

i) Variante MTZ2

On commence par donner la forme matricielle de la variante (11)–(16) précédente dans laquelle les exclusions $i \neq j$ sont relaxées. Le problème (11)–(16) s'écrit sous forme matricielle :

$$\min_{\mathbf{x}, \mathbf{u}} \mathbf{d}^\top \mathbf{x} \quad (19)$$

$$A\mathbf{x} = \mathbf{b}_1 \quad (20)$$

$$B\mathbf{x} - C\mathbf{u} \leq \mathbf{b}_2 \quad (21)$$

$$\mathbf{x} \in \{0, 1\}^{n^2}, \quad \mathbf{u} \in \{2, \dots, n\}^{n-1} \quad (22)$$

Les variables (inconnues) sont regroupées *ligne par ligne* dans les vecteurs

$$\begin{aligned} \mathbf{x} &= (x_{11}, x_{12}, \dots, x_{1n} \mid x_{21}, x_{22}, \dots, x_{2n} \mid \dots \mid x_{n1}, x_{n2}, \dots, x_{nn})^\top \in \{0, 1\}^{n^2} \\ \mathbf{u} &= (u_2, \dots, u_n)^\top \in \mathbb{R}^{n-1} \end{aligned} \quad (23)$$

Il y a donc au total $(n^2 + n - 1)$ variables. De plus, on a :

$$\mathbf{d} = (d_{11}, d_{12}, \dots, d_{1n} \mid d_{21}, d_{22}, \dots, d_{2n} \mid \dots \mid d_{n1}, d_{n2}, \dots, d_{nn})^\top \in \mathbb{R}^{n^2} \quad (24)$$

$$\mathbf{b}_1 = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{2n}, \quad \mathbf{b}_2 = (n-2) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{(n-1)^2}. \quad (25)$$

— La matrice A est de taille $2n \times n^2$ avec

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \quad (26)$$

avec les matrices A_1 et A_2 de taille $n \times n^2$ données par

$$A_1 = \left(\begin{array}{ccc|ccc|ccc} 1 & \dots & 1 & 0 & & & 0 & & \\ & 0 & 1 & \dots & 1 & & \vdots & & \\ & \vdots & & & & \ddots & 0 & & \\ & 0 & & 0 & & & 1 & \dots & 1 \end{array} \right) \Bigg\}_n \quad (27)$$

$$A_2 = \left(\begin{array}{ccc|ccc|ccc} 1 & & & 1 & & & & & \\ & \ddots & & & \ddots & & & & \\ & & & & & \dots & & & \\ & & & & & & 1 & \ddots & \\ & & & & & & & & 1 \end{array} \right) \Bigg\}_n \quad (28)$$

— La matrice B est de taille $(n-1)^2 \times n^2$ et vaut

$$B = \left(\begin{array}{c|c|c|c|c} 0 & J_n & 0 & \dots & 0 \\ \hline 0 & 0 & J_n & \dots & 0 \\ \hline \vdots & & \ddots & \ddots & \\ \hline 0 & \dots & \dots & 0 & J_n \end{array} \right) \Bigg\}_{n-1} \quad (29)$$

$\underbrace{\hspace{1.5cm}}_n \quad \underbrace{\hspace{1.5cm}}_n \quad \underbrace{\hspace{1.5cm}}_n \quad \underbrace{\hspace{1.5cm}}_n$
 $(n-1) \text{ blocks}$

La matrice J_n est de taille $(n-1) \times n$ et elle est donnée par

$$J_n = \left(\begin{array}{c|cc} 0 & (n-1) & 0 \\ \vdots & & \ddots \\ 0 & 0 & (n-1) \end{array} \right). \quad (30)$$

— Enfin, la matrice C est de taille $(n-1)^2 \times (n-1)$ et elle est donnée par

$$C = \left(\begin{array}{c} C_1 \\ C_2 \\ \vdots \\ C_{n-1} \end{array} \right). \quad (31)$$

Les matrices C_j sont de taille $(n-1) \times (n-1)$ et valent

$$C_1 = \left(\begin{array}{c|cc} 0 & 0 & 0 \\ -1 & 1 & \\ \vdots & & \ddots \\ -1 & & 1 \end{array} \right), \quad C_j = \left(\begin{array}{ccc|ccc} 1 & & & -1 & & \\ & \ddots & & \vdots & & 0 \\ & & 1 & -1 & & \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & -1 & 1 & \\ & 0 & & \vdots & & \ddots \\ & & & -1 & & 1 \end{array} \right) \leftarrow j\text{-ième ligne} \quad (32)$$

\uparrow
 $j\text{-ième colonne}$

Remarque. Si on range les inconnues x_{ij} *colonne par colonne* (et non plus ligne par ligne comme en (23)), avec $\mathbf{x} = (x_{11}, x_{21}, \dots, x_{n1} | x_{12}, x_{22}, \dots, x_{n2} | \dots | x_{1n}, x_{2n}, \dots, x_{nn})^\top \in \{0, 1\}^{n^2}$, les matrices sont modifiées de la façon suivante :

- la matrice A est inchangée (on permute A_1 et A_2), la matrice B aussi.
- la matrice C change de signe : la contrainte (35) devient $B\mathbf{x} + C\mathbf{u} \leq \mathbf{b}_2$ (car on permute i et j dans (14)).

ii) MTZ1

On donne à présent la forme matricielle de la formulation initiale MTZ1 (4)–(9) avec les exclusions $i \neq j$. Le problème (4)–(9) s'écrit sous forme matricielle :

$$\min_{\mathbf{x}, \mathbf{u}} \mathbf{d}^\top \mathbf{x} \quad (33)$$

$$A' \mathbf{x} = \mathbf{b}_1 \quad (34)$$

$$B' \mathbf{x} - C' \mathbf{u} \leq \mathbf{b}_2 \quad (35)$$

$$\mathbf{x} \in \{0, 1\}^{n(n-1)}, \quad \mathbf{u} \in \{2, \dots, n\}^{n-1} \quad (36)$$

Les variables (inconnues) sont regroupées dans les vecteurs

$$\begin{aligned} \mathbf{x} &= (x_{12}, x_{13}, \dots, x_{1n} | x_{21}, x_{23}, \dots, x_{2n} | \dots | x_{n1}, \dots, x_{n(n-1)})^\top \in \{0, 1\}^{n(n-1)} \\ \mathbf{u} &= (u_2, \dots, u_n)^\top \in \mathbb{R}^{n-1} \end{aligned} \quad (37)$$

De plus, on a :

$$\mathbf{d} = (d_{12}, d_{13}, \dots, d_{1n} | d_{21}, d_{23}, \dots, d_{2n} | \dots | d_{n1}, \dots, d_{n(n-1)})^\top \in \mathbb{R}^{n(n-1)} \quad (38)$$

$$\mathbf{b}_1 = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{2n}, \quad \mathbf{b}_2 = (n-2) \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{(n-1)(n-2)}. \quad (39)$$

— La matrice A' est de taille $2n \times n(n-1)$ avec

$$A' = \begin{pmatrix} A'_1 \\ A'_2 \end{pmatrix} \quad (40)$$

où A'_1 et A'_2 sont de taille $n \times n(n-1)$, obtenues à partir des matrices A_1 et A_2 données par (27),(28) en supprimant les colonnes j pour $j \in \{1, n+1, 2n+2, \dots, n^2\} = \{1+k(n+1), k \in \llbracket 0, n-1 \rrbracket\}$ afin de ne pas prendre en compte les quantités $x_{11}, x_{22}, \dots, x_{nn}$ dans (23).

— La matrice B' est de taille $(n-1)(n-2) \times n(n-1)$ et vaut

$$B' = \begin{pmatrix} 0 & J_n & 0 & \dots & 0 \\ 0 & 0 & J_n & \dots & 0 \\ \vdots & & \ddots & \ddots & \\ 0 & \dots & \dots & 0 & J_n \end{pmatrix} \begin{matrix} \} n-2 \\ \} n-2 \\ \vdots \\ \} n-2 \end{matrix} \quad (41)$$

$\underbrace{\qquad\qquad\qquad}_{(n-1) \text{ blocks}}$

La matrice J_n est de taille $(n-2) \times (n-1)$ et elle est donnée par

$$J_n = \begin{pmatrix} 0 & (n-1) & 0 \\ \vdots & & \ddots \\ 0 & 0 & (n-1) \end{pmatrix}. \quad (42)$$

— La matrice C' est de taille $(n-1)(n-2) \times (n-1)$ et elle est donnée par

$$C' = \begin{pmatrix} C'_1 \\ C'_2 \\ \vdots \\ C'_{n-1} \end{pmatrix}. \quad (43)$$

Les matrices C'_j sont de taille $(n-2) \times (n-1)$ et sont obtenues à partir des matrices C_j données par (32) en supprimant la ligne j .

2.2 Formulation DFJ

Cette formulation est due à Dantzig, Fulkerson, Johnson (1954) [1]. On reprend la formulation (4),(5), (6),(8) en ajoutant des contraintes supplémentaires sur les variables x_{ij} pour éliminer les sous-tours mais sans ajouter de variable supplémentaire. On notera V l'ensemble des villes (indices, par exemple $V = \llbracket 1, n \rrbracket$) et $A = \{(i, j); i, j \in V \text{ avec } i \neq j\}$ l'ensemble des arêtes *orientées*. On associe de cette façon le graphe *orienté* $G = (V, A)$ au TSP asymétrique. La formulation s'écrit :

$$\min \sum_{(i,j) \in A} d_{ij} x_{ij} \quad (44)$$

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij} = 1, \quad \forall i \in V \quad (45)$$

$$\sum_{\substack{i \in V \\ i \neq j}} x_{ij} = 1, \quad \forall j \in V \quad (46)$$

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1, \quad \forall S \subsetneq V, |S| \geq 2 \quad (47)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V, i \neq j \quad (48)$$

La contrainte (47) permet d'éviter les sous-tours. En effet, s'il existe un sous-circuit $\mathcal{S} = (i_1, i_2, \dots, i_k, i_1)$ avec $1 < k < n$ alors avec $S = \mathcal{S} \subsetneq V$, on a $\sum_{j \in V \setminus S} x_{ij} = 0$ ce qui contredit (47).

On peut écrire la contrainte (47) de façon équivalente, sous une forme plus simple à utiliser dans la pratique.

Lemme 3 *Sous la contrainte (45), la contrainte (46) est équivalente à*

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ij} \leq |S| - 1, \quad \forall S \subsetneq V, |S| \geq 2 \quad (49)$$

Preuve. Soit $S \subsetneq V$, $|S| \geq 2$ et $i \in S$. On scinde en deux la somme dans (45) pour obtenir :

$$\sum_{\substack{j \in S \\ j \neq i}} x_{ij} = 1 - \sum_{j \in V \setminus S} x_{ij} \quad (50)$$

On somme alors sur $i \in S$:

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ij} = |S| - \sum_{i \in S} \sum_{j \in V \setminus S} x_{ij}.$$

La contrainte (47) donne bien (49). \square

Cette formulation possède $2^{n-1} + 2(n-1)$ contraintes et $n(n-1)$ variables binaires. Le nombre exponentiel de contraintes rend difficile (impossible) leur résolution directe. Dans la pratique, on résout sans les contraintes (47) (ou (49)). Quand on rencontre un ou des sous-circuits, on ajoute la contrainte (49) sur chacune des composantes connexes (ayant plus de 2 sommets) du graphe correspondant. Et on recommence de façon itérative jusqu'à obtenir un trajet sans sous-circuits. A l'itération k , si on rencontre un sous-circuit, correspondant à une composante connexe $S \subsetneq V$ du graphe, constitué d'un ensemble d'arête A_k , on n'impose pas exactement la contrainte (49) mais plutôt la contrainte suivante :

$$\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i \\ (i,j) \in A_k}} x_{ij} \leq |S| - 1. \quad (51)$$

Autrement dit, on prend en compte uniquement les arêtes constituant le sous-circuit et non pas l'ensemble des arêtes de la composante connexe S . Les résultats numériques (non présentés ici) ont montré que les contraintes (51) induisent le même nombre d'itérations dans la procédure d'élimination que les contraintes (49) mais que la résolution du PL est beaucoup plus rapide.

2.2.1 Forme matricielle

Sous forme matricielle avec les variables $\mathbf{x} \in \{0, 1\}^{n(n-1)}$ rangées selon (37), la formulation DFJ asymétrique s'écrit :

$$\min_{\mathbf{x}, \mathbf{u}} \mathbf{d}^\top \mathbf{x} \quad (52)$$

$$A' \mathbf{x} = \mathbf{b}_1 \quad (53)$$

$$\boldsymbol{\alpha}_S^\top \mathbf{x} \leq |S| - 1 \quad (54)$$

$$\mathbf{x} \in \{0, 1\}^{n(n-1)} \quad (55)$$

où A' et \mathbf{b}_1 sont donnés par (39) et (40). Le vecteur $\boldsymbol{\alpha}_S$ a la même structure que \mathbf{x} (cf. (37)) :

$$\boldsymbol{\alpha}_S = (\alpha_{12}, \alpha_{13}, \dots, \alpha_{1n} | \alpha_{21}, \alpha_{23}, \dots, \alpha_{2n} | \dots | \alpha_{n1}, \dots, \alpha_{n(n-1)})^\top \in \{0, 1\}^{n(n-1)} \quad (56)$$

et

$$\alpha_{ij} = \begin{cases} 1 & \text{si l'arête } (i, j) \in A_k \\ 0 & \text{sinon} \end{cases} \quad (57)$$

où A_k est l'ensemble des arêtes d'un sous-circuit rencontré à l'itération k et $S \subsetneq V$ l'ensemble des sommets de la composante connexe associée à ce sous-circuit.

2.3 Formulation par flot

Formulation par flot due à Gavish-Graves (1978) [2]. On conserve les variables x_{ij} pour $i, j \in \llbracket 1, n \rrbracket$ (cf. variante précédente) et on introduit un flot sur le graphe par l'intermédiaire de variables y_{ij} représentant un flot associée à l'arête (i, j) qui relie le sommet i à j . Pour le sommet 1 de départ, le flot sortant vaut $n - 1$:

$$\sum_{j \neq 1} y_{1j} = n - 1 \quad (58)$$

Pour chaque sommet autre que le sommet 1 de départ, le flot sortant est diminué d'une unité par rapport au flot entrant :

$$\sum_{i \neq j} y_{ij} - \sum_{k \neq j} y_{jk} = 1, \quad \forall j \neq 1 \quad (59)$$

Enfin, si l'arête (i, j) n'est pas empruntée i.e. $x_{ij} = 0$, le flot associé à cette arête est nul i.e. $y_{ij} = 0$:

$$y_{ij} \leq (n - 1)x_{ij}, \quad \forall i, j \in \llbracket 1, n \rrbracket, i \neq j \quad (60)$$

Dans (58)–(60), on peut relaxer les exclusions $i \neq j, k \neq j \dots$ sauf en maintenant $j \neq 1$ dans la contrainte (59). En effet, on a le résultat suivant.

Lemme 4 *Toute solution réalisable satisfaisant les contraintes (5),(6) et (58),(59),(60) étendues par relaxation des exclusions $i \neq j, k \neq j \dots$ sauf pour l'exclusion $j \neq 1$ maintenue dans la contrainte (59), vérifie*

(a) $y_{j1} = 0$ pour tout $j \in \llbracket 1, n \rrbracket$.

(b) $x_{ii} = y_{ii} = 0$ pour tout $i \in \llbracket 1, n \rrbracket$.

Preuve. (a) La contrainte (59) avec relaxation des exclusions $i \neq j$ et $k \neq j$ s'écrit

$$\sum_{i=1}^n y_{ij} - \sum_{k=1}^n y_{jk} = 1, \quad \forall j \neq 1 \quad (61)$$

On somme sur $j \neq 1$ et on réordonne les sommes pour obtenir :

$$\sum_{i=1}^n \sum_{j \neq 1} y_{ij} - \sum_{j \neq 1} \sum_{k=1}^n y_{jk} = n - 1$$

que l'on réécrit en :

$$\sum_{j \neq 1} y_{1j} + \sum_{i \neq 1} \sum_{j \neq 1} y_{ij} - \sum_{j \neq 1} y_{j1} - \sum_{j \neq 1} \sum_{k \neq 1} y_{jk} = n - 1$$

soit

$$\sum_{j=1}^n y_{1j} - y_{11} - \sum_{j \neq 1} y_{j1} = n - 1$$

On déduit de (58) avec relaxation de l'exclusion $j \neq 1$, que

$$y_{11} + \sum_{j \neq 1} y_{j1} = 0,$$

c'est-à-dire $y_{j1} = 0$ pour tout j (y compris pour $j = 1$).

(b) On suppose qu'il existe $j \in \llbracket 1, n \rrbracket$ tel que $x_{jj} = 1$. Alors, d'après (5) et (6), on a

$$x_{ij} = 0, \quad \forall i \neq j \quad \text{et} \quad x_{jk} = 0, \quad \forall k \neq j \quad (62)$$

ce qui implique, compte tenu de (60) :

$$y_{ij} = 0, \quad \forall i \neq j \quad \text{et} \quad y_{jk} = 0, \quad \forall k \neq j. \quad (63)$$

— On suppose que $j \neq 1$. On a

$$\sum_{i=1}^n y_{ij} - \sum_{k=1}^n y_{jk} = y_{jj} + \underbrace{\sum_{i \neq j} y_{ij}}_{=0} - y_{jj} - \underbrace{\sum_{k \neq j} y_{jk}}_{=0} = 0$$

ce qui contredit (61).

— On suppose que $j = 1$. D'après (58) (avec l'exclusion relaxée), on a $\sum_j y_{1j} = n - 1$ et d'après (63) qui est vraie aussi pour $j = 1$, on en déduit que $\sum_j y_{1j} = y_{11}$, d'où $y_{11} = n - 1$ ce qui contredit (a) avec $y_{11} = 0$.

Dans tous les cas, on a une contradiction et donc nécessairement $x_{jj} = 0$ pour tout $i \in \llbracket 1, n \rrbracket$. Enfin, la contrainte (60) (avec l'exclusion $i \neq j$ relaxée) permet de conclure aussi à $y_{ii} = 0, \forall i$. \square

2.3.1 Formes matricielles pour GG.

i) Exclusions relaxées GG2

On donne la forme matricielle correspondant à (4),(5),(6) avec (58),(59),(60) en relaxant les exclusions $i \neq j$ (cf. section précédente).

$$\min_{\mathbf{x}, \mathbf{y}} \mathbf{d}^\top \mathbf{x} \quad (64)$$

$$A\mathbf{x} = \mathbf{b}_1 \quad (65)$$

$$\mathbf{e}^\top \mathbf{y} = n - 1 \quad (66)$$

$$(\tilde{A}_2 - \tilde{A}_1)\mathbf{y} = \mathbf{b}_3 \quad (67)$$

$$(n - 1)\mathbf{x} - \mathbf{y} \geq \mathbf{0} \quad (68)$$

$$\mathbf{x} \in \{0, 1\}^{n^2}, \quad (69)$$

$$\mathbf{y} \in \mathbb{R}^{n^2}, \mathbf{y} \geq \mathbf{0} \quad (70)$$

Il y a $2n^2$ variables (inconnues) qui sont regroupées *ligne par ligne* dans les vecteurs

$$\begin{aligned} \mathbf{x} &= (x_{11}, x_{12}, \dots, x_{1n} \mid x_{21}, x_{22}, \dots, x_{2n} \mid \dots \mid x_{n1}, \dots, x_{nn})^\top \in \{0, 1\}^{n^2} \\ \mathbf{y} &= (y_{11}, y_{12}, \dots, y_{1n} \mid y_{21}, y_{22}, \dots, y_{2n} \mid \dots \mid y_{n1}, \dots, y_{nn})^\top \in \mathbb{R}^{n^2} \end{aligned} \quad (71)$$

— La matrice A de taille $2n \times n^2$ et le vecteur $\mathbf{b}_1 \in \mathbb{R}^{2n}$ sont donnés par (26) et (25). On a aussi

$$\mathbf{b}_3 = (1, \dots, 1)^\top \in \mathbb{R}^{n-1}, \quad \mathbf{e} = (\underbrace{1, \dots, 1}_n \mid \underbrace{0, \dots, 0}_{n(n-1)})^\top \in \mathbb{R}^{n^2} \quad (72)$$

— Les matrices \tilde{A}_1, \tilde{A}_2 de taille $(n - 1) \times n^2$ sont obtenues à partir des matrices A_1, A_2 données par (27),(28) en supprimant leurs premières lignes pour prendre en compte l'exclusion $j \neq 1$ dans (59).

Il y a au total $3n + n^2$ contraintes.

ii) GG1

On donne à présent la forme matricielle de la formulation initiale GG1 correspondant à (4),(5),(6) avec (58),(59),(60) avec les exclusions $i \neq j$.

$$\min_{\mathbf{x}, \mathbf{y}} \mathbf{d}^\top \mathbf{x} \quad (73)$$

$$A'\mathbf{x} = \mathbf{b}_1 \quad (74)$$

$$\mathbf{e}^\top \mathbf{y} = n - 1 \quad (75)$$

$$(\tilde{A}'_2 - \tilde{A}'_1)\mathbf{y} = \mathbf{b}_3 \quad (76)$$

$$(n - 1)\mathbf{x} - \mathbf{y} \geq \mathbf{0} \quad (77)$$

$$\mathbf{x} \in \{0, 1\}^{n(n-1)}, \quad (78)$$

$$\mathbf{y} \in \mathbb{R}^{n(n-1)}, \mathbf{y} \geq \mathbf{0} \quad (79)$$

Il y a $2n(n-1)$ variables (inconnues) qui sont regroupées dans les vecteurs

$$\begin{aligned} \mathbf{x} &= (x_{12}, x_{13}, \dots, x_{1n} \mid x_{21}, x_{23}, \dots, x_{2n} \mid \dots \mid x_{n1}, \dots, x_{n(n-1)})^\top \in \{0, 1\}^{n(n-1)} \\ \mathbf{y} &= (y_{12}, y_{13}, \dots, y_{1n} \mid y_{21}, y_{23}, \dots, y_{2n} \mid \dots \mid y_{n1}, \dots, y_{n(n-1)})^\top \in \mathbb{R}^{n(n-1)} \end{aligned} \quad (80)$$

— Les vecteurs \mathbf{d} , \mathbf{b}_1 , \mathbf{b}_2 et \mathbf{b}_3 sont donnés par (38),(39),(72). On a aussi

$$\mathbf{e} = (\underbrace{1, \dots, 1}_{n-1} \mid \underbrace{0, \dots, 0}_{(n-1)^2})^\top \in \mathbb{R}^{n(n-1)} \quad (81)$$

— La matrice A' de taille $2n \times n(n-1)$ est donnée par (40).

— Les matrices $\tilde{A}'_1, \tilde{A}'_2$ de taille $(n-1) \times n(n-1)$ sont obtenues à partir des matrices A'_1 et A'_2 données par (40), en supprimant leurs premières lignes (pour prendre en compte l'exclusion $j \neq 1$ dans (59)).

Il y a au total $3n + n(n-1)$ contraintes.

2.4 Résultats numériques avec données non-symétriques

On compare les formulations DFJ asymétrique, MTZ et GG avec leurs variantes. Plus précisément, on a considéré les formulations suivantes.

DFJa : formulation DFJ *asymétrique*.

MTZ1 : avec les exclusions $i \neq j$: variables x_{ij} pour $i \neq j$.

MTZ2 : sans les exclusions : variables x_{ii} prises en compte.

MTZ3 : sans les exclusions mais variables $x_{ii} = 0$ forcées sur les bornes.

GG1 : avec les exclusions $i \neq j$: variables x_{ij} et y_{ii} pour $i \neq j$.

GG2 : sans les exclusions : variables x_{ii} et y_{ii} prises en compte.

GG3 : sans les exclusions mais variables $x_{ii} = y_{ii} = 0$ forcées sur les bornes.

Le solveur utilisé est **GUROBI** 11.0.3 interfacé dans **Python** (sauf pour GG avec interface dans **MATLAB**). Les temps de construction des matrices du modèle sont négligeables par rapport à la résolution du PL par **GUROBI**. On indiquera seulement les temps de résolution par **GUROBI** (après construction du modèle). Les calculs ont été effectués sur un MacBook Pro, 4 GHz Apple M3 Max, 14 cœurs, mémoire 36 Go.

Les coefficients de la matrice des distances $D = (d_{ij})$ sont des réels choisis aléatoirement de façon *uniforme* dans l'intervalle $[1, 51]$. On teste plusieurs jeux de données en fonction de la taille n et de matrices de distances *non-symétriques*. Les résultats sont reportés dans le tableau 1. Pour DFJa, les contraintes (51) sont ajoutées de façon itérative pour supprimer les sous-circuits. Le nombre d'itérations d'élimination des sous-circuits est également donné dans le tableau.

Les formulations MTZ3 et GG3 n'ont pas apportées d'améliorations notables par rapport à MTZ2 et GG2. On constate que la formulation DFJa est la plus rapide avec ensuite MTZ1. Les formulations MTZ1 et MTZ2 ont des performances assez proches (sauf peut-être pour des problèmes de très grandes tailles).

3 TSP symétrique

3.1 Formulation

Lorsque les distances sont symétriques, on peut utiliser une autre formulation modifiée de DFJ où les variables sont associées aux arêtes du graphe (chemins entre les villes), voir [1],[4]. On notera V l'ensemble des villes (sommets du graphe) et $A = \{k = \{i, j\}; i, j \in V \text{ avec } i \neq j\}$ l'ensemble des arêtes non-orientées. On associe de cette façon le graphe *non-orienté* $G = (V, A)$ au TSP symétrique. On introduit les variables

$$x_k = \begin{cases} 1 & \text{si l'arête } k \text{ est parcourue} \\ 0 & \text{sinon.} \end{cases}$$

Distances non-symétriques

n	d_{min}	DFJa	MTZ1	MTZ2	GG1	GG2
2000	2082.30	2230 (76)	2527	3148	-	-
1500	1581.95	259 (17)	963	1105	-	-
1000	1080.72	64.9 (13)	231	250	-	-
600	683.33	27.5 (12)	72.2	74.04	430	-
500	585.15	7.8 (7)	25.5	26.4	1731	2100
400	477.95	4.5 (6)	21.7	19.7	600	468
350	431.22	0.7 (1)	21.2	23.0	351	130
300	380.89	0.9 (2)	10.5	11.1	115	40
280	364.21	2.6 (6)	7.6	7.2	148	171
260	347.29	1.9 (6)	14.8	9.0	75	141
240	329.65	10 (25)	5.8	6.1	45	87
220	302.33	8.1 (19)	8.8	19.0	32	32.7
200	280.01	3.7 (15)	4.5	4.8	27	31
150	235.68	1.4 (10)	1.7	1.8	6.4	8.0
100	180.95	0.4 (6)	0.7	0.8	1.65	1.8
50	143.71	0.1 (4)	0.15	0.16	0.39	0.5
10	113.46	0.006 (0)	0.006	0.005	0.05	0.05

TABLE 1 – Temps CPU (en s) du solveur Gurobi 11.0.3 pour les différentes formulations en fonction de la taille n du TSP *asymétrique*. Les données (distances) sont *non-symétriques*. Le nombre d'itérations pour éliminer les sous-circuits est donné entre parenthèse pour DFJa.

On note $m = |A|$ le nombre d'arêtes, $n = |V|$ le nombre de villes. Si le graphe est complet (les sommets sont tous reliés entre eux), on a $m = n(n-1)/2$. Pour simplifier on note $V = \{1, \dots, n\}$ et $A = \{1, \dots, m\}$. Les distances étant symétriques, on considère les distances des arêtes $D = (d_k)_{1 \leq k \leq m}$. La formulation s'écrit

$$\min \sum_{k=1}^m d_k x_k \quad (82)$$

$$\sum_{k \in \delta(i)} x_k = 2, \quad \forall i \in \llbracket 1, n \rrbracket \quad (83)$$

$$\sum_{k \in \delta(W)} x_{ij} \geq 2, \quad \forall W \subsetneq V, |W| \geq 2 \quad (84)$$

$$x_k \in \{0, 1\}, \quad \forall k \in \llbracket 1, m \rrbracket, \quad (85)$$

où $\delta(i)$ désigne l'ensemble des arêtes ayant i comme sommet et $\delta(W)$ l'ensemble des arêtes allant d'un sommet de W à un sommet de $V \setminus W$.

La contrainte (84) permet de supprimer les sous-circuits. Comme pour la formulation DFJ, la contrainte (84) peut s'exprimer de façon équivalente, sous une forme plus simple à utiliser d'un point de vue pratique.

Lemme 5 *Soit $W \subset V$. On note A_W l'ensemble des arêtes contenues dans W i.e. $A_W = \{k \in A, k = \{i, j\} \text{ avec } i, j \in W\}$. Sous la contrainte (83), la contrainte (84) est équivalente à :*

$$\sum_{k \in A_W} x_k \leq |W| - 1, \quad \forall W \subsetneq V, |W| \geq 2 \quad (86)$$

Preuve. Soit $i \in W$. On scinde en deux la somme sur les arêtes dans la contrainte (83) :

$$\sum_{\substack{k \in \delta(i) \\ k \in A_W}} x_k + \sum_{\substack{k \in \delta(i) \\ k \notin A_W}} x_k = 2$$

On somme alors sur $i \in W$:

$$\sum_{i \in W} \sum_{\substack{k \in \delta(i) \\ k \in A_W}} x_k = 2|W| - \sum_{i \in W} \sum_{\substack{k \in \delta(i) \\ k \notin A_W}} x_k.$$

Or $\sum_{i \in W} \sum_{\substack{k \in \delta(i) \\ k \notin A_W}} x_k = \sum_{k \in \delta(W)} x_k$ et de plus $\sum_{i \in W} \sum_{\substack{k \in \delta(i) \\ k \in A_W}} x_k = 2 \sum_{k \in A_W} x_k$ car on compte deux fois les arêtes.

On obtient donc

$$2 \sum_{k \in A_W} x_k = 2|W| - \sum_{k \in \delta(W)} x_k \quad (87)$$

En utilisant (84), on en déduit (86). \square

Il y a un nombre exponentiel de contraintes mais dans la pratique, on procède comme pour la formulation DFJ asymétrique, c'est-à-dire de façon itérative, on ajoute la contrainte (86) sur les composantes connexes rencontrées à chaque itération. Comme pour la formulation DFJ asymétrique, ce n'est pas exactement la contrainte (86) qui est utilisée mais une variante où la somme porte uniquement sur les arêtes constituant le sous-circuit et non pas sur toutes les arêtes de la composante connexe.

3.2 Résultats numériques : comparaison des formulations avec données symétriques

La matrice des distances est *symétrique*. Les distances sont des *entiers* choisis aléatoirement de façon *uniforme* dans l'intervalle $\llbracket 1, 51 \rrbracket$. Les résultats obtenus sont reportés dans le tableau 2. On note DFJs la formulation DFJ *symétrique*.

Distances symétriques

n	d_{min}	DFJs	DFJa	MTZ1	MTZ2	GG1	GG2
2000	2000	7405 (58)	-	-	-	-	
1500	1500	2526 (46)	-	2389	-	-	
1000	1000	131 (10)	>6591 (>1000)	700	-	>5000	
600	600	7.0 (6)	1937 (778)	76	-	407	
500	500	8 (11)	558 (373)	23	-	219	
400	401	7.2 (14)	372 (393)	23	-	208	-
350	354	1.4 (5)	239 (328)	118	-	111	>5200
300	306	1.1 (5)	195 (342)	62	-	80	187
280	285	0.4 (2)	101 (215)	73	-	178	378
260	276	1.1 (6)	26 (84)	50	-	144	271
240	250	2.1 (12)	9.4 (42)	16	-	81.1	143
220	238	0.15 (1)	7.4 (41)	8.7	-	64.0	129
200	217	0.04 (0)	5.6 (31)	9.3	-	38.2	44.6
150	193	0.4 (7)	2.3 (20)	5.1	>3166	13.9	19.3
100	135	0.26 (9)	1.4 (27)	1.3	7	2.5	3.5
50	110	0.02 (1)	0.1 (5)	0.24	0.2	0.45	0.46
10	64	0.01 (1)	0.02 (2)	0.08	0.01	0.07	0.07

TABLE 2 – Temps CPU (en s) du solveur Gurobi 11.0.3 pour les différentes formulations en fonction de la taille n du TSP. Les données sont *symétriques*.

La formulation DFJs obtient les meilleures performances (temps CPU et taille du problème) suivie d'assez près par MTZ1. Dans les formulations itératives DFJ, après l'ajout d'une contrainte, le nouveau PL est résolu à nouveau sans donner de solution initiale (réalisable ou duale-réalisable). Les performances pourraient sans doute être améliorées en tenant compte de la solution à l'itération précédente.

Références

- [1] Dantzig, G.B., Fulkerson D.R. and Johnson S.M. *Solutions of a large scale travelling salesman problem*, Ops. Res., 2, 393-410 (1954).
- [2] Gavish, B. and Graves S.C. *The travelling salesman problem and related problems*, Working Paper OR-078-78 (1978), Operations Research Center, MIT, Cambridge, MA.
- [3] Miller C.E., Tucker A.W. and Zemlin R.A. *Integer programming formulation of travelling salesman problems*, J. ACM, 3, 326-329 (1960).
- [4] L. H. Wolsey, *Integer Programming*, John Wiley & Sons, 1998.