UNIVERSITÉ SORBONNE PARIS NORD
99 AVENUE JEAN-BAPTISTE CLÉMENT
93430 VILLETANEUSE

SAÉ Optimisation 2025–2026 Conception d'un métro circulaire : le problème Ring-Star

Auteurs :

Papa Birane MBENGUE
Mouhamadou Lamine SOW

Équipe de suivi :

M. Lucas Létocart
M. Pierre Fouilhoux
M. Nabil H. Mustafa

30 janvier 2026

Résumé

Ce rapport étudie le **problème Ring-Star**, une modélisation classique pour concevoir une ligne de métro circulaire (un *anneau*) complétée par des rattachements en *étoile* vers les stations. Sur un ensemble de n pôles d'activité, il s'agit de choisir exactement p stations (dont une station imposée), de construire un cycle passant par ces stations et d'affecter chaque pôle à une station afin de minimiser un coût total combinant *infrastructure* (longueur de l'anneau) et *accessibilité* (distances de rattachement). Nous montrons la NP-difficulté du problème via des cas particuliers, proposons une formulation PLNE compacte résolue par CBC/PuLP, puis développons une heuristique constructive et une métaheuristique de descente stochastique. Des expériences sur des instances euclidiennes de TSPLIB (`ulysses16`, `eil51`) mettent en évidence un bon compromis qualité/temps pour la métaheuristique, et les limites de la résolution exacte lorsque p est proche de $n/2$.

Mots-clés : optimisation combinatoire, Ring-Star, PLNE, heuristique, métaheuristique, TSPLIB.

Notations

Symbole	Signification
V	ensemble des pôles d'activité (sommets), $\lvert V\rvert = n$
p	nombre de stations à sélectionner (le sommet 1 est imposé)
D_{ij}	distance (euclidienne) entre les pôles i et j
α	pondération du coût métro dans la fonction objectif ($\alpha = 1$ dans nos tests)
y_{ij}	variable binaire d'affectation : $y_{ij} = 1$ si i est rattaché à la station j
x_{ij}	variable binaire : $x_{ij} = 1$ si l'arête non orientée $\{i, j\}$ appartient à l'anneau
z_{ij}	variables de flot (arcs orientés) pour imposer la connexité et éliminer les sous-tours

TABLE 1 – Notations principales utilisées dans le rapport.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 Contexte et objectif

L'optimisation des réseaux de transport public constitue un enjeu central de l'aménagement urbain. Dans de nombreuses configurations, une *ligne circulaire* est attractive car elle distribue des flux autour d'un centre et peut connecter efficacement plusieurs zones d'activité. En contrepartie, la construction d'un anneau complet est coûteuse et il est souvent nécessaire de choisir un nombre limité de stations, quitte à compléter la desserte par des liaisons de rabattement (marche, bus, navette).

1.2 Définition du problème Ring-Star

Le problème Ring-Star formalise ce compromis. À partir d'un ensemble de n pôles d'activité $V = \{1, \ldots, n\}$ et d'une matrice de distances D, on cherche :

- un ensemble $S \subseteq V$ de p stations, avec la station $1 \in S$ imposée ;
- un cycle (anneau) passant exactement une fois par chaque station de S ;
- une affectation de chaque pôle $i \in V$ vers une station $a(i) \in S$.

L'objectif est de minimiser un coût total de la forme :

$$\min \quad \alpha \sum_{\{i,j\} \in \text{anneau}} D_{ij} \; + \; \sum_{i \in V} D_{i,a(i)}.$$

Le premier terme mesure la longueur de l'anneau (coût d'infrastructure), tandis que le second mesure l'accessibilité via les rattachements en étoile (coût « marche »). La figure 1.1 illustre le principe.

1.3 Instances étudiées

Nos expériences s'appuient sur des instances euclidiennes de TSPLIB : `ulysses16` ($n = 16$) et `eil51` ($n = 51$). La figure 1.2 montre les nuages de points (données brutes). La résolution du Ring-Star consiste à choisir p points comme stations, tracer un cycle sur ces stations et affecter les autres points.

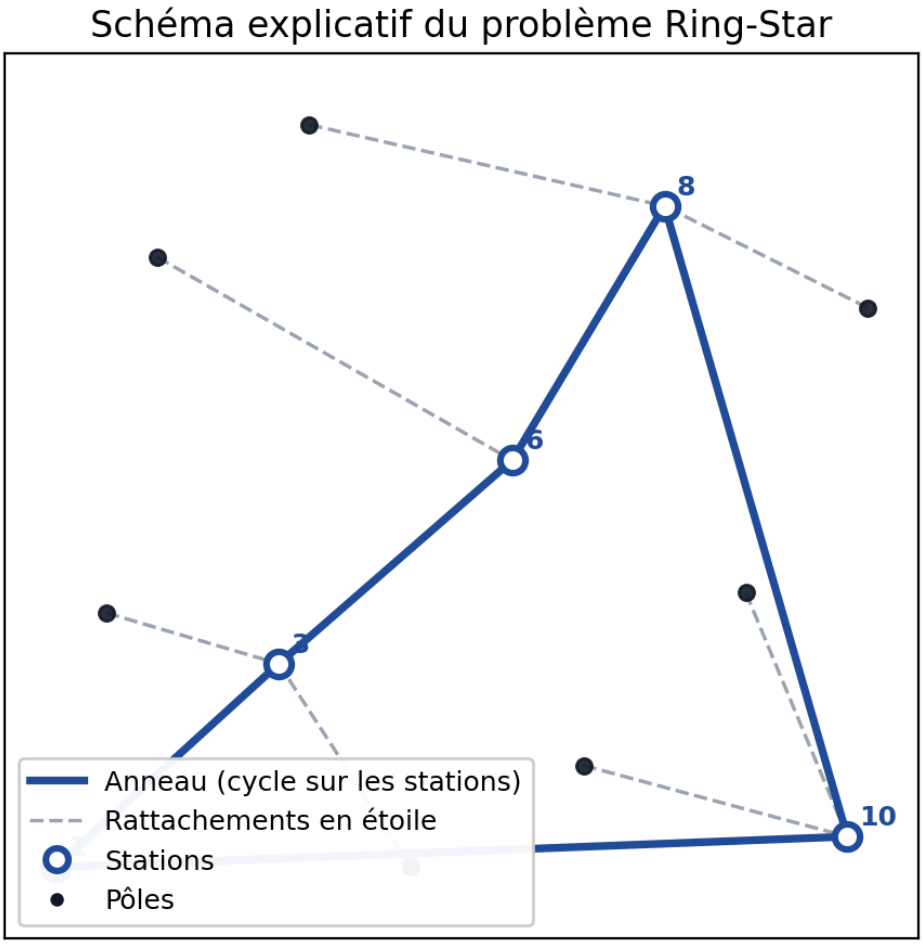
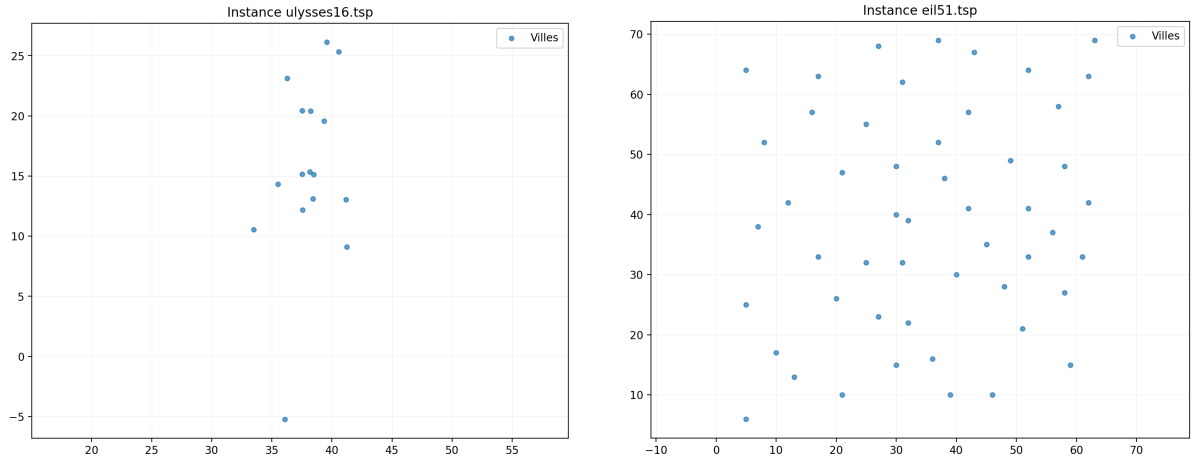


FIGURE 1.1 – Schéma de principe du Ring-Star : un anneau relie les p stations sélectionnées, et chaque pôle non station est rattaché à sa station la plus proche (liaisons en étoile).



(a) Instance `ulysses16` ($n = 16$). (b) Instance `eil51` ($n = 51$).

FIGURE 1.2 – Nuages de points issus de TSPLIB (données d'entrée).

1.4 Organisation du rapport

Le chapitre 2 établit la NP-difficulté. Le chapitre 3 présente une formulation PLNE compacte. Le chapitre 4 décrit une heuristique constructive et une métaheuristique d'amélioration. Enfin, le chapitre 5 analyse les résultats expérimentaux et l'influence de p sur la difficulté de la résolution exacte.

Chapitre 2

NP-difficulté du problème Ring-Star

Avant de concevoir des algorithmes de résolution, il est essentiel de caractériser la difficulté du problème. Nous montrons la NP-difficulté du Ring-Star en exhibant deux problèmes NP-difficiles comme cas particuliers.

2.1 Cas particulier p=n : réduction au TSP

Si $p = n$, chaque pôle est une station. Il n'existe alors aucun rattachement en étoile (coût marche nul) et le problème se réduit à trouver un cycle passant une fois par chaque sommet et minimisant la somme des longueurs : c'est exactement le *problème du voyageur de commerce* (TSP), NP-difficile. Ainsi, Ring-Star généralise le TSP, donc il est NP-difficile.

2.2 Cas particulier alpha=0 : réduction au p-médian

Si $\alpha = 0$, le coût de l'anneau devient nul : il ne reste qu'à choisir p stations et à affecter chaque pôle à une station pour minimiser la somme des distances d'affectation. Ce cas particulier correspond au *problème du p-médian*, également NP-difficile. On en déduit à nouveau la NP-difficulté du Ring-Star.

2.3 Conséquences

En pratique, on ne peut pas espérer un algorithme exact polynomial pour toutes les instances. Cela justifie l'usage combiné (i) d'une PLNE pour les petites tailles et (ii) d'heuristiques et métaheuristiques pour les configurations réalistes.

Chapitre 3

Modélisation mathématique (PLNE)

Pour obtenir des solutions optimales sur des instances de taille raisonnable, nous proposons une formulation en **programmation linéaire en nombres entiers (PLNE)** résolue avec PuLP et le solveur CBC.

3.1 Variables de décision

On introduit :

— $y_{ij} \in \{0,1\}$: $y_{ij} = 1$ si le pôle i est affecté à la station j ;

— $x_{ij} \in \{0,1\}$ (pour $i < j$) : $x_{ij} = 1$ si l'arête $\{i, j\}$ appartient à l'anneau ;

— $z_{ij} \geq 0$ (pour $i \neq j$) : flot orienté utilisé pour garantir la connexité et éliminer les sous-tours.

3.2 Formulation compacte

La fonction objectif est :

$$\min \quad \alpha \sum_{1 \leq i < j \leq n} D_{ij}\, x_{ij} \; + \; \sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij}\, y_{ij}. \tag{3.1}$$

Sous les contraintes suivantes :

$$\sum_{j=1}^{n} y_{jj} = p \qquad \text{(exactement } p \text{ stations)} \tag{3.2}$$

$$\sum_{j=1}^{n} y_{ij} = 1 \qquad \forall i \in \{1,\dots,n\} \quad \text{(affectation unique)} \tag{3.3}$$

$$y_{ij} \le y_{jj} \qquad \forall i,j \in \{1,\dots,n\} \quad \text{(affectation} \Rightarrow \text{station)} \tag{3.4}$$

$$y_{11} = 1 \qquad \text{(station 1 imposée)} \tag{3.5}$$

$$x_{ij} \le y_{ii},\ x_{ij} \le y_{jj} \qquad \forall 1 \le i < j \le n \quad \text{(arête} \Rightarrow \text{deux stations)} \tag{3.6}$$

$$\sum_{j<i} x_{ji} + \sum_{j>i} x_{ij} = 2\, y_{ii} \qquad \forall i \in \{1,\dots,n\} \quad \text{(degré 2 pour les stations)} \tag{3.7}$$

$$\sum_{j\neq i} z_{ij} - \sum_{j\neq i} z_{ji} = y_{ii} \qquad \forall i \in \{2,\dots,n\} \quad \text{(flot vers chaque station)} \tag{3.8}$$

$$\sum_{j\neq 1} z_{1j} - \sum_{j\neq 1} z_{j1} = p - 1 \qquad \text{(source)} \tag{3.9}$$

$$z_{ij} + z_{ji} \le (p-1)\, x_{ij} \qquad \forall 1 \le i < j \le n \quad \text{(capacité liée à l'anneau)} \tag{3.10}$$

$$z_{ij} \ge 0 \qquad \forall i \neq j. \tag{3.11}$$

Les contraintes de flot garantissent qu'un unique cycle connecte toutes les stations (pas de sous-tours), tout en conservant un modèle de taille polynomiale.

Chapitre 4

Algorithmes approchés

Compte tenu de la NP-difficulté, les méthodes exactes deviennent rapidement coûteuses. Nous proposons une heuristique constructive pour produire une solution faisable, puis une métaheuristique d'amélioration.

4.1 Heuristique constructive

L'heuristique gloutonne suit les étapes suivantes :

1. **Initialisation par grille :** on partitionne le plan et on retient des représentants spatiaux pour obtenir un premier ensemble de stations bien réparties.
2. **Complément maximin :** tant que $|S| < p$, on ajoute le pôle maximisant sa distance à l'ensemble S (évite les regroupements).
3. **Affectation :** chaque pôle est rattaché à la station la plus proche (variables y).
4. **Construction du cycle :** on construit un cycle sur S via un plus-proche-voisin (heuristique TSP).

Algorithm 1 Heuristique constructive (gloutonne)

Entrée : points V, entier p, distances D, station imposée 1
Construire une grille et extraire un premier ensemble S
while $|S| < p$ **do**
    Ajouter au S le pôle maximisant sa distance minimale à S (maximin)
end while
Affecter chaque pôle i à la station $j \in S$ la plus proche (variables y_{ij})
Construire un cycle sur S par plus-proche-voisin (variables x_{ij})
Retourner (S, x, y) et le coût total

4.2 Descente stochastique par échanges (SWAP)

À partir d'une solution, la descente stochastique explore un voisinage par échanges : on remplace une station (hors station imposée) par un pôle non station, puis on recalcule affectations et cycle. La nouvelle solution est acceptée si elle améliore le coût.

Algorithm 2 Métaheuristique de descente stochastique (SWAP)

Entrée : solution initiale (S, x, y), distances D, α, station imposée 1, itérations k
$S^\star \leftarrow S$ (meilleure solution courante)
for $t = 1$ **to** k **do**
  Tirer au hasard une station $s \in S \setminus \{1\}$ et un pôle $v \in V \setminus S$
  Former $S' \leftarrow (S \setminus \{s\}) \cup \{v\}$
  Recalculer l'affectation et le cycle sur S', obtenir le coût $C(S')$
  if $C(S') < C(S)$ **then**
    Accepter : $S \leftarrow S'$
    if $C(S) < C(S^\star)$ **then**
      $S^\star \leftarrow S$
    end if
  end if
end for
Retourner $S^\star$

4.3 Complexité (ordre de grandeur)

La construction gloutonne effectue principalement des affectations (coût $\mathcal{O}(np)$) et un cycle sur p stations (coût $\mathcal{O}(p^2)$). La descente stochastique répète ces opérations k fois ; dans notre implémentation, $k = \max(500, 10n)$, ce qui reste compatible avec des temps quasi instantanés pour les tailles considérées.

Chapitre 5

Résultats expérimentaux

5.1 Protocole expérimental

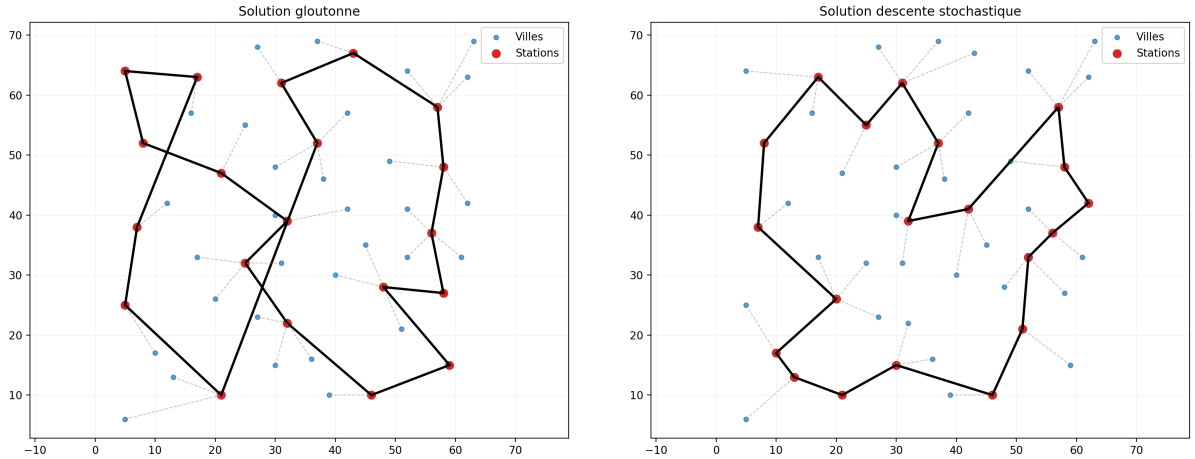
Les paramètres sont fixés de manière reproductible :

- pondération $\alpha = 1$;
- station imposée : sommet 1 (indice 0 dans le code) ;
- graine aléatoire fixée à 0 ;
- nombre d'itérations : $k = \max(500, 10n)$.

Nous comparons trois approches : heuristique gloutonne, descente stochastique, et PLNE (lorsque le solveur termine).

5.2 Analyse qualitative

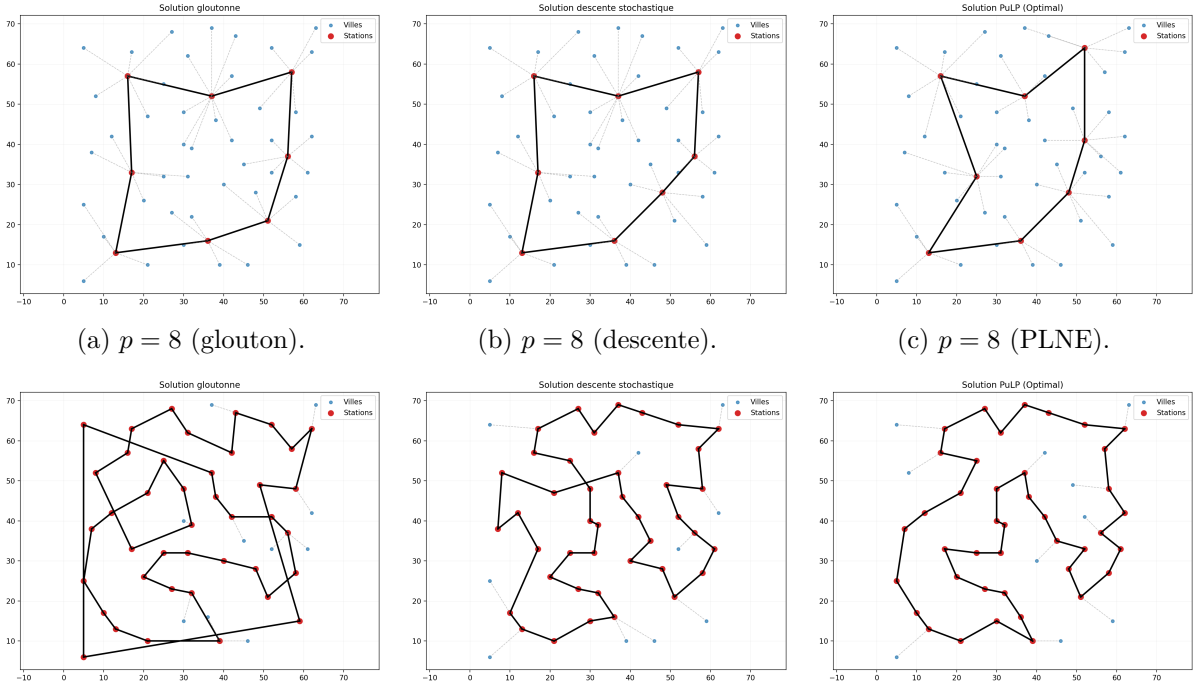
Sur `eil51` avec $p = 20$, la descente stochastique réduit sensiblement le coût et produit un anneau généralement plus régulier (moins de détours) tout en conservant une bonne couverture spatiale (figure 5.1).



(a) Glouton. (b) Après descente.

FIGURE 5.1 – Comparaison sur `eil51` avec $p = 20$: anneau (cycle) et rattachements en étoile.

La figure 5.2 illustre également deux configurations extrêmes : p faible ($p = 8$) et p élevé ($p = 41$). Quand p est grand, le coût métro domine et la qualité du cycle construit par plus-proche-voisin devient critique : la métaheuristique améliore nettement l'anneau, et la PLNE fournit la référence optimale lorsque le solveur termine.



(a) $p = 8$ (glouton). (b) $p = 8$ (descente). (c) $p = 8$ (PLNE).

(d) $p = 41$ (glouton). (e) $p = 41$ (descente). (f) $p = 41$ (PLNE).

FIGURE 5.2 – Comparaisons sur `eil51` pour $p = 8$ (haut) et $p = 41$ (bas).

5.3 Résultats synthétiques

Les tableaux 5.1 et 5.2 résument coûts (total, métro, marche) et temps. Le *gap* est l'écart relatif au coût optimal PLNE, lorsqu'il est disponible.

p	Méthode	Coût total	Métro	Marche	Temps (s)	Gap (%)
5	Glouton	91.964	57.727	34.237	0.000	40.67
5	Descente	65.378	18.248	47.130	0.010	0.00
5	PLNE (opt.)	65.378	18.248	47.130	1.589	0.00
8	Glouton	94.053	72.347	21.706	0.000	47.60
8	Descente	65.015	23.560	41.455	0.012	2.03
8	PLNE (opt.)	63.721	31.363	32.358	9.299	0.00
12	Glouton	104.982	101.653	3.329	0.000	68.55
12	Descente	65.132	38.717	26.415	0.013	4.57
12	PLNE (opt.)	62.286	40.097	22.189	0.449	0.00

TABLE 5.1 – Résultats sur `ulysses16` ($n = 16$, $\alpha = 1$).

p	Méthode	Coût total	Métro	Marche	Temps (s)	Gap (%)
8	Glouton	594.236	163.678	430.559	0.000	1.25
8	Descente	589.003	160.115	428.887	0.037	0.36
8	PLNE (opt.)	586.893	166.606	420.286	34.243	0.00
20	Glouton	549.603	313.182	236.422	0.001	–
20	Descente	482.040	233.842	248.198	0.059	–
41	Glouton	596.795	535.184	61.611	0.001	47.78
41	Descente	428.565	342.068	86.498	0.097	6.13
41	PLNE (opt.)	403.830	319.788	84.043	4.141	0.00

TABLE 5.2 – Résultats sur `eil51` ($n = 51$, $\alpha = 1$). Le PLNE n'a pas été résolu pour $p = 20$ dans le temps imparti.

5.4 Courbes coût/temps

Les figures suivantes donnent une vue d'ensemble : coûts (figure 5.3) et temps de calcul (figure 5.4). La métaheuristique améliore systématiquement l'heuristique gloutonne, tout en restant quasi instantanée sur ces tailles, tandis que le PLNE peut devenir coûteux selon p.

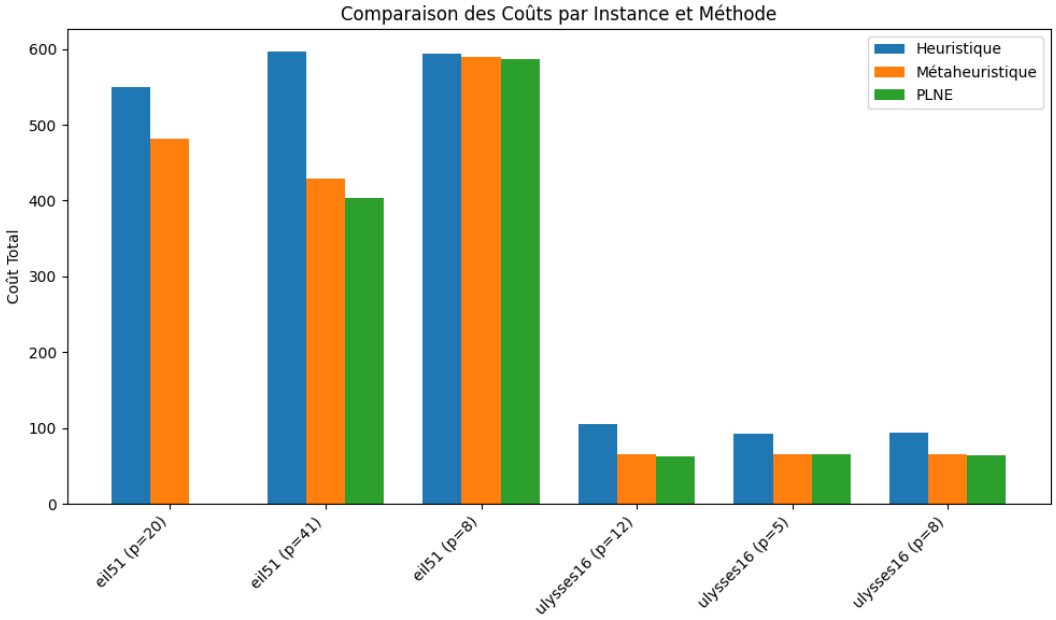


FIGURE 5.3 – Comparaison des coûts totaux par instance et par méthode.

5.5 Influence du paramètre p sur la difficulté du PLNE

Le nombre de sous-ensembles de p stations parmi n est $\binom{n}{p}$, quantité maximale lorsque $p \approx n/2$. Cette explosion combinatoire se reflète en pratique dans le temps de résolution du PLNE. Sur `ulysses16`, la figure 5.5 illustre une évolution non monotone du temps en fonction de p.

Pour `eil51`, des valeurs de p proches de $n/2$ rendent la résolution exacte délicate en temps raisonnable. La figure 5.6 montre un exemple de tentative de résolution qui ne converge pas rapidement vers une preuve d'optimalité.

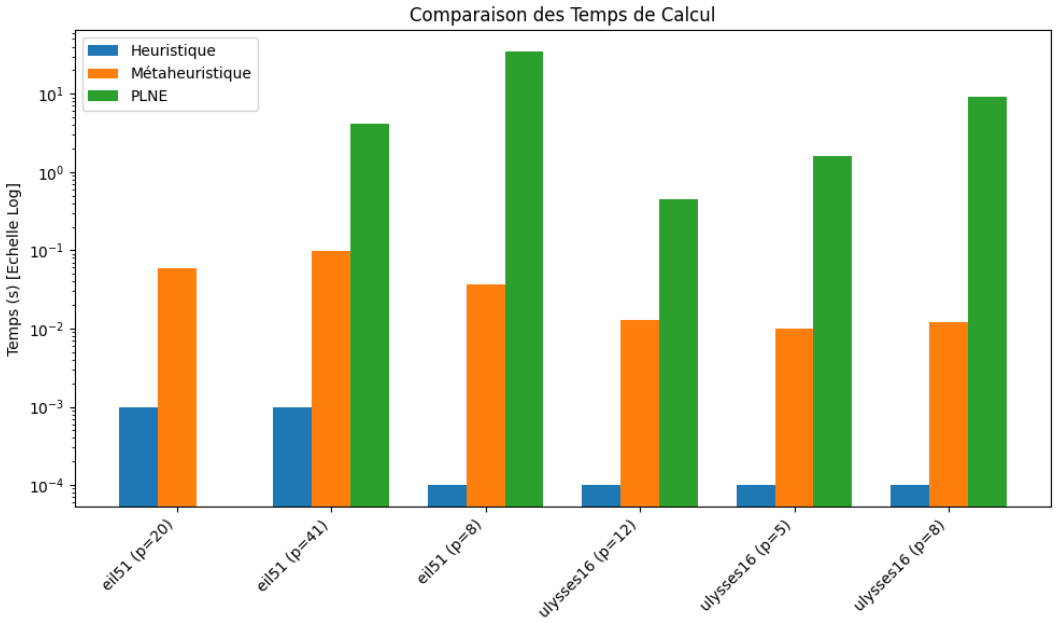


FIGURE 5.4 – Comparaison des temps de calcul (échelle logarithmique lorsque nécessaire).

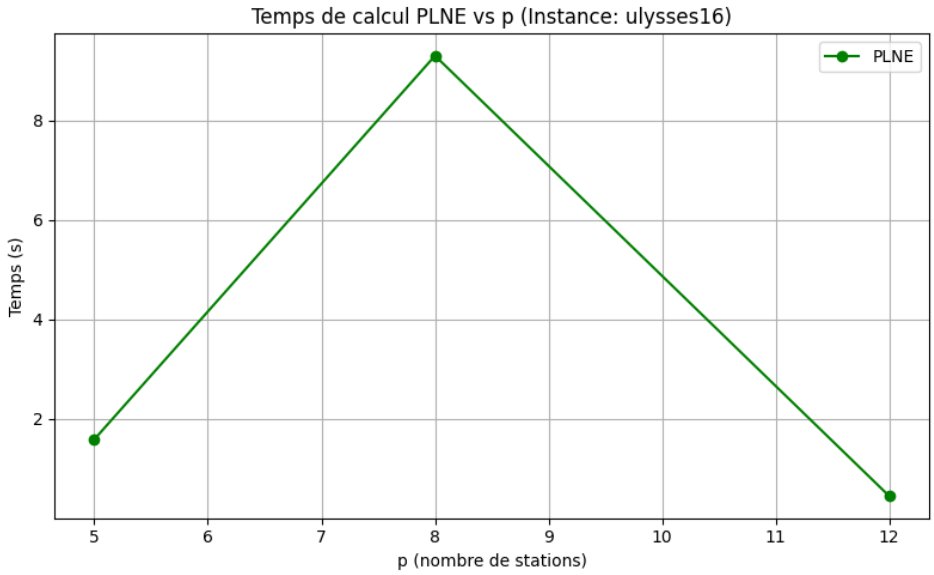


FIGURE 5.5 – Temps de calcul PLNE sur `ulysses16` en fonction de p.

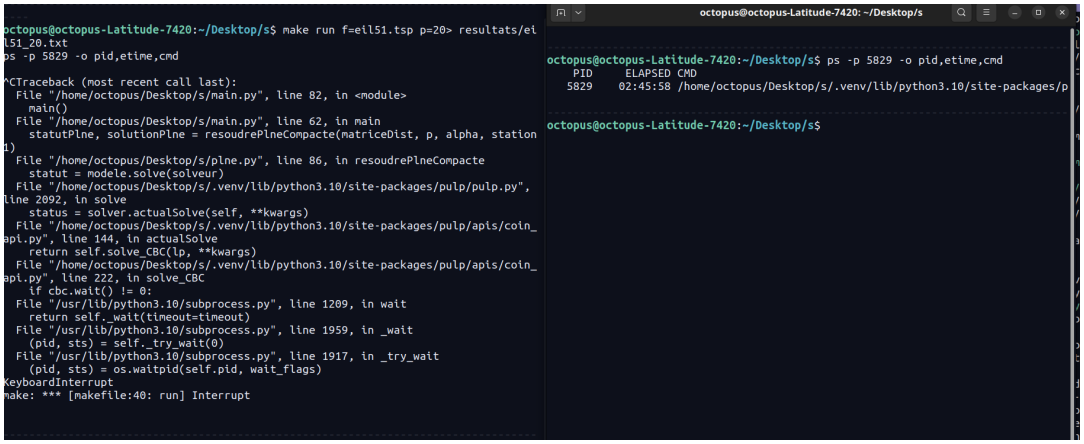


FIGURE 5.6 – Exemple d'exécution PLNE sur `eil51` avec $p = 21$: difficulté à obtenir une solution optimale certifiée dans un temps court.

Chapitre 6

Conclusion générale

Ce projet a permis d'étudier le problème Ring-Star à la fois sur le plan théorique et algorithmique. La NP-difficulté justifie l'usage d'une formulation exacte PLNE uniquement sur des instances ou paramètres favorables, tandis que l'heuristique et la métaheuristique offrent des solutions de bonne qualité en un temps très faible.

Les résultats montrent que la descente stochastique améliore nettement l'heuristique constructive, et se rapproche souvent de l'optimum lorsque celui-ci est disponible. Pour aller plus loin, des pistes naturelles consistent à : (i) renforcer la construction du cycle (par exemple via 2-opt/3-opt), (ii) utiliser des métaheuristiques plus exploratoires (recuit simulé, recherche tabou), et (iii) étudier des critères multi-objectifs séparant explicitement coût métro et accessibilité.