

Projet MOGPL : optimisation robuste dans l'incertain total

Kitoko David & Rigollier Simon

30 novembre 2024

Table des matières

1	Introduction	1
2	Mise en situation 1 : Sélection de projets avec incertitude et contraintes budgétaires	2
3	Linéarisation des critères MaxMin et MinMax Regret	2
3.1	Formulation du Problème sur le critère MaxMin	2
3.2	Formulation du Problème sur le critère MinMax Regret	3
3.3	Visualisation graphique des solutions x^* et x'^* sur les critères MaxMin et MinMax regret	4
3.4	Analyse des temps de résolution des méthodes MaxMin et MinMax Regret	4
4	Linéarisation du critère MaxOWA et MinOWA des Regrets	5
4.1	Analyse de la Solution Optimale d'un Programme Linéaire avec Variables Binaires	5
4.2	Dualité et Calcul des Composantes du Vecteur $L(z)$	6
4.3	Formulation subsidiaire de l'OWA	7
4.4	MaxOwa : Formulation du programme linéaire	8
4.5	MinOwa : Formulation du programme linéaire	9
4.6	Analyse des temps de résolution des méthodes MaxOWA et MinOWA des Regrets	10
5	Introduction sur les graphes : Recherche du chemin le plus rapide	10
5.1	Variables de décision	10
5.2	Fonction objectif	11
5.3	Contraintes de flux	11
5.4	Formulation complète du programme linéaire	11
6	Mise en situation 2 : Application à la recherche d'un chemin robuste dans un graphe	11
6.1	Résolution des Chemins les Plus Rapides par Programmation Linéaire	12
6.2	Recherche d'un Chemin Robuste	13
6.3	Analyse de l'évolution du temps de résolution pour la détermination d'un chemin optimal	17
7	Conclusion	18

1 Introduction

Les problèmes d'optimisation combinatoire consistent à maximiser ou minimiser une fonction objectif f sur un ensemble X de solutions définies par des contraintes ou des propriétés structurelles.

Cependant, dans de nombreux contextes, l'évaluation des solutions est incertaine et dépend de plusieurs scénarios possibles $S = \{s_1, \dots, s_n\}$. Chaque solution $x \in X$ est alors associée à un vecteur $z(x) = (z_1(x), \dots, z_n(x))$, où $z_i(x)$ représente son évaluation dans le scénario s_i .

En absence d'information sur la probabilité des scénarios, on parle d'incertitude totale, nécessitant des approches robustes pour identifier des solutions performantes dans tous les cas envisageables. Le problème devient alors de déterminer une solution optimale ou satisfaisante selon différents critères de robustesse.

2 Mise en situation 1 : Sélection de projets avec incertitude et contraintes budgétaires

Prenons une situation où l'on doit sélectionner des projets parmi une liste de 10, avec un budget limité à 100 unités. Chaque projet a une utilité variable selon deux scénarios possibles (s_1 et s_2), représentant des évolutions différentes de l'environnement socio-économique. L'objectif est de choisir un sous-ensemble de projets qui maximise l'utilité totale, tout en respectant les contraintes budgétaires. Pour cette tâche, quatre critères de robustesse sont envisagés :

<i>projets</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>
<i>utilité dans s_1</i>	<i>70</i>	<i>18</i>	<i>16</i>	<i>14</i>	<i>12</i>	<i>10</i>	<i>8</i>	<i>6</i>	<i>4</i>	<i>2</i>
<i>utilité dans s_2</i>	<i>2</i>	<i>4</i>	<i>6</i>	<i>8</i>	<i>10</i>	<i>12</i>	<i>14</i>	<i>16</i>	<i>18</i>	<i>70</i>
<i>coût (en Keuros)</i>	<i>60</i>	<i>10</i>	<i>15</i>	<i>20</i>	<i>25</i>	<i>20</i>	<i>5</i>	<i>15</i>	<i>20</i>	<i>60</i>

FIGURE 1 – L'utilité des 10 projets dans 2 scénarios ainsi que les coûts des projets

- **MaxMin** : Maximiser la pire performance ($\min_i z_i(x)$) parmi les scénarios.
- **MinMax Regret** : Minimiser le plus grand regret, où le regret est défini comme l'écart entre la meilleure décision possible et celle choisie.
- **MaxOWA (Ordered Weighted Average)** : Maximiser une moyenne pondérée décroissante des utilités ($\sum_i w_i z_{(i)}(x)$), mettant davantage l'accent sur les scénarios défavorables.
- **MinOWA des Regrets** : Minimiser une moyenne pondérée décroissante des regrets, en priorisant la réduction des écarts dans les scénarios critiques.

Ces approches permettent de prendre en compte les incertitudes liées aux scénarios et offrent différentes perspectives pour trouver une solution robuste. Les résultats obtenus varient selon le critère appliqué, chacun mettant l'accent sur un aspect particulier des compromis entre scénarios.

3 Linéarisation des critères MaxMin et MinMax Regret

3.1 Formulation du Problème sur le critère MaxMin

Soit un problème à objectifs multiples, où $z_1(x), z_2(x), \dots, z_k(x)$ sont les fonctions objectifs définies en fonction des décisions x . Le critère **MaxMin** consiste à résoudre :

$$\max_{x \in X} \min_{i=1, \dots, k} z_i(x).$$

Pour linéariser ce problème, nous introduisons une variable auxiliaire θ représentant le minimum des performances. Le problème devient alors :

$$\max \theta, \quad \text{sous les contraintes } z_i(x) \geq \theta, \forall i = 1, \dots, k.$$

Dans l'exemple donné, le problème s'écrit avec :

- **Variables de décision** : x_1 (continu, représentant θ) et x_2, x_3, \dots, x_{11} (binaires).
- **Fonction objectif** : Maximiser x_1 .

— **Contraintes :**

$$\begin{aligned}x_1 - 70x_2 - 18x_3 - 16x_4 - \dots - 2x_{11} &\leq 0, \\x_1 - 2x_2 - 4x_3 - 6x_4 - \dots - 70x_{11} &\leq 0, \\60x_2 + 10x_3 + 15x_4 + \dots + 60x_{11} &\leq 100.\end{aligned}$$

3.1.1 Résultats et Commentaires

— **Solution optimale x^* :** Les valeurs retournées pour x_1, \dots, x_{11} sont :

$$\begin{aligned}x_1 &= 66.0, \\x_2 &= 0.0, \\x_3 &= 1.0, \\&\vdots \\x_{11} &= 0.0.\end{aligned}$$

— **Valeur de la fonction objectif :** La valeur optimale de θ^* , correspondant au critère **MaxMin**, est donnée par :

$$\theta^* = 66.0.$$

— **Vecteur image $z(x^*)$:** Le vecteur image correspond à l'évaluation de la solution optimale x^* selon les différents critères ou scénarios considérés dans le problème. Il regroupe les valeurs des fonctions objectifs $z_1(x), z_2(x), \dots, z_k(x)$ pour la solution x^* . Dans ce cas, le vecteur image est donné par :

$$z(x^*) = (z_1(x^*), z_2(x^*)) = (66.0, 66.0).$$

3.2 Formulation du Problème sur le critère MinMax Regret

Soit un problème à objectifs multiples où les fonctions objectifs $z_1(x)$ et $z_2(x)$ dépendent des décisions x . Le critère **MinMax Regret** consiste à minimiser le regret maximal défini comme suit :

$$R_i(x) = \max_{s \in S} (\text{utilité optimale dans le scénario } s_i - z_i(x))$$

et l'objectif est :

$$\min_x \max (R_1(x), R_2(x)).$$

Pour linéariser ce problème, nous maximisons d'abord chaque critère indépendamment, puis nous minimisons le regret maximal entre ces deux solutions :

1. ****Maximisation des critères individuels** :**

$$x_1^* = \max_x z_1(x), \quad x_2^* = \max_x z_2(x).$$

2. ****Calcul du regret maximal** :** Le regret pour chaque critère est calculé comme la différence entre l'utilité maximale dans chaque scénario et la valeur obtenue pour chaque évaluation des scénarios $z_1(x)$ et $z_2(x)$.

3. ****Minimisation du regret maximal** :** Le problème de minimisation du regret maximal peut être formulé comme suit :

$$\min_x \max \left(\max_{s \in S} (\text{utilité optimale dans } s_1 - z_1(x)), \max_{s \in S} (\text{utilité optimale dans } s_2 - z_2(x)) \right)$$

Résultats et Commentaires

— **Solution optimale x'^* pour maximiser $z_1(x)$:** La solution optimale pour maximiser $z_1(x)$ est donnée par :

$$x_1^* = [0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0]$$

La valeur est $z_1^* = 112.0$.

- **Solution optimale x'^* pour maximiser $z_2(x)$** : La solution optimale pour maximiser $z_2(x)$ est donnée par :

$$x_2^* = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0]$$

La valeur est $z_2^* = 118.0$.

- **Valeur du regret MaxMin** : La fonction objectif du problème de regret MaxMin consiste à minimiser le maximum des regrets, $R_1(x)$ et $R_2(x)$. Le regret maximal minimisé est de :

$$\max(R_1(x), R_2(x)) = 118.0.$$

- **Vecteur image $z(x'^*)$** : Le vecteur image associé à la solution optimale x'^* est donné par :

$$z(x'^*) = (z_1(x'^*), z_2(x'^*)) = (112.0, 118.0).$$

3.3 Visualisation graphique des solutions x^* et x'^* sur les critères MaxMin et MinMax regret

- **Solution x^*** : Le vecteur image est donné par $z(x^*) = (z_1(x^*), z_2(x^*)) = (66.0, 66.0)$, qui est la solution optimale du critère **MaxMin**.
- **Solution x'^*** : Le vecteur image est donné par $z(x'^*) = (z_1(x'^*), z_2(x'^*)) = (112.0, 118.0)$, qui est la solution optimale du critère **MinMax Regret**.

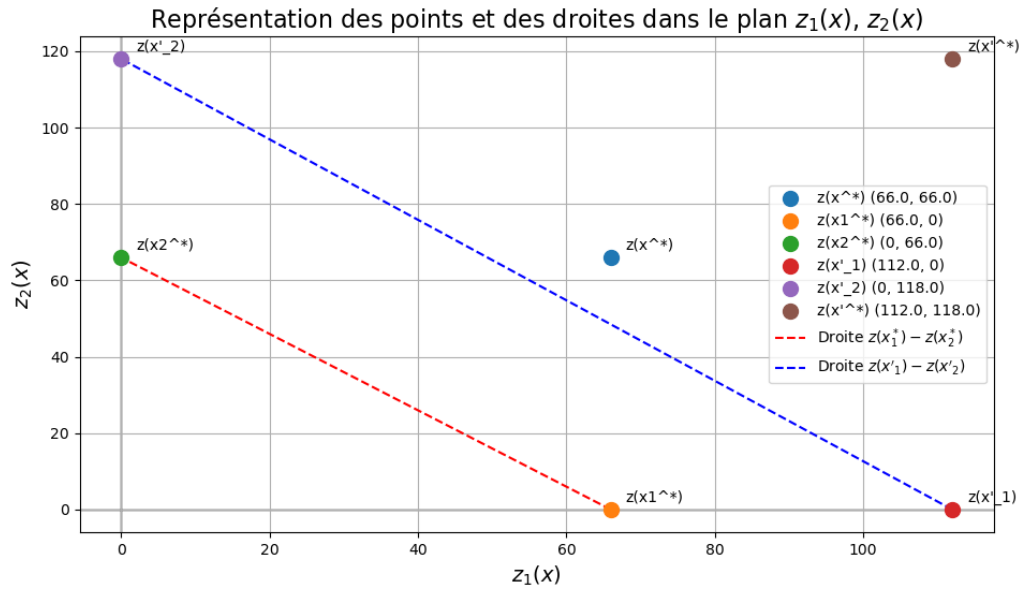


FIGURE 2 – Visualisation graphique des solutions x^* et x'^* sur les critères **MaxMin** et **MinMax Regret**

Les solutions x^* et x'^* ne se trouvent pas sur les droites reliant les points $z(x1^*)$ et $z(x2^*)$, ni sur celle reliant $z(x'_1)$ et $z(x'_2)$. Cela montre qu'elles ne peuvent pas être obtenues par une simple combinaison linéaire positive des points correspondant aux extrêmes de $z_1(x)$ et $z_2(x)$.

3.4 Analyse des temps de résolution des méthodes MaxMin et MinMax Regret

Les graphiques suivants comparent les temps moyens de résolution des deux méthodes pour différentes tailles de problème, avec un nombre de scénarios n variant entre 5, 10 et 15, et un nombre de projets p variant entre 10, 15 et 20.

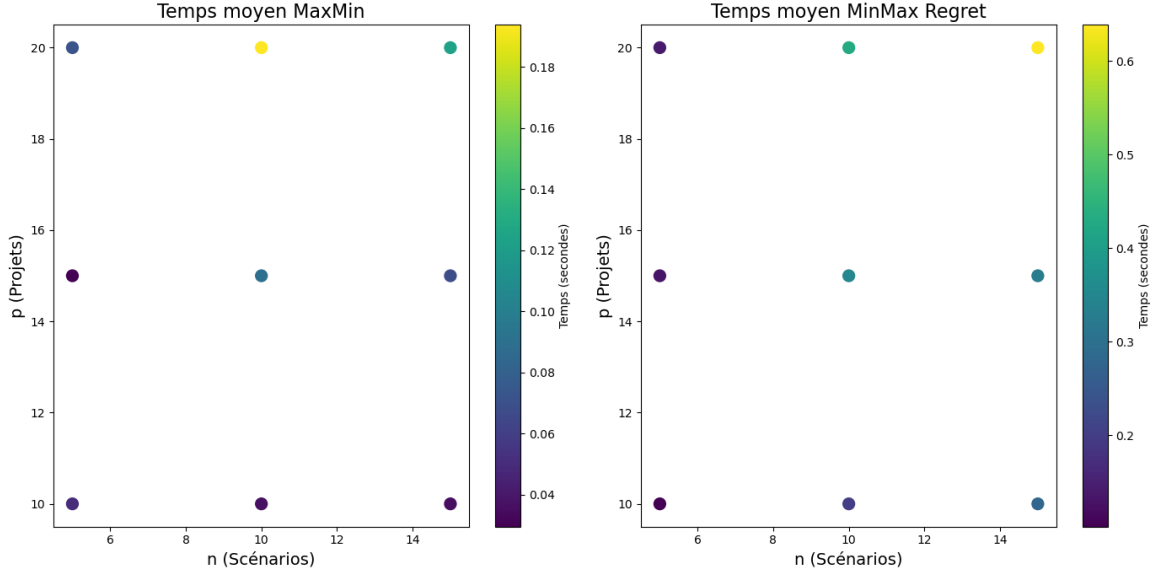


FIGURE 3 – Temps moyen de résolution pour la méthode **MaxMin** et **MinMax Regret**

Analyse et Conclusion

Les deux graphiques illustrent les temps moyens de résolution pour les critères **MaxMin** et **MinMax Regret** en fonction des paramètres n (scénarios) et p (projets), montrant des impacts distincts selon les critères. Pour le critère **MaxMin**, les temps de calcul restent globalement faibles, même avec l'augmentation de n et p , avec une légère hausse pour les configurations les plus complexes ($n = 10$, $p = 20$). En revanche, pour le critère **MinMax Regret**, l'augmentation de n et p entraîne des temps significativement plus élevés, atteignant jusqu'à **0,6 seconde** dans les cas les plus complexes. L'impact combiné du nombre de scénarios et de projets est donc bien plus marqué pour **MinMax Regret**, révélant une plus grande sensibilité de ce critère à la taille du problème. Ainsi, le critère **MaxMin** se montre plus rapide et robuste face à une complexité croissante, tandis que **MinMax Regret**, bien que plus coûteux en temps de calcul, peut être privilégié lorsque la minimisation du regret est prioritaire, malgré des contraintes de résolution.

4 Linéarisation du critère MaxOWA et MinOWA des Regrets

Soit $z \in \mathbb{R}^n$ un vecteur de taille n , dont les éléments sont représentés par $z = (z_1, z_2, \dots, z_n)$. Nous définissons pour chaque k un vecteur $L(z) = (L_1(z), L_2(z), \dots, L_n(z))$, dont la k -ième composante est la somme des k plus petits éléments de z après tri croissant, soit :

$$L_k(z) = \sum_{i=1}^k z(i)$$

où $z(i)$ désigne les composantes de z triées dans l'ordre croissant,

4.1 Analyse de la Solution Optimale d'un Programme Linéaire avec Variables Binaires

Pour l'optimisation de cette fonction, on considère un problème linéaire sous forme de programme avec des variables binaires. L'objectif est de minimiser la somme des produits $a_{ik}z_i$, où a_{ik} détermine si l'élément z_i est sélectionné dans la somme. Le programme linéaire correspondant est formulé comme suit :

$$\min \sum_{i=1}^n a_{ik}z_i$$

sous la contrainte :

$$\sum_{i=1}^n a_{ik} = k, \quad a_{ik} \in \{0, 1\}, \quad i = 1, \dots, n$$

Ici, a_{ik} est une variable binaire, prenant la valeur 1 si l'élément z_i est sélectionné pour faire partie de la somme et 0 sinon. La contrainte garantit que exactement k éléments sont choisis parmi les n disponibles.

4.1.1 Analyse et Conclusion

Le programme linéaire décrit ci-dessus cherche à minimiser la somme des produits $a_{ik}z_i$ en sélectionnant les k éléments les plus significatifs de z . Chaque variable binaire a_{ik} permet de choisir un élément du vecteur z , et la contrainte sur le nombre total de choix assure que la somme est obtenue uniquement à partir des k éléments les plus pertinents, selon le critère de minimisation.

Ainsi, la solution optimale du programme linéaire consiste à choisir les k plus petits éléments de z (ceux qui minimisent la somme), ce qui correspond à la somme des k premiers éléments de z après leur tri croissant. Cette sélection est précisément ce que nous désignons par $L_k(z)$, ce qui démontre que $L_k(z)$ représente la valeur optimale de ce programme linéaire dans le cadre de la minimisation.

4.2 Dualité et Calcul des Composantes du Vecteur $L(z)$

Le problème primal associé à la fonction $L_k(z)$ peut être formulé comme un programme linéaire relaxé, où les variables binaires a_{ik} sont remplacées par des variables continues. Le programme linéaire est formulé comme suit :

$$\min \sum_{i=1}^n a_{ik} z_i$$

Sous les contraintes :

$$\sum_{i=1}^n a_{ik} = k, \quad 0 \leq a_{ik} \leq 1, \quad i = 1, \dots, n$$

Les variables a_{ik} sont continues et représentent les coefficients de sélection des éléments du vecteur z . La contrainte $\sum_{i=1}^n a_{ik} = k$ impose que exactement k éléments soient choisis parmi les n éléments de z .

Le programme linéaire ci-dessus peut être formulé de manière duale. Pour chaque contrainte, nous introduisons des variables duales :

— r_k : variable duale associée à la contrainte suivante :

$$\sum_{i=1}^n a_{ik} = k, \quad \forall k \in \{1, 2, \dots, K\}.$$

Cette contrainte fixe la somme des variables a_{ik} égale à k pour chaque k , et r_k représente la variable duale associée à cette contrainte.

— b_{ik} : variables duales associées aux contraintes suivantes :

$$a_{ik} \leq 1, \quad \forall i \in \{1, 2, \dots, n\}, \quad \forall k \in \{1, 2, \dots, K\}.$$

Ces contraintes imposent que chaque variable a_{ik} soit inférieure ou égale à 1, et b_{ik} est la variable duale associée à cette contrainte.

Le problème primal est donc donné par :

$$\mathcal{L}(a_{ik}, r_k, b_{ik}) = \sum_{i=1}^n a_{ik} z_i + r_k \left(\sum_{i=1}^n a_{ik} - k \right) + \sum_{i=1}^n b_{ik} (a_{ik} - 1)$$

Le problème dual est formulé comme suit :

$$\max kr_k - \sum_{i=1}^n b_{ik}$$

Sous les contraintes :

$$r_k \geq 0, \quad b_{ik} \geq 0, \quad \forall i, \quad \forall k$$

Ce problème dual peut être résolu par programmation linéaire pour déterminer les valeurs des variables duales r_k et b_{ik} , qui permettront ensuite de calculer les composantes du vecteur $L_k(z)$.

4.2.1 Calcul des composantes de $L(2, 9, 6, 8, 5, 4)$

Le vecteur $z = (2, 9, 6, 8, 5, 4)$ a été utilisé pour résoudre le programme dual D_k pour différentes valeurs de k . En utilisant l'optimiseur Gurobi, nous avons calculé les composantes du vecteur $L_k(z)$ pour plusieurs valeurs de k .

— **Pour $k = 1$:**

Solution optimale : $r_1 = 9.0, \quad b_0 = 7.0, \quad b_1 = 0.0, \quad b_2 = 3.0, \quad b_3 = 1.0, \quad b_4 = 4.0, \quad b_5 = 5.0$

La valeur optimale de la fonction objectif est : $L_1(z) = -11.0$.

— **Pour $k = 2$:**

Solution optimale : $r_2 = 9.0, \quad b_0 = 7.0, \quad b_1 = 0.0, \quad b_2 = 3.0, \quad b_3 = 1.0, \quad b_4 = 4.0, \quad b_5 = 5.0$

La valeur optimale de la fonction objectif est : $L_2(z) = -2.0$.

— **Pour $k = 3$:**

Solution optimale : $r_3 = 9.0, \quad b_0 = 7.0, \quad b_1 = 0.0, \quad b_2 = 3.0, \quad b_3 = 1.0, \quad b_4 = 4.0, \quad b_5 = 5.0$

La valeur optimale de la fonction objectif est : $L_3(z) = 7.0$.

— **Pour $k = 4$:**

Solution optimale : $r_4 = 9.0, \quad b_0 = 7.0, \quad b_1 = 0.0, \quad b_2 = 3.0, \quad b_3 = 1.0, \quad b_4 = 4.0, \quad b_5 = 5.0$

La valeur optimale de la fonction objectif est : $L_4(z) = 16.0$.

— **Pour $k = 5$:**

Solution optimale : $r_5 = 9.0, \quad b_0 = 7.0, \quad b_1 = 0.0, \quad b_2 = 3.0, \quad b_3 = 1.0, \quad b_4 = 4.0, \quad b_5 = 5.0$

La valeur optimale de la fonction objectif est : $L_5(z) = 25.0$.

— **Pour $k = 6$:**

Solution optimale : $r_6 = 9.0, \quad b_0 = 7.0, \quad b_1 = 0.0, \quad b_2 = 3.0, \quad b_3 = 1.0, \quad b_4 = 4.0, \quad b_5 = 5.0$

La valeur optimale de la fonction objectif est : $L_6(z) = 34.0$.

4.3 Formulation subsidiaire de l'OWA

L'OWA (Ordered Weighted Averaging) est une fonction définie par :

$$g(x) = \sum_{k=1}^n w_k L_k(z(x))$$

où $z(x) = (z_1(x), z_2(x), \dots, z_n(x))$ est un vecteur représentant les évaluations de la solution x sous différents scénarios, et $L_k(z(x))$ représente la somme des k -plus grands éléments de $z(x)$, triés par ordre décroissant, soit :

$$L_k(z(x)) = \sum_{i=1}^k z(i)(x)$$

Les w_k sont des poids associés aux critères, et $w_1 \geq w_2 \geq \dots \geq w_n$ (les w_k sont donc des poids décroissants).

4.3.1 Définition des nouveaux poids w'_k

Nous proposons de réécrire la fonction OWA en introduisant un nouveau vecteur de poids $w' = (w'_1, w'_2, \dots, w'_n)$, défini par :

$$w'_k = w_k - w_{k+1} \quad \text{pour } k = 1, \dots, n-1$$

et

$$w'_n = w_n$$

L'objectif est de réécrire la somme $g(x) = \sum_{k=1}^n w_k L_k(z(x))$ sous la forme suivante :

$$g(x) = \sum_{k=1}^n w'_k L_k(z(x))$$

4.3.2 Décomposition de $g(x)$

L'expression de $g(x)$ peut être réécrite de la manière suivante en développant les termes :

$$g(x) = w_1 L_1(z(x)) + w_2 L_2(z(x)) + \dots + w_n L_n(z(x))$$

En développant les $L_k(z(x))$, on obtient :

$$g(x) = w_1 z_1(x) + w_1 z_2(x) + \dots + w_1 z_n(x) + (w_2 - w_1) z_2(x) + (w_2 - w_1) z_3(x) + \dots + (w_2 - w_1) z_n(x) + \dots + (w_n - w_{n-1}) z_n(x)$$

Cela revient à regrouper les termes par $L_k(z(x))$, ce qui nous permet d'obtenir l'expression suivante :

$$g(x) = \sum_{k=1}^n w'_k L_k(z(x))$$

4.3.3 Propriétés du vecteur w'

Le vecteur $w' = (w'_1, w'_2, \dots, w'_n)$ est constitué de termes $w'_k = w_k - w_{k+1}$ pour $k = 1, \dots, n-1$, et $w'_n = w_n$. Puisque les poids w_k sont décroissants, cela implique que $w_k \geq w_{k+1}$ pour tout k , et donc $w'_k = w_k - w_{k+1} \geq 0$ pour $k = 1, \dots, n-1$. De plus, $w'_n = w_n \geq 0$, car w_n est positif ou nul.

Ainsi, le vecteur w' a toutes ses composantes positives ou nulles, ce qui garantit que les poids associés à chaque $L_k(z(x))$ sont également positifs ou nuls.

4.4 MaxOwa : Formulation du programme linéaire

Le problème d'optimisation peut être formulé comme un programme linéaire, sous les contraintes suivantes :

$$\max \sum_{k=1}^n w'_k \left(r_k - \sum_{i=1}^n b_{ik} \right)$$

Sous les contraintes :

$$r_k - b_{ik} \leq z_i(x), \quad \forall i \in \{1, \dots, n\}$$

où r_k est une variable continue associée à chaque k -ième somme pondérée, et b_{ik} sont des variables continues représentant l'interaction entre les projets et les poids w'_k .

De plus, les variables b_{ik} doivent être non négatives :

$$b_{ik} \geq 0, \quad \forall i, k \in \{1, \dots, n\}$$

Nous ajoutons la contrainte d'enveloppe budgétaire :

$$\sum_{i=1}^n x_i \cdot \text{coût}_i \leq 100$$

où $x_i \in \{0, 1\}$ est une variable binaire qui représente la sélection ou non du projet i .

4.4.1 Résultat

Le problème d'optimisation a été résolu à l'aide de Gurobi, et les résultats obtenus permettent de conclure sur la sélection optimale des projets sous les contraintes budgétaires. La solution optimale du modèle linéaire a donné une valeur de la fonction objectif de 132. Les projets choisis dans la solution optimale sont ceux dont les variables de décision $x_k = 1$, c'est-à-dire les projets avec les indices 8, 9, 10, 13, 14, et 15.

4.5 MinOwa : Formulation du programme linéaire

Le critère **MinOWA des Regrets** peut être formulé comme un problème de minimisation des regrets totaux, en tenant compte des différences entre les utilités maximales et minimales des projets sélectionnés. Soit $z_1(x)$ et $z_2(x)$ les vecteurs d'utilité des projets dans les deux scénarios. Le regret pour chaque projet est défini par :

$$r_i(x) = \max(z_1(x)_i, z_2(x)_i) - z_i(x)$$

L'objectif est de minimiser la somme des regrets sur l'ensemble des projets, tout en respectant les contraintes budgétaires. La formulation linéaire du problème de **MinOWA des Regrets** est la suivante :

$$\min \sum_{k=1}^n w_k r_k(x)$$

sous les contraintes :

$$\sum_{i=1}^n c_i x_i \leq 100 \quad (\text{budget total})$$

$$r_i(x) \geq 0, \quad \forall i = 1, \dots, n$$

où x_i est une variable binaire indiquant si le projet i est sélectionné.

4.5.1 Application au problème de l'exemple 1

Dans l'exemple 1, les 10 projets sont associés à des coûts et des utilités dans les scénarios s_1 et s_2 . Le critère **MinOWA des Regrets** sera appliqué pour sélectionner les projets qui minimisent les regrets totaux, tout en respectant la contrainte de budget de 100 Keuros.

4.5.2 Résolution avec Gurobi

En résolvant ce programme linéaire avec Gurobi, nous avons obtenu la solution optimale pour le problème du **MinOWA des Regrets**. La solution optimale consiste à sélectionner les projets suivants :

$$x_1 = 1, \quad x_2 = 1, \quad x_6 = 1, \quad x_8 = 1$$

Ces projets ont été choisis afin de minimiser les regrets tout en respectant la contrainte de budget de 100 Keuros.

La valeur optimale de la fonction objectif, qui représente la somme des regrets, est de 84.

4.6 Analyse des temps de résolution des méthodes MaxOWA et MinOWA des Regrets

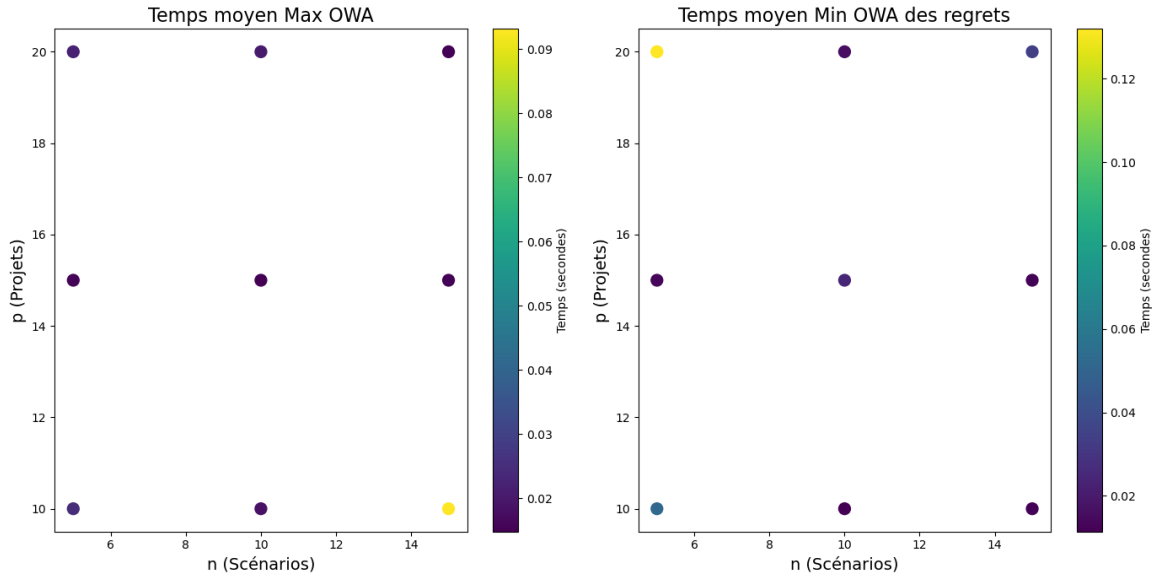


FIGURE 4 – temps de résolution des méthodes MaxOWA et MinOWA des Regrets

- **Temps moyen MaxOWA :**
 - Les temps moyens sont faibles, avec des valeurs maximales légèrement supérieures à 0,09 seconde.
 - Les temps augmentent légèrement pour des faibles valeurs de n (scénarios) et p (projets), mais restent globalement rapides.
- **Temps moyen MinOWA des Regrets :**
 - Les temps moyens sont légèrement plus élevés que pour Max OWA, atteignant une valeur maximale d'environ 0,12 seconde.
 - Les performances se dégradent pour des faibles valeurs de n et p , mais s'améliorent pour des combinaisons plus élevées.
- **Comparaison et Conclusion :**
 - Le modèle MaxOWA est plus rapide en moyenne que le MinOWA des Regrets, tout en conservant des temps faibles dans toutes les configurations de n (scénarios) et p (projets).
 - Les deux modèles présentent des performances stables pour des valeurs élevées de n et p .

5 Introduction sur les graphes : Recherche du chemin le plus rapide

Formulation du problème de chemin le plus rapide avec un programme linéaire. On cherche à minimiser le temps total pour aller du sommet initial s au sommet de destination t dans un graphe orienté et pondéré. Le temps pour parcourir chaque arc (i, j) dépend d'un scénario donné $s \in S$, et est noté t_{ij}^s .

5.1 Variables de décision

Pour chaque arc (i, j) du graphe, on définit une variable de décision :

- x_{ij} est une variable binaire qui vaut 1 si l'arc (i, j) est utilisé dans le chemin, et 0 sinon.

5.2 Fonction objectif

La fonction objectif consiste à minimiser le temps total pour parcourir le chemin de s à t sous le scénario donné s . Elle est exprimée ainsi :

$$\text{Minimiser} \quad \sum_{(i,j) \in A} t_{ij}^s \cdot x_{ij}$$

où A est l'ensemble des arcs du graphe.

5.3 Contraintes de flux

Pour garantir que le chemin relie le sommet initial s au sommet de destination t , on utilise les contraintes de flux suivantes :

1. Contrainte d'entrée et de sortie des sommets intermédiaires : Pour chaque sommet $k \in V \setminus \{s, t\}$, le nombre d'arcs entrants dans k doit être égal au nombre d'arcs sortants de k , ce qui assure la continuité du chemin.

$$\sum_{(i,k) \in A} x_{ik} - \sum_{(k,j) \in A} x_{kj} = 0 \quad \forall k \in V \setminus \{s, t\}$$

2. Contrainte pour le sommet initial s : Un seul arc doit sortir du sommet initial.

$$\sum_{(s,j) \in A} x_{sj} - \sum_{(i,s) \in A} x_{is} = 1$$

3. Contrainte pour le sommet de destination t : Un seul arc doit entrer dans le sommet de destination.

$$\sum_{(t,j) \in A} x_{tj} - \sum_{(i,t) \in A} x_{it} = 1$$

4. Binarité des variables de décision : Les variables x_{ij} doivent être binaires, indiquant si l'arc (i, j) est utilisé ou non dans le chemin.

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

5.4 Formulation complète du programme linéaire

Le programme linéaire est formulé ainsi :

$$\text{Minimiser} \quad \sum_{(i,j) \in A} t_{ij}^s \cdot x_{ij}$$

sous les contraintes :

$$\sum_{(i,k) \in A} x_{ik} - \sum_{(k,j) \in A} x_{kj} = 0 \quad \forall k \in V \setminus \{s, t\}$$

$$\sum_{(s,j) \in A} x_{sj} - \sum_{(i,s) \in A} x_{is} = 1$$

$$\sum_{(t,j) \in A} x_{tj} - \sum_{(i,t) \in A} x_{it} = 1$$

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

6 Mise en situation 2 : Application à la recherche d'un chemin robuste dans un graphe

On considère un problème de planification sous incertitude impliquant deux scénarios distincts, notés $S = \{s_1, s_2\}$. Les données du problème sont fournies sous la forme de deux instances, où chaque arc du graphe est associé à un vecteur de coûts représentant les temps de parcours correspondants dans chacun des scénarios s_1 et s_2 . Ces coûts permettent de modéliser les différences de performance des chemins possibles en fonction des scénarios envisagés.

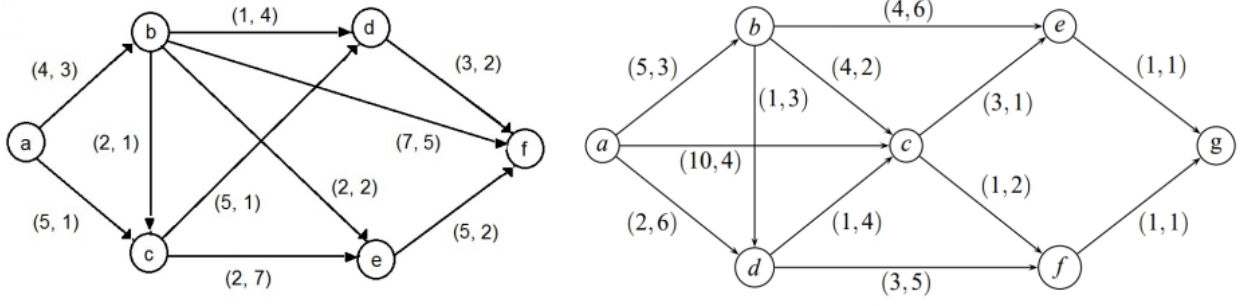


FIGURE 5 – Deux instance du problème de plus court chemin robuste à 2 scénarios

6.1 Résolution des Chemins les Plus Rapides par Programmation Linéaire

6.1.1 Description du problème

Nous considérons deux graphes, chacun représentant une instance distincte du problème.

— **Graphe de gauche** : Trouver le chemin le plus rapide de a à f .

— **Graphe de droite** : Trouver le chemin le plus rapide de a à g .

Chaque arc est associé à un vecteur de coûts représentant les temps de parcours dans les deux scénarios s_1 et s_2 .

6.1.2 Variables de décision

Pour chaque arc (i, j) , nous définissons une variable binaire x_{ij} :

$$x_{ij} = \begin{cases} 1 & \text{si l'arc } (i, j) \text{ est inclus dans le chemin,} \\ 0 & \text{sinon.} \end{cases}$$

6.1.3 Fonction objectif

Pour chaque scénario $s \in \{s_1, s_2\}$, la fonction objectif consiste à minimiser le temps total de parcours entre le sommet initial et le sommet final :

$$\text{Minimiser } \sum_{(i,j) \in A} t_{ij}^s \cdot x_{ij}$$

où A est l'ensemble des arcs du graphe, et t_{ij}^s est le temps de parcours de l'arc (i, j) dans le scénario s .

6.1.4 Contraintes

Les contraintes garantissent la validité du chemin dans le graphe :

— **Contraintes de flux pour les sommets intermédiaires** : Pour tout sommet $k \in V \setminus \{a, t\}$ (où t est le sommet final), le flux entrant doit être égal au flux sortant :

$$\sum_{(i,k) \in A} x_{ik} - \sum_{(k,j) \in A} x_{kj} = 0 \quad \forall k \in V \setminus \{a, t\}.$$

— **Contraintes pour le sommet initial a** : Un seul arc doit sortir de a , garantissant qu'il est le point de départ :

$$\sum_{(a,j) \in A} x_{aj} = 1.$$

— **Contraintes pour le sommet final t** : Un seul arc doit entrer dans t , garantissant qu'il est le point d'arrivée :

$$\sum_{(i,t) \in A} x_{it} = 1.$$

— **Binarité des variables** : Les variables x_{ij} doivent être binaires :

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A.$$

6.1.5 Représentation des résultats

Scénario	Chemin optimal	Arcs parcourus	Temps total
s_1	$A \rightarrow B \rightarrow D \rightarrow F$	$(A, B), (B, D), (D, F)$	$t_{AB}^{s_1} + t_{BD}^{s_1} + t_{DF}^{s_1} = 8$
s_2	$A \rightarrow C \rightarrow D \rightarrow F$	$(A, C), (C, D), (D, F)$	$t_{AC}^{s_2} + t_{CD}^{s_2} + t_{DF}^{s_2} = 4$

TABLE 1 – Résumé des chemins optimaux et leurs temps totaux dans chaque scénario du graphe de gauche

Scénario	Chemin optimal	Arcs parcourus	Temps total
s_1	$A \rightarrow D \rightarrow C \rightarrow F \rightarrow G$	$(A, D), (D, C), (C, F), (F, G)$	$t_{AD}^{s_1} + t_{DC}^{s_1} + t_{CF}^{s_1} + t_{FG}^{s_1} = 5$
s_2	$A \rightarrow C \rightarrow E \rightarrow G$	$(A, C), (C, E), (E, G)$	$t_{AC}^{s_2} + t_{CE}^{s_2} + t_{EG}^{s_2} = 6$

TABLE 2 – Résumé des chemins optimaux et leurs temps totaux dans chaque scénario du graphe de droite

les chemins optimaux ont été déterminés pour chaque scénario à l'aide de la programmation linéaire, permettant de trouver les trajets les plus rapides tout en respectant les contraintes du problème. Nous allons maintenant explorer la recherche d'un chemin robuste, capable de performer efficacement malgré les incertitudes et variations entre les scénarios.

6.2 Recherche d'un Chemin Robuste

6.2.1 Formulation du Problème

Le problème de chemin robuste consiste à trouver un chemin P qui optimise la fonction $v_g(P) = g(-t_1(P), \dots, -t_n(P))$, où :

- $t_i(P)$ représente le temps total de parcours du chemin P dans le scénario s_i ,
- g est une fonction de robustesse parmi les quatre proposées : **Maximin**, **Minimax regret**, **MaxOWA**, et **MinOWA** des regrets.

L'objectif est de garantir une performance acceptable du chemin P , quelle que soit l'incertitude liée aux scénarios.

6.2.2 Formulation par Programme Linéaire

Pour chacune des fonctions g , nous proposons les formulations linéaires suivantes :

a. MaxMin Maximiser la pire performance dans les différents scénarios :

$$\max \min \{-t_1(P), -t_2(P), \dots, -t_n(P)\}.$$

Programme linéaire :

- Introduire une variable auxiliaire z représentant la pire performance :

$$\max z$$

- Ajouter des contraintes pour chaque scénario :

$$z \leq -t_i(P), \quad \forall i \in \{1, \dots, n\}.$$

b. MinMax Regret Minimiser le regret maximum. Le regret pour un chemin P et un scénario s_i est défini comme la différence entre $t_i(P)$ et le temps optimal t_i^* pour ce scénario :

$$\min \max \{t_i(P) - t_i^*\}.$$

Programme linéaire :

— Introduire une variable auxiliaire z représentant le regret maximum :

$$\min z$$

— Ajouter des contraintes pour chaque scénario :

$$t_i(P) - t_i^* \leq z, \quad \forall i \in \{1, \dots, n\}.$$

c. MaxOWA (Ordered Weighted Average) Maximiser une moyenne pondérée décroissante des $-t_i(P)$. Pour un vecteur de pondérations $w = (w_1, w_2, \dots, w_n)$, les scénarios sont triés par ordre décroissant de $-t_i(P)$:

$$\max \sum_{i=1}^n w_i \cdot (-t_{(i)}(P)),$$

où $t_{(i)}(P)$ est le i^e scénario après tri.

Programme linéaire :

— Introduire des variables y_i représentant les valeurs triées $-t_i(P)$, avec des contraintes pour ordonner les $t_i(P)$.

— Maximiser la moyenne pondérée :

$$\max \sum_{i=1}^n w_i \cdot y_i.$$

d. MinOWA des regrets Minimiser une moyenne pondérée décroissante des regrets :

$$\min \sum_{i=1}^n w_i \cdot r_{(i)},$$

où $r_{(i)}$ représente les regrets triés $t_i(P) - t_i^*$.

Programme linéaire :

— Introduire des variables pour les regrets r_i et leur tri $r_{(i)}$.

— Ajouter des contraintes pour lier r_i aux temps de parcours et au temps optimal :

$$r_i = t_i(P) - t_i^*.$$

— Minimiser la moyenne pondérée :

$$\min \sum_{i=1}^n w_i \cdot r_{(i)}.$$

6.2.3 Résultats et Commentaires

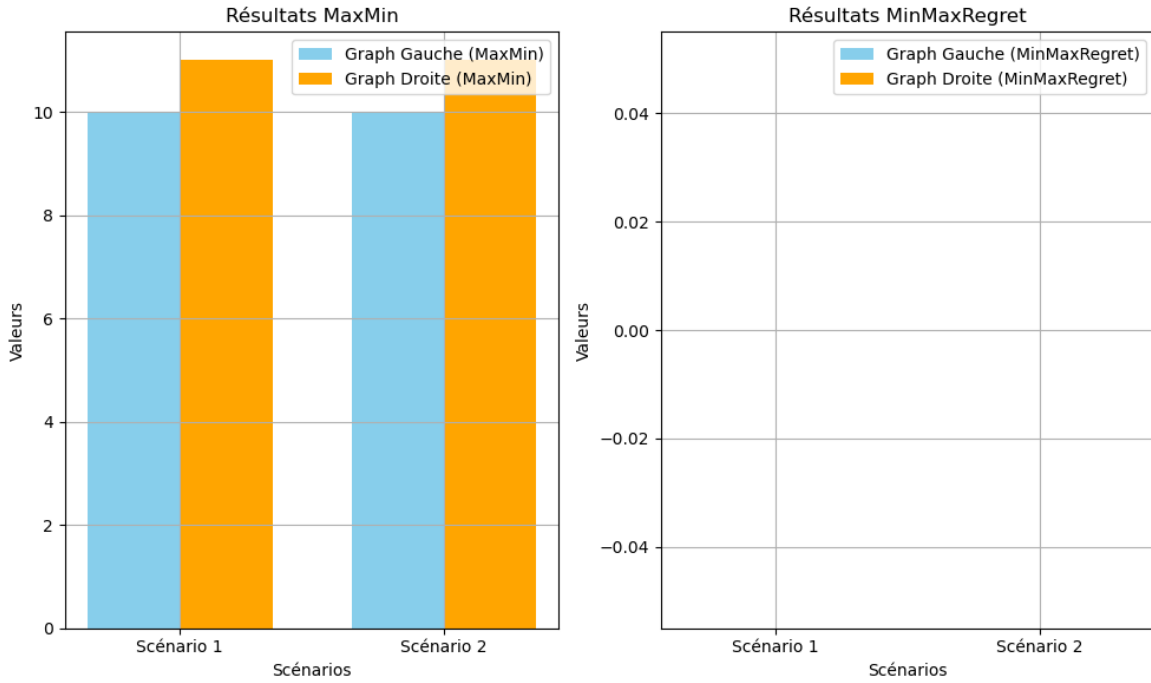


FIGURE 6 – Comparaison des Valeurs Optimales pour les Fonctions de Robustesse (**MaxMin** et **MinMax Regret**)

Critère MaxMin : Les résultats pour le critère MaxMin montrent des valeurs constantes et élevées (10) pour le scénario dans le graphes de gauche et (11) dans celui de droite. Cela indique que le chemin sélectionné maximise le pire cas de performance (le minimum parmi les deux scénarios). Ce résultat souligne la robustesse des solutions proposées, avec une grande similarité dans les performances pour les deux graphes.

Critère MinMax Regret : Les valeurs obtenues pour le critère MinMax Regret sont nulles pour les deux scénarios dans les deux graphes. Une valeur nulle pour le regret maximal signifie que le chemin sélectionné est optimal pour tous les scénarios. Autrement dit, il n'y a aucun regret à utiliser le chemin choisi, car il correspond exactement à la solution idéale pour chaque scénario. Cela démontre une robustesse parfaite des solutions selon ce critère.

Comparaison des deux critères :

- Le **MaxMin** garantit une solution robuste en maximisant les pires performances possibles. Cette approche est ici stable entre les deux graphes.
- Le **MinMax Regret**, dans cette configuration, offre une solution idéale avec des regrets nuls, indiquant une absence totale de compromis entre les scénarios. Ce résultat est particulièrement intéressant dans ce contexte.

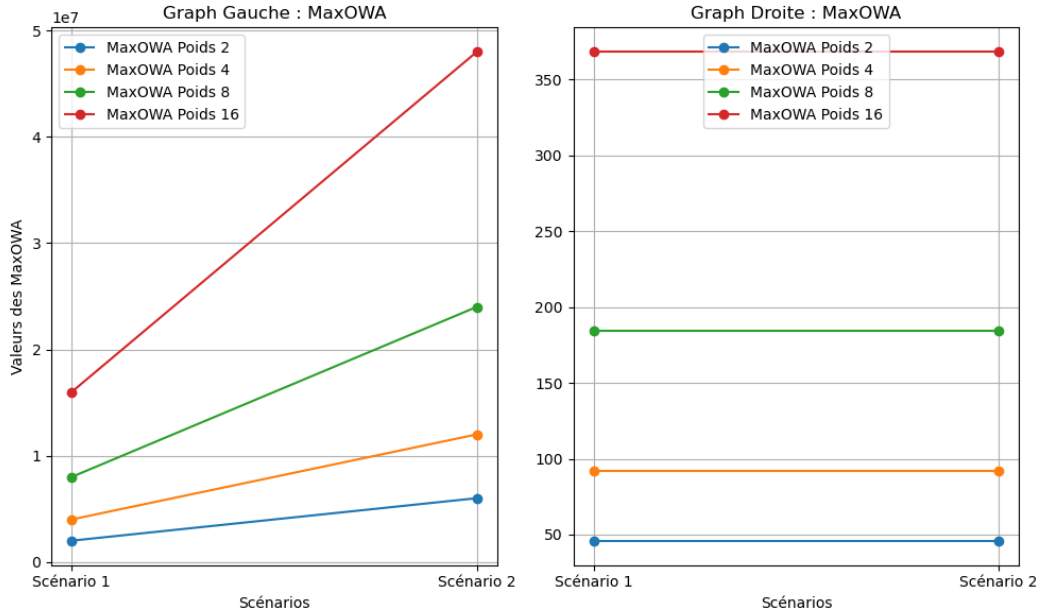


FIGURE 7 – Comparaison des Valeurs Optimales pour la Fonction de Robustesse **MaxOWA**

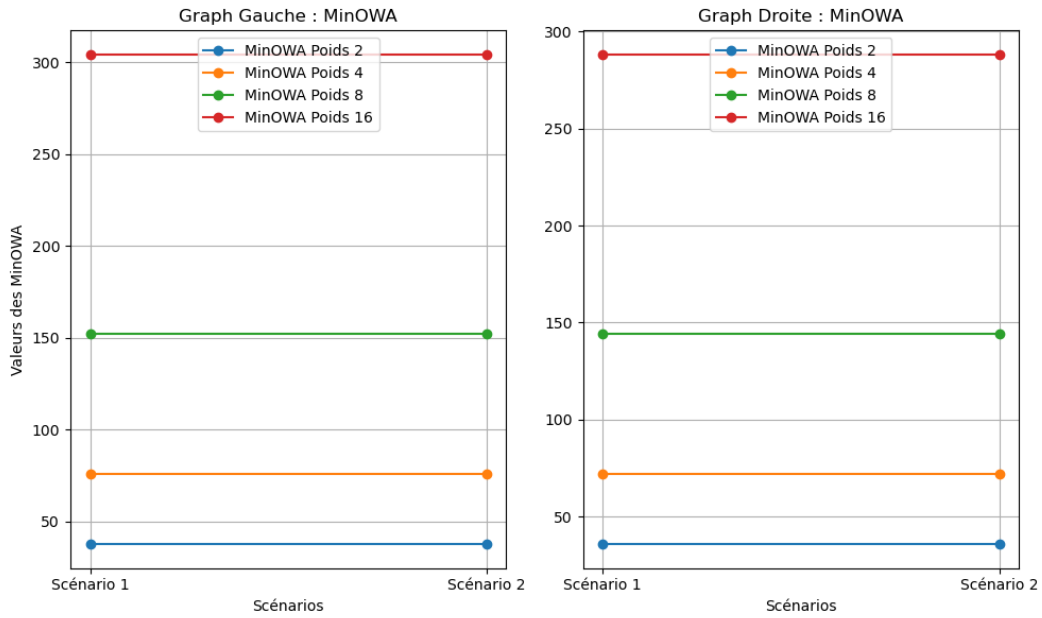


FIGURE 8 – Comparaison des Valeurs Optimales pour la Fonction de Robustesse **MinOWA**

Graphique MinOWA :

Le premier graphique compare les valeurs obtenues par la fonction **MinOWA** en fonction des poids $k = 2, 4, 8, 16$ pour les graphes de gauche et de droite. Les observations principales sont les suivantes :

- Les valeurs restent constantes pour chaque scénario, quel que soit le poids utilisé.
- Cette stabilité indique que les poids k n'ont pas un impact significatif sur la fonction MinOWA dans ces configurations.
- Les résultats sont similaires pour les graphes gauche et droit, montrant une robustesse de la fonction MinOWA face aux variations de poids.

Graphique MaxOWA :

Le second graphique illustre les valeurs obtenues par la fonction **MaxOWA**, et met en évidence des comportements très différents :

- Les valeurs de MaxOWA augmentent considérablement avec l'augmentation des poids k .
- Pour le graphe gauche, les valeurs explosent avec des poids élevés ($k = 16$), tandis que pour le graphe droit, les augmentations sont plus modérées mais restent significatives.
- Ce comportement reflète la sensibilité de MaxOWA aux poids, en amplifiant les écarts et en favorisant les grandes valeurs dans le calcul.

Comparaison générale :

- La fonction **MinOWA** offre des résultats stables et insensibles aux variations de poids, ce qui peut être utile dans des contextes où la minimisation des risques est prioritaire.
- La fonction **MaxOWA**, en revanche, montre une forte variabilité selon les poids, la rendant plus adaptée aux scénarios où il est important de maximiser les opportunités ou les gains potentiels.

6.3 Analyse de l'évolution du temps de résolution pour la détermination d'un chemin optimal

Dans cette section, nous analysons l'évolution du temps moyen nécessaire pour résoudre le problème de recherche d'un chemin optimal selon les quatre critères étudiés : **MaxMin**, **MinMax Regret**, **MaxOWA**, et **MinOWA des regrets**, en fonction du nombre de scénarios (n) et du nombre de sommets (p) d'un graphe.

6.3.1 Plan d'étude

- **Paramètres étudiés :**
 - Nombre de scénarios (n) : $n = 2, 5, 10$.
 - Nombre de sommets (p) : $p = 10, 15, 20$.
- **Génération des instances :**
 - Pour chaque couple (n, p) , 10 instances sont générées aléatoirement.
 - Les coûts des arcs sont tirés aléatoirement dans l'intervalle $[1, 100]$.
 - La densité des arcs dans les graphes est comprise entre 30% et 50%.
- **Résolution et mesure des temps :**
 - Les instances sont résolues en utilisant les quatre critères (**MaxMin**, **MinMax Regret**, **MaxOWA**, **MinOWA des regrets**).
 - Les temps de résolution sont mesurés et moyennés pour chaque couple (n, p) .

6.3.2 Résultats et Commentaires

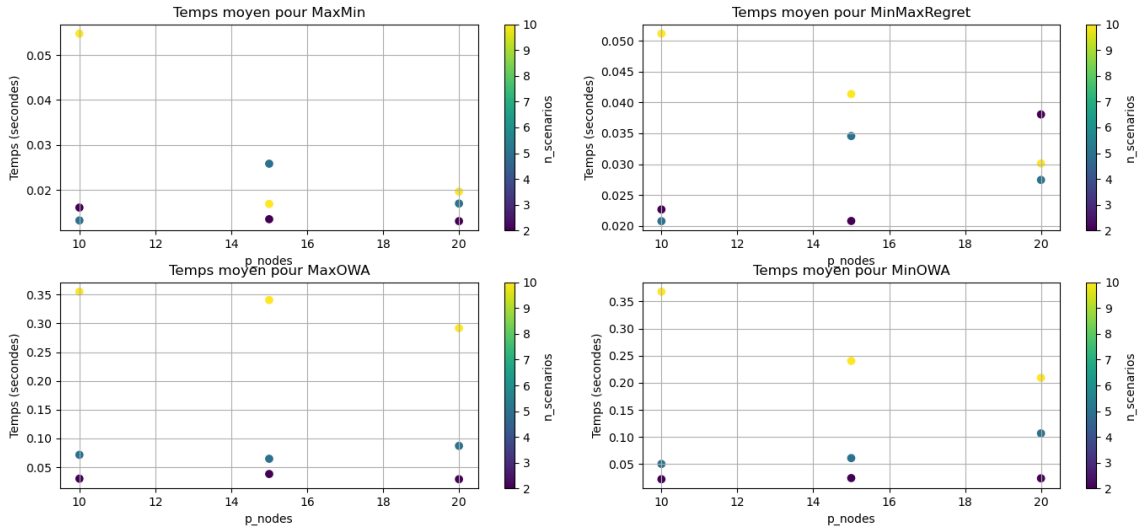


FIGURE 9 – Comparaison des Valeurs Optimales pour chaque Fonction de Robustesse en fonction du nombre de scénarios et de noeuds

Ces graphes présentent le **temps moyen d'exécution** pour différents algorithmes (MaxMin, MinMaxRegret, MaxOWA, MinOWA) en fonction de deux paramètres :

- p_{nodes} : le nombre de nœuds.
- $n_{scenarios}$: le nombre de scénarios (représenté par les couleurs).
- Le **temps d'exécution augmente avec** p_{nodes} pour tous les algorithmes.
- Le **nombre de scénarios** $n_{scenarios}$ influence également les résultats : plus $n_{scenarios}$ est élevé, plus le temps d'exécution est long.

Comparaison des algorithmes

- Les algorithmes **MaxMin** et **MinMax Regret** (graphiques en haut) sont globalement plus rapides et restent performants même avec des valeurs élevées de p_{nodes} et $n_{scenarios}$.
- Les algorithmes **MaxOWA** et **MinOWA** (graphiques en bas) ont des **temps moyens nettement supérieurs**, particulièrement sensibles à l'augmentation de p_{nodes} et $n_{scenarios}$.
- **MaxMin** et **MinMax Regret** sont plus adaptés pour des situations nécessitant une faible durée de calcul.
- **MaxOWA** et **MinOWA** sont mieux adaptés pour des contextes où des calculs plus lourds sont acceptables.
- L'effet de $n_{scenarios}$ est particulièrement marqué pour **MaxOWA** et **MinOWA**, indiquant une complexité croissante avec ce paramètre.

7 Conclusion

Cette étude explore divers critères d'optimisation robuste dans des contextes incertains, appliqués à des problèmes comme le sac-à-dos et le plus court chemin. Elle propose des modèles linéaires pour des critères tels que **MaxMin**, **MinMax Regret**, **MaxOWA** et **MinOWA des Regrets** via des algorithmes comme Gurobi. Les résultats montrent comment la robustesse optimise les solutions en intégrant les pires scénarios, avec des tests sur des instances variées confirmant l'efficacité et les limites de chaque approche en termes de complexité et de temps de calcul. Cette étude pourrait être étendue en appliquant les modèles à des problèmes réels, comme la gestion des réseaux de transport. Cela permettrait de tester l'efficacité des approches dans des situations concrètes et d'évaluer leur performance pour optimiser la gestion des flux et la prise de décision dans des contextes incertains.