



ÉCOLE NATIONALE SUPÉRIEURE DE TECHNIQUES AVANCÉES

Compte rendu

Course d'avions

SOD321 : Optimisation Discrète

Réalisé par :

Eya JLASSI

Mohamed Aziz MHADHBI

Encadré par :

MME Sourour ELLOUMI

3A SOD

Année universitaire : 2024/2025

Remerciements

En premier lieu, nous remercions notre enseignante et encadrante Mme. Sourour ELLOUMI pour toutes les connaissances qu'elle nous a communiquées tout au long de l'élaboration de ce projet, pour sa grande disponibilité, et ses précieux conseils.

Nous remercions également toutes les personnes qui nous ont conseillées et relues lors de la rédaction de ce projet, notamment nos camarades de la 3^e année Techniques Avancées.

Enfin, nous adressons une pensée affective à nos parents, nos familles et nos amis pour leurs supports tout au long de ce projet.

Table des matières

| | |
|--|-----------|
| Introduction | 1 |
| 1 Modélisation du problème de voyageur de commerce | 3 |
| 1.1 Variables du Problème | 3 |
| 1.1.1 Variables du Problème | 3 |
| 1.1.2 Relation entre les variables | 4 |
| 1.2 Contraintes du PVC | 4 |
| 1.2.1 Contraintes de base | 4 |
| 1.2.2 Contraintes spécifiques à notre problème | 5 |
| 1.2.3 Contraintes relatives à la distance | 5 |
| 2 Contrainte de chemin élémentaire | 6 |
| 2.1 Formulation Polynomiale MTZ | 6 |
| 2.2 Formulation exponentielle DFJ | 7 |
| 3 Résultats | 8 |
| 3.1 Visualisation des Résultats Obtenus | 8 |
| 3.2 Comparaison des algorithmes | 10 |
| 3.3 Temps de résolution | 10 |
| 3.4 La valeur de la borne par relaxation continue | 10 |
| 3.5 Nombre de nœuds explorés par l’algorithme Branch-and-Cut | 10 |
| 3.6 Génération de contraintes | 11 |
| Conclusion | 13 |

Introduction

Ce projet vise à résoudre une version modifiée du problème classique du voyageur de commerce (PVC), inspirée du défi aérien de la Coupe Breitling 100/24. Dans ce contexte, plusieurs équipes de pilotes de petits avions sont chargées de parcourir la France en 24 heures, en réalisant autant de décollages et d’atterrissages que possible, avec pour objectif d’atteindre idéalement 100 étapes sur différents aérodromes. Les équipes commencent toutes leur périple depuis un aérodrome de départ spécifique et doivent atteindre un aérodrome d’arrivée prédéterminé en respectant certaines contraintes.

Ce rapport présente deux méthodes distinctes développées pour résoudre cette version du problème du voyageur de commerce. La première approche repose sur la formulation en contraintes polynomiales, basée sur la méthode Miller-Tucker-Zemlin (MTZ). Cette méthode permet de représenter les contraintes d’ordonnancement de manière linéaire en limitant les sous-tours non souhaités grâce à des variables supplémentaires. La deuxième approche utilise une formulation exponentielle appelée DFJ (Dantzig–Fulkerson–Johnson), qui exploite des techniques de génération de colonnes pour construire des solutions optimales en explorant de manière plus ciblée l’espace des solutions possibles.

Pour évaluer l’efficacité des deux approches, nous procédons à des comparaisons approfondies pour chaque instance du problème. Nous analysons notamment la valeur des bornes obtenues par relaxation continue pour chacune des formulations, le nombre de nœuds explorés par l’algorithme Branch-and-Cut du solveur utilisé, ainsi que le temps de calcul nécessaire pour aboutir à une solution satisfaisante. Ces indicateurs permettent de comparer les performances et la pertinence de chaque approche dans ce contexte.

Les données d’entrée nécessaires à la résolution du problème sont fournies sous forme de texte structuré, décrivant les éléments suivants :

- **n** : le nombre total d’aérodromes impliqués dans le parcours.
- **d** : l’identifiant de l’aérodrome de départ.
- **f** : l’identifiant de l’aérodrome d’arrivée.
- **Amin** : le nombre minimum d’aérodromes que l’équipe doit visiter au cours de son trajet.
- **Nr** : le nombre de régions administratives traversées dans le périmètre du parcours.
- **R** : la distance maximale qu’un avion peut parcourir sans atterrissage intermédiaire.
- **regions** : la région assignée à chaque aérodrome, pour tenir compte des contraintes régionales.

- **coords** : les coordonnées géographiques de chaque aéroport, permettant de calculer les distances inter-aéroports.
- **d** : une matrice de distances précalculées entre chaque paire d'aéroports, nécessaire pour évaluer le coût de chaque transition.

Cette structuration permet de modéliser efficacement le problème dans un cadre optimisé et de formuler des solutions adaptées au contexte spécifique du défi, en maximisant le nombre de parcours tout en respectant les contraintes de temps et de distance.

Chapitre 1

Modélisation du problème de voyageur de commerce

Pour résoudre efficacement ce problème, il est essentiel de commencer par une modélisation adéquate de sa structure globale, de spécifier précisément les contraintes, puis d'introduire des méthodes qui garantissent un chemin élémentaire et minimal.

Rappelons que l'objectif consiste à identifier le chemin le plus court reliant un point de départ à un point d'arrivée, en passant par un nombre minimum A_{\min} d'aérodromes et en visitant chaque région au moins une fois.

De plus, une distance maximale R est imposée, qui représente la limite de vol sans escale pour un avion.

La fonction objectif de ce problème s'écrit ainsi :

$$\min_x \sum_i \sum_j d_{ij} x_{ij}$$

avec : d_{ij} représente la distance qui sépare l'aérodrome i de j .

1.1 Variables du Problème

Dans le cadre de ce problème, deux variables binaires principales sont définies : l'une spécifie le trajet parcouru, et l'autre indique si un aérodrome a été visité ou non.

1.1.1 Variables du Problème

Ainsi, on définit la variable binaires x_{ij} qui représente les arêtes choisies, prenant la valeur 1 si l'on passe de l'aérodrome i à l'aérodrome j et 0 sinon.

$$x_{ij} = \begin{cases} 1 & \text{si on va de l'aérodrome } i \text{ à l'aérodrome } j \\ 0 & \text{sinon} \end{cases}$$

On définit ensuite la variable binaire y_i qui nous dit si on a visité l'aérodrome i . Cette variable prend la valeur 1 si on visite l'aérodrome i .

$$y_i = \begin{cases} 1 & \text{on visite l'aérodrome } i \\ 0 & \text{sinon} \end{cases}$$

Les indices $i \in \{1, 2, \dots, n\}$ et $j \in \{1, 2, \dots, n\}$ avec n le nombre total des aérodromes.

1.1.2 Relation entre les variables

Il est nécessaire d'imposer une contrainte initiale pour établir un lien entre les variables x et y :

$$\sum_j x_{ji} = y_i \quad \forall i \in \{1, 2, \dots, n\}$$

De plus, considérons que le parcours commence à l'aérodrome d et se termine à l'aérodrome f . Ainsi, nous avons :

$$y[d] = 1 \text{ et } y[f] = 1$$

L'ajout de la variable y nous a aidé à formaliser la contrainte de la disparité géographique ainsi que la contrainte du nombre minimale d'aérodromes à visiter.

1.2 Contraintes du PVC

Nous avons désormais exposé les contraintes essentielles pour le problème, à l'exception d'une : celle des sous-tours. Il nous faut maintenant définir une contrainte permettant d'éviter la formation de sous-tours. C'est précisément cette contrainte qui constitue la motivation principale de ce projet.

Dans un premier temps, nous proposons des formulations pour les contraintes du problème classique du voyageur de commerce.

1.2.1 Contraintes de base

La première contrainte garantit qu'il ne peut y avoir qu'un seul arc entrant et un seul arc sortant pour chaque aérodrome :

$$\begin{aligned} \sum_i x_{ij} &\leq 1 \quad \forall j \in \{1, 2, \dots, n\} \\ \sum_j x_{ij} &\leq 1 \quad \forall i \in \{1, 2, \dots, n\} \end{aligned}$$

La deuxième contrainte, qui est également classique, concerne la conservation du flot. Cette contrainte assure que le nombre d'arcs sortants est égal au nombre d'arcs entrants pour tout aérodrome $i \in \{1, 2, \dots, n\}$:

$$\sum_j x_{ij} - x_{ji} = \begin{cases} 1 & \text{si } i = d \\ -1 & \text{si } i = f \\ 0 & \text{sinon} \end{cases}$$

Ces conditions sont analogues aux :

$$\begin{aligned}\sum_j x_{dj} &= 1 \\ \sum_i x_{if} &= 1 \\ \sum_j x_{ij} &= y[i] \quad \forall i \in \{1, 2, \dots, n\} \\ \sum_j x_{ji} &= y[i] \quad \forall i \in \{1, 2, \dots, n\}\end{aligned}$$

1.2.2 Contraintes spécifiques à notre problème

Nous voulons garantir la visite d'au moins A_{\min} aérodrômes. Cette contrainte se formule aisément grâce à l'introduction de la variable y :

$$\sum_i y_i \geq A_{\min}$$

Ensuite, pour assurer une disparité géographique, nous souhaitons que certaines régions soient visitées au moins une fois. Pour modéliser cette contrainte, nous introduisons cette contrainte :

$$\sum_{i \in R_k} y_i \geq 1 \quad \forall k \in \{1, \dots, N_r\}$$

où : R_k représente la région k .

1.2.3 Contraintes relatives à la distance

La dernière contrainte que nous devons ajouter concerne la distance maximale que peut parcourir l'avion sans effectuer d'atterrissage intermédiaire :

$$x_{ij} d_{ij} \leq R \quad \forall i, j \in \{1, \dots, n\}$$

Nous avons désormais exposé les contraintes essentielles pour le problème, à l'exception d'une : celle des sous-tours. Il est maintenant nécessaire de définir une contrainte permettant d'éviter la formation de sous-tours, qui constitue la motivation principale de ce projet.

Chapitre 2

Contrainte de chemin élémentaire

Dans cette section, nous allons aborder les contraintes nécessaires pour éliminer les sous-tours. Pour ce faire, nous avons introduit deux algorithmes : l'un ayant une complexité polynomiale et l'autre une complexité exponentielle.

2.1 Formulation Polynomiale MTZ

La première formulation des contraintes pour éliminer les sous-tours est la formulation MTZ (Miller-Tucker-Zemlin) décrite par L. Taccari [1]. Cette approche repose sur l'introduction d'une variable auxiliaire t , qui représente la position d'un aéroport dans le parcours.

Dans cette formulation, chaque aéroport i est associé à une variable t_i , qui indique l'ordre dans lequel cet aéroport est visité dans le trajet. Les valeurs de t_i sont définies de manière à ce qu'elles soient cohérentes avec l'ordre de visite des aéroports. Plus précisément, la contrainte principale pour éviter la formation de sous-tours s'exprime comme suit :

$$t_i - t_j + (n - 1) \cdot x_{ij} \leq n - 2 \quad \forall i, j \in \{1, 2, \dots, n\}, i \neq j$$

où n est le nombre total d'aéroports. Cette contrainte stipule que si le trajet inclut un arc de i à j (c'est-à-dire si $x_{ij} = 1$), alors la différence de position entre i et j doit respecter une certaine condition. Plus précisément, elle garantit que les aéroports dans un sous-tour ne peuvent pas avoir des positions qui violent l'ordre imposé par les variables t .

L'idée derrière cette formulation est que si $x_{ij} = 1$ (c'est-à-dire qu'il existe un trajet direct de i à j), alors t_i doit nécessairement être inférieur à t_j pour éviter les sous-tours. Si un sous-tour se forme, cela entraînerait une violation de cette contrainte, ce qui permet de les éliminer.

La formulation finale du PVC avec les contraintes polynomiales est la suivante :

$$\begin{array}{ll}
\min_x & \sum_i \sum_j d_{ij} x_{ij} \\
\text{s.t.} & \sum_j x_{ij} = y_i \quad \forall i \neq d, f \\
& \sum_j x_{ij} = y_i \quad \forall j \neq d, f \\
& \sum_j x_{dj} = 1 \\
& \sum_j x_{jf} = 1 \\
& \sum_i y_i \geq A_{\min} \\
& \sum_{i \in R_k} y_i \geq 1 \quad \forall k \in 1 \dots N_r \\
& x_{ij} d_{ij} \leq R \quad \forall i, j \\
& t_j \geq t_i + 1 + (n - 1)(x_{ij} - 1) \quad \forall i, j, i \neq d, j \neq f
\end{array}$$

2.2 Formulation exponentielle DFJ

Pour le problème du voyageur de commerce (TSP), des formulations efficaces ont été développées, même avec un nombre exponentiel de contraintes. On peut donc établir une autre formulation pour éliminer les sous-tours en se basant sur les contraintes d'élimination des sous-tours classiques de Dantzig–Fulkerson–Johnson (Dantzig, Fulkerson et Johnson, 1954) qui est représentée comme suit :

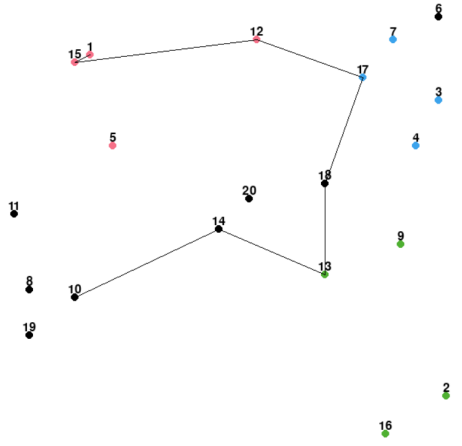
$$\sum_{(i,j) \in A(S)} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V_{st}, |S| \geq 2.$$

Ces inégalités empêchent la formation de sous-tours dans chaque sous-ensemble S en garantissant que le nombre d'arcs sélectionnés dans S est inférieur au nombre de nœuds présents dans ce sous-ensemble.

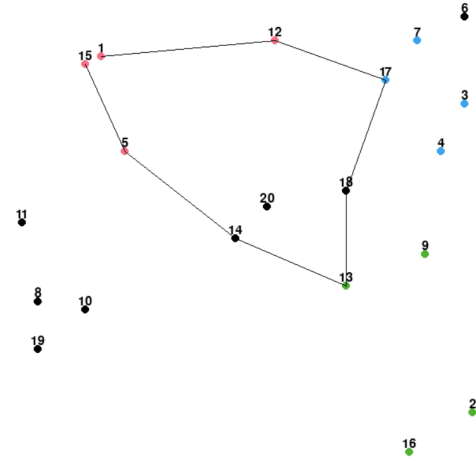
Cette formulation comporte $O(m)$ variables et $O(2^n)$ contraintes.

La formulation finale de notre problème avec un nombre exponentiel de contraintes s'écrit de cette façon :

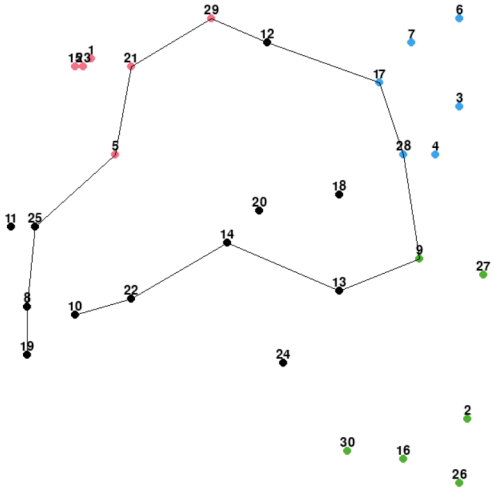
$$\begin{array}{ll}
\min_x & \sum_i \sum_j d_{ij} x_{ij} \\
\text{s.t.} & \sum_j x_{ij} = y_i \quad \forall i \neq d, f \\
& \sum_j x_{ij} = y_i \quad \forall j \neq d, f \\
& \sum_j x_{dj} = 1 \\
& \sum_j x_{jf} = 1 \\
& \sum_i y_i \geq A_{\min} \\
& \sum_{i \in R_k} y_i \geq 1 \quad \forall k \in 1 \dots N_r \\
& x_{ij} d_{ij} \leq R \quad \forall i, j \\
& \sum_{(i,j) \in A(S)} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V_{st}, |S| \geq 2
\end{array}$$



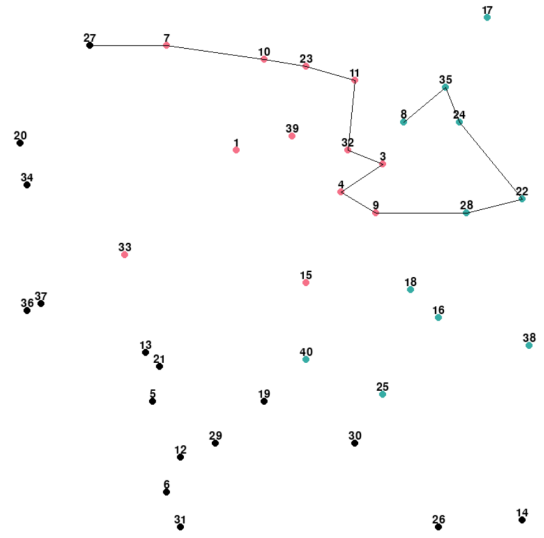
(a) INSTANCE 3 : $n=20$



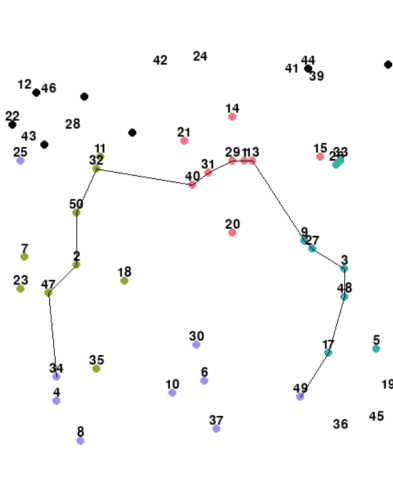
(b) INSTANCE 4 : $n=20$



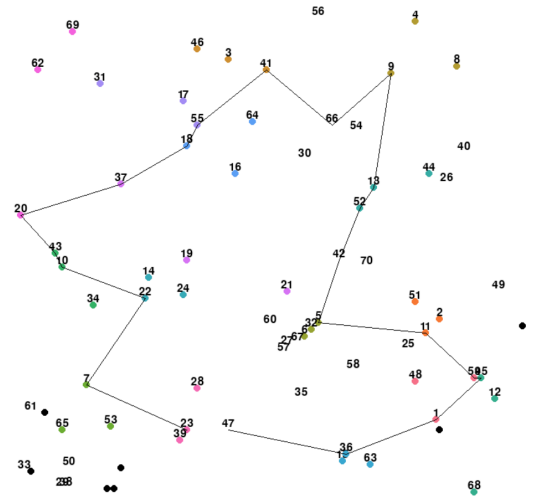
(c) INSTANCE 5 : $n=30$



(d) INSTANCE 6 : $n=40$



(e) INSTANCE 5 : $n=50$



(f) INSTANCE 6 : $n=70$

3.2 Comparaison des algorithmes

La formulation de **Miller-Tucker-Zemlin (MTZ)** est simple, avec un nombre polynomial de contraintes et de variables, mais souffre d'une relaxation linéaire de faible qualité, rendant sa résolution inefficace pour de grands problèmes TSP. En revanche, l'approche **Dantzig-Fulkerson-Johnson (DFJ)** utilise un nombre exponentiel de coupes pour éliminer les sous-tours, ce qui complexifie sa structure mais offre une relaxation linéaire plus serrée, accélérant la résolution pour des instances de grande taille grâce aux techniques modernes de génération de coupes.

3.3 Temps de résolution

| Formulation | Inst6_1 | Inst20_1 | Inst20_2 | Inst20_3 | Inst30_1 | Inst40_1 | Inst50_1 | Inst70_1 |
|-------------|---------|----------|----------|----------|----------|----------|----------|----------|
| MTZ | 0.02 | 11.03 | 1.52 | 10.38 | 35.81 | 3.38 | 1353.27 | 38407.21 |
| DFJ | 0.02 | 38244.9 | 9516.36 | - | - | - | - | - |

TABLE 3.1 – Temps de résolution

- ◊ **Observations** : Pour la formulation avec un nombre polynomial de contraintes, le problème peut être résolu en un temps raisonnable jusqu'à l'instance de 50 aéroports. D'autre part, la formulation en nombre exponentiel de contraintes ne donne un résultat dans un temps raisonnable que pour la petite instance de 6 aéroports alors que pour les autres instances cela prend un temps énorme qu'on peut pas le déterminer.

3.4 La valeur de la borne par relaxation continue

| Formulation | Inst6_1 | Inst20_1 | Inst20_2 | Inst20_3 | Inst30_1 | Inst40_1 | Inst50_1 | Inst70_1 |
|-------------|---------|----------|----------|----------|----------|----------|----------|----------|
| MTZ | 8.27 | 36.32 | 41.11 | 39.11 | 53.41 | 72.31 | 70.4 | 141.92 |
| DFJ | 10.5 | 40 | 53 | 40 | - | - | - | .. |

TABLE 3.2 – Valeurs de la borne par relaxation continue

- ◊ **Observations** : D'après les résultats obtenus, nous constatons que pour toutes les instances testées la relaxation continue du problème en formulation polynomiale est toujours moins bonne que les autres. Nous pouvons donc confirmer que ces formulations en nombre exponentiel de contraintes qui sont plus lourdes en contraintes offrent des meilleures bornes.

3.5 Nombre de nœuds explorés par l'algorithme Branch-and-Cut

| Formulation | Inst6_1 | Inst20_1 | Inst20_2 | Inst20_3 | Inst30_1 | Inst40_1 | Inst50_1 | Inst70_1 |
|-------------|---------|----------|----------|----------|----------|----------|----------|----------|
| MTZ | 1 | 104890 | 11800 | 109194 | 143958 | 3485 | 690618 | 1403403 |
| DFJ | 1 | 1625 | 1776 | - | - | - | - | - |

TABLE 3.3 – Noeuds explorés par l’algorithme Branch and Cut

- ◊ **Observations** : Pour la modélisation en nombre exponentiel de contraintes et comme la résolution ne peut pas être faite dans un temps raisonnable pour la plupart des instances, nous n’arrivons pas à déterminer le nombre de noeuds explorés par l’algorithme Branch and Cut., mais au moins pour les instances testés , on observe que c’est beaucoup moins petit que la formulation **MTZ**. En revanche, pour la modélisation en nombre polynomial de contraintes, on remarque l’instance 4 (instance_20_3) est la plus complexe à résoudre en nombre de noeuds de Branch and Cut par rapport aux autres instances de 20 aérodromes, et lorsque on augmente le nombre d’aérodromes à 70 (instance 8) le nombre des noeuds explorés est énorme.

3.6 Génération de contraintes

Puisque la contrainte d’élimination de sous tours dans la modélisation **DFJ** entraîne un nombre exponentiel de contraintes, nous avons essayé de procéder à un algorithme de génération de contraintes.

Il faut donc résoudre le sous problème qui renvoie les sous tours d’un optimum \tilde{x} . Puis, il faut ajouter ces sous-tours et appliquer la contrainte d’élimination de sous tours sur eux. Le sous-problème s’écrit de la façon suivante :

$$\max \sum_{i,j} z_i z_j \tilde{x}_{i,j} - \left(\sum_i z_i - 1 \right)$$

sous les contraintes :

$$\begin{aligned} \sum_i z_i &\geq 1, \\ \sum_i z_i &\leq n - 1, \\ z_i &\in \{0, 1\}. \end{aligned}$$

Les étapes de la génération de contraintes sont les suivantes :

- Résoudre le problème principal.
- Résoudre le sous-problème afin d’identifier un ensemble S pour lequel la contrainte d’élimination de sous-tour associée n’est pas vérifiée par la solution du problème maître. Effectivement, si le sous-problème a une valeur optimale supérieure strictement à 0 alors l’ensemble S recherché est constitué des sommets i dont les z_i valent 1. L’ensemble S ainsi exhibé viole la contrainte.
- Rajouter cette contrainte au problème maître afin de le forcer à la vérifier.
- Résoudre à nouveau le problème principal.

On répète jusqu'à ce que le sous-problème ait une valeur optimale négative ou nulle, ce qui assure l'inexistence de sous-ensemble S violant la contrainte d'élimination de sous-tour. La solution optimale sera donc la dernière fournie par le problème maître modifié. Nous avons testé cette méthode sur quelques instances qui ont pu donner des résultats dans un temps assez raisonnable. Les résultats fournis sont présentés dans le tableau suivant.

| Formulation | Inst6_1 | Inst20_2 |
|-------------|---------|----------|
| ValOPT | 12 | 102 |
| ValBorne | 8 | 59 |
| TempsRésol | 0.02 | 66.35 |

TABLE 3.4 – Test de la méthode de génération de contraintes

Conclusion

Le projet illustre l'équilibre crucial entre la qualité des solutions et la rapidité d'exécution des algorithmes d'optimisation, comme le montre la comparaison entre les modèles MTZ et DFJ. La formulation **DFJ**, bien que plus complexe et nécessitant davantage de temps en raison de la génération de coupes, produit des bornes plus serrées et des solutions plus précises. À l'inverse, la formulation **MTZ** est plus rapide grâce à sa simplicité et son nombre polynomial de contraintes, mais au prix de solutions de qualité inférieure en termes de relaxation linéaire. Ce compromis entre vitesse et précision est central dans le choix des algorithmes, influençant leur application selon les besoins spécifiques : solutions rapides pour les situations urgentes, et solutions précises lorsque l'optimisation est primordiale.