

# INFO-F310 — Projet TSP : comparaison MTZ et DFJ

Anass L'marzguioui 000537609

Islam Mekhiouba 000538266

17/12/2025

## 1 Introduction

Le problème du voyageur de commerce (TSP) consiste à trouver un cycle hamiltonien de coût minimal visitant chaque ville exactement une fois et revenant au point de départ. Nous comparons deux formulations ILP : MTZ et DFJ, ainsi qu'une version DFJ avec génération itérative de contraintes.

## 2 Formulations

### 2.1 Variables et contraintes communes

Variables  $x_{ij} \in \{0, 1\}$  pour  $i \neq j$ . Contraintes de degré :

$$\sum_{j \neq i} x_{ij} = 1, \quad \sum_{j \neq i} x_{ji} = 1 \quad \forall i.$$

Objectif :

$$\min \sum_{i \neq j} c_{ij} x_{ij}.$$

### 2.2 MTZ

Variables auxiliaires  $u_i$  pour  $i \in \{1, \dots, n-1\}$  avec  $1 \leq u_i \leq n-1$ . Contraintes MTZ :

$$u_i - u_j + (n-1)x_{ij} \leq n-2 \quad \forall i \neq j, i, j \in \{1, \dots, n-1\}.$$

Si  $x_{ij} = 1$ , alors  $u_j \geq u_i + 1$  : un sous-tour induit une contradiction.

### 2.3 DFJ

Pour tout sous-ensemble  $S \subset V$  avec  $2 \leq |S| \leq n-1$  :

$$\sum_{i \in S} \sum_{j \in S, j \neq i} x_{ij} \leq |S| - 1.$$

Génération itérative : résoudre sans coupes, détecter les cycles, ajouter une coupe DFJ par sous-tour, répéter.

## 3 Tâche 2 : exemple à 5 villes

Deux sous-tours :

$$0 \rightarrow 1 \rightarrow 0 \quad \text{et} \quad 2 \rightarrow 3 \rightarrow 4 \rightarrow 2.$$

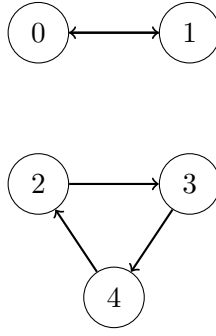


FIGURE 1 – Deux sous-tours sur 5 villes.

**Violation MTZ.** Dans le cycle  $(2, 3, 4)$ ,  $x_{23} = x_{34} = x_{42} = 1$  implique :  $u_3 \geq u_2 + 1$ ,  $u_4 \geq u_3 + 1$ ,  $u_2 \geq u_4 + 1$  donc contradiction.

**Violation DFJ.** Pour  $S = \{2, 3, 4\}$ , la somme des arcs internes vaut 3, mais DFJ impose  $\leq |S| - 1 = 2$ .

## 4 Tâche 3 : résultats expérimentaux

### 4.1 Méthodologie

Solveur : CBC via PuLP. Temps mesuré : uniquement `prob.solve()`. DFJ énumératif : uniquement pour  $n \leq 15$ .

### 4.2 Résultats par instance

TABLE 1: Comparaison des formulations (entier) : objectif, temps solveur, tailles de modèles.

Instance	n	Obj MTZ	T MTZ	Vars MTZ	Constr MTZ	Obj DFJ_enum	T DFJ_enum	Vars DFJ_enum	Constr DFJ_enum	Obj DFJ_iter	T DFJ_iter	Vars DFJ_iter	Constr DFJ_iter	Iter DFJ_iter
10_circle1	10	298.57	1.939	99	92	298.57	0.503	90	1032	298.57	0.274	90	28	2
10_euclidean1	10	207.85	0.263	99	92	207.85	0.222	90	1032	207.85	0.249	90	25	1
10_euclidean2	10	323.92	0.304	99	92	323.92	0.202	90	1032	323.92	0.276	90	28	2
10_line1	10	156.14	1.603	99	92	156.14	0.222	90	1032	156.14	0.413	90	33	4
10_random_asym1	10	191.53	0.110	99	92	191.53	0.208	90	1032	191.53	0.085	90	20	0
10_random_sym1	10	280.99	0.442	99	92	280.99	0.218	90	1032	280.99	0.266	90	28	2
12_circle1	12	265.22	1.324	143	134	265.22	2.450	132	4106	265.22	0.597	132	39	4
12_line1	12	174.80	5.279	143	134	174.80	1.201	132	4106	174.80	0.865	132	43	6
14_circle1	14	304.48	2.713	195	184	304.48	4.859	182	16396	304.48	0.242	182	34	1
14_line1	14	188.02	20.775	195	184	188.02	5.535	182	16396	188.02	0.794	182	53	7
15_euclidean1	15	306.70	4.105	224	212	306.70	10.487	210	32781	306.70	0.880	210	50	5
15_euclidean2	15	345.91	10.825	224	212	345.91	11.162	210	32781	345.91	0.405	210	45	3
15_random_asym1	15	342.94	0.238	224	212	342.94	10.298	210	32781	342.94	0.219	210	32	1
15_random_sym1	15	377.50	1.140	224	212	377.50	10.143	210	32781	377.50	0.191	210	37	1
20_euclidean1	20	370.13	0.912	399	382					370.13	0.273	380	50	1
20_euclidean2	20	396.75	23.903	399	382					396.75	0.343	380	56	2
20_random_asym1	20	324.08	0.507	399	382					324.08	0.242	380	43	1
20_random_sym1	20	371.10	4.267	399	382					371.10	1.357	380	59	5
25_euclidean1	25	410.26	12.907	624	602					410.26	1.696	600	72	4
25_euclidean2	25	418.56	44.769	624	602					418.56	0.628	600	74	4

### 4.3 Synthèse (moyennes par taille $n$ )

n	Formulation	Temps moyen (s)	Temps médian (s)	Constr moy.	Vars moy.	Gap moy.	Iter moy.
10	DFJ_enum	0.263	0.220	1032.0	90.0	0.000	1.83
10	DFJ_iter	0.260	0.270	27.0	90.0		
10	MTZ	0.777	0.373	92.0	99.0	0.129	
12	DFJ_enum	1.825	1.825	4106.0	132.0	-0.000	5.00
12	DFJ_iter	0.731	0.731	41.0	132.0		
12	MTZ	3.301	3.301	134.0	143.0	0.347	
14	DFJ_enum	5.197	5.197	16396.0	182.0	-0.000	4.00
14	DFJ_iter	0.518	0.518	43.5	182.0		
14	MTZ	11.744	11.744	184.0	195.0	0.444	
15	DFJ_enum	10.523	10.393	32781.0	210.0	0.000	2.50
15	DFJ_iter	0.424	0.312	41.0	210.0		
15	MTZ	4.077	2.623	212.0	224.0	0.167	
20	DFJ_iter	0.554	0.308	52.0	380.0		2.25
20	MTZ	7.397	2.590	382.0	399.0	0.087	
25	DFJ_iter	1.162	1.162	73.0	600.0		4.00
25	MTZ	28.838	28.838	602.0	624.0	0.141	

**Analyse.** DFJ\_iter est en général le plus rapide : il conserve une relaxation serrée et n'ajoute que quelques contraintes utiles. MTZ est plus compact mais souvent pénalisé par une relaxation moins serrée.

## 5 Tâche 4 : relaxation continue et integrality gap

$$\text{gap} = \frac{z_{\text{int}} - z_{\text{relax}}}{z_{\text{int}}}.$$

### 5.1 Gaps par instance

Instance	n	Type	Gap(MTZ)	Gap(DFJ_enum)
instance_10_circle_1.txt	10	circle	0.163	0.000
instance_10_euclidean_1.txt	10	euclidean	0.036	0.000
instance_10_euclidean_2.txt	10	euclidean	0.034	0.000
instance_10_line_1.txt	10	line	0.477	0.000
instance_10_random_asym_1.txt	10	random_asym	0.000	0.000
instance_10_random_sym_1.txt	10	random_sym	0.066	0.000
instance_12_circle_1.txt	12	circle	0.332	0.000
instance_12_line_1.txt	12	line	0.362	-0.000
instance_14_circle_1.txt	14	circle	0.269	0.000
instance_14_line_1.txt	14	line	0.619	-0.000
instance_15_euclidean_1.txt	15	euclidean	0.253	0.000
instance_15_euclidean_2.txt	15	euclidean	0.295	0.000
instance_15_random_asym_1.txt	15	random_asym	0.001	0.000
instance_15_random_sym_1.txt	15	random_sym	0.118	0.000
instance_20_euclidean_1.txt	20	euclidean	0.146	
instance_20_euclidean_2.txt	20	euclidean	0.151	
instance_20_random_asym_1.txt	20	random_asym	0.014	
instance_20_random_sym_1.txt	20	random_sym	0.039	
instance_25_euclidean_1.txt	25	euclidean	0.154	
instance_25_euclidean_2.txt	25	euclidean	0.129	

## 6 Figures

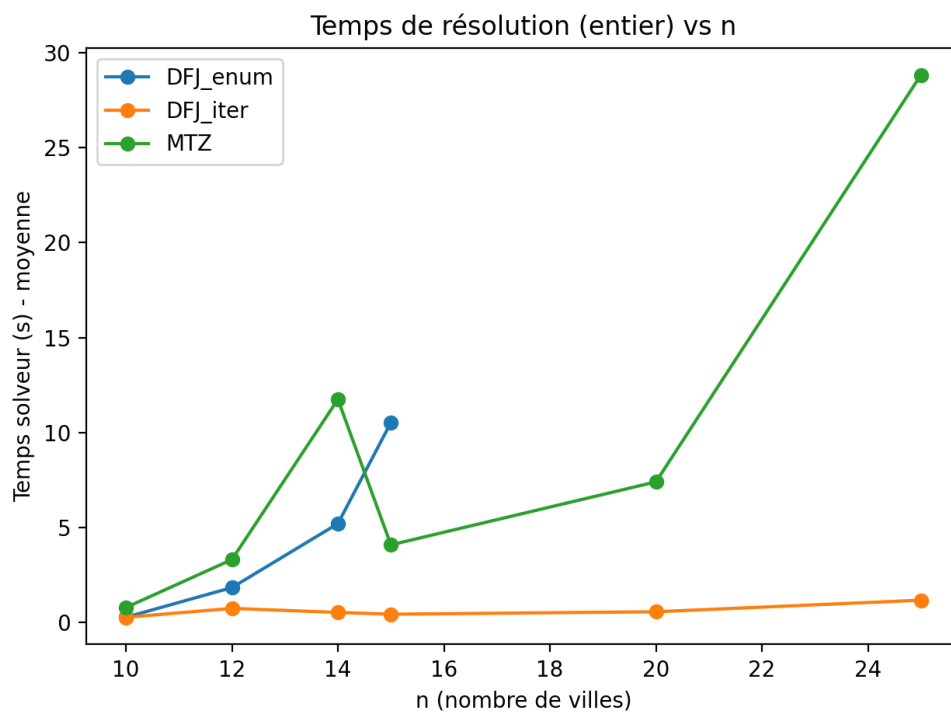


FIGURE 2 – Temps solveur (entier) moyen en fonction de  $n$ .

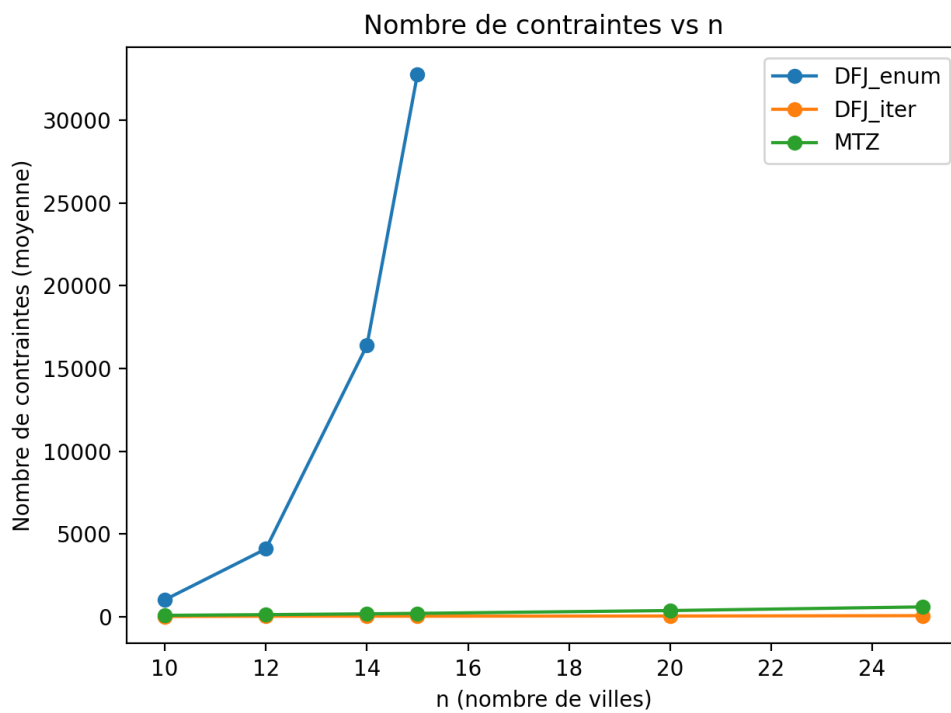


FIGURE 3 – Nombre moyen de contraintes en fonction de  $n$ .

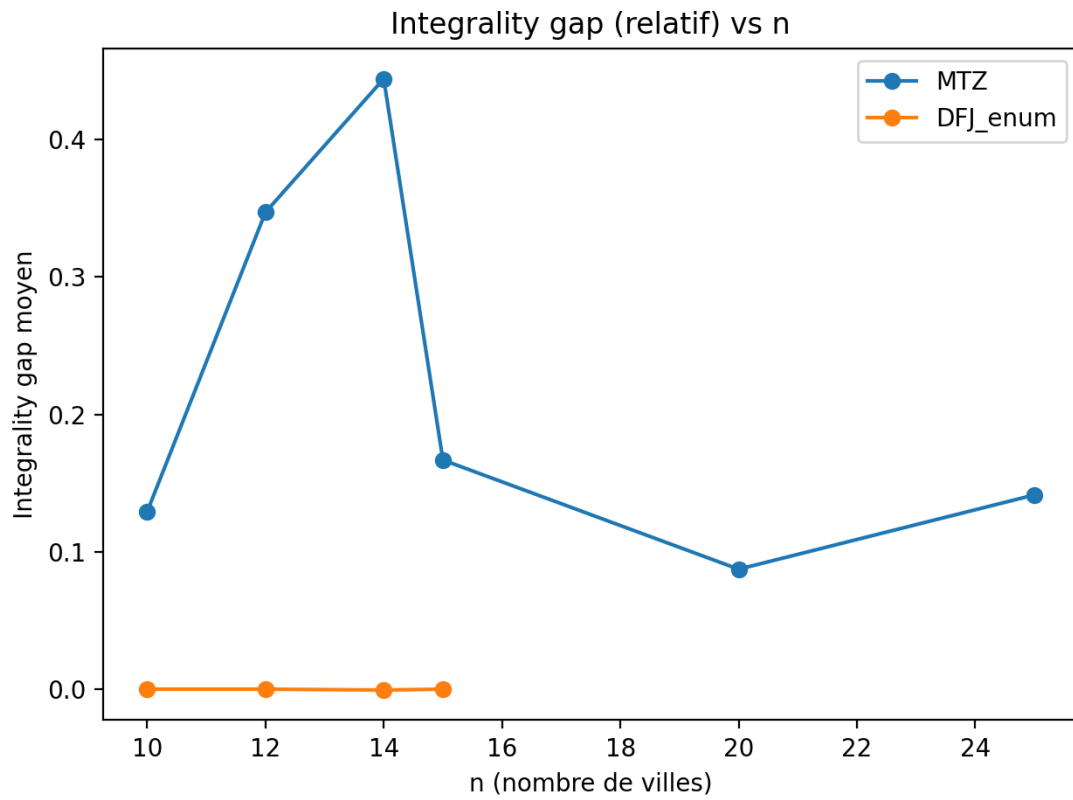


FIGURE 4 – Integrality gap moyen (MTZ vs DFJ\_enum) en fonction de  $n$ .

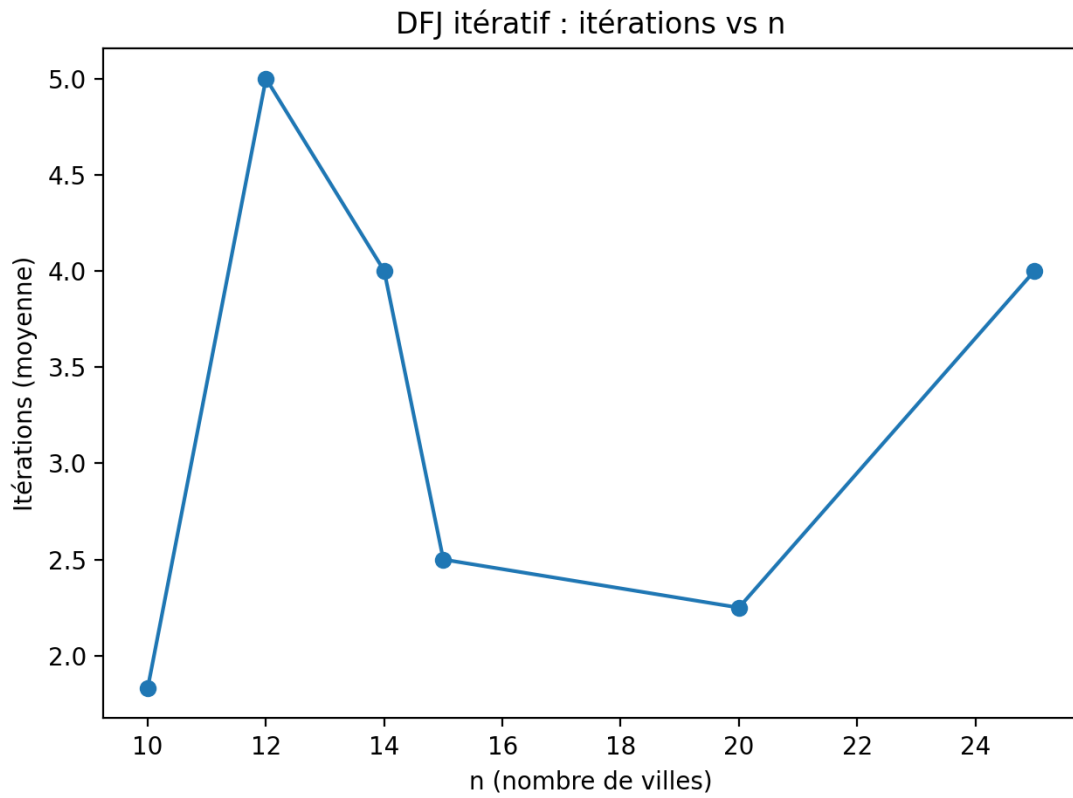


FIGURE 5 – DFJ itératif : itérations moyennes en fonction de  $n$ .

## 7 Tâche 5 : questions de réflexion

### 7.1 Pourquoi énumérer les tours est impraticable, mais DFJ est praticable ?

Énumérer explicitement toutes les tournées implique un nombre factoriel  $((n - 1)!/2)$ . DFJ a un nombre exponentiel théorique de contraintes, mais la génération itérative n'ajoute que les contraintes violées (sous-tours rencontrés), ce qui suffit en pratique.

### 7.2 Pourquoi DFJ peut être plus rapide que MTZ malgré plus de contraintes ?

DFJ a une relaxation plus serrée : le solveur explore moins de nœuds en branch-and-bound. MTZ, plus compact, peut produire des relaxations plus lâches, augmentant l'exploration.

### 7.3 Bonus : 2 sous-tours

Si exactement deux cycles  $S$  et  $\bar{S}$  existent, ajouter une seule contrainte DFJ sur l'un des deux suffit (l'autre devient redondante car complémentaire). On peut vérifier en n'ajoutant qu'une coupe dans ce cas et en comparant temps/itérations.

## 8 Conclusion

DFJ itératif offre le meilleur compromis pratique : relaxation forte et peu de contraintes ajoutées. MTZ reste simple mais peut souffrir d'un gap plus élevé sur certaines instances.