

# MOGPL Projet 2024-2025

Master Informatique – Semestre 1

Sorbonne Université

**Optimisation robuste dans l'incertain rotal**

**Auteurs :**

Karim ELLOUZE

Philipp HANUSSEK

# Chapitre 1

## Linéarisation des critères maxmin et minmax regret

### 1.1 Programme linéaire *maxmin*

$$\begin{aligned} \max \quad & t \\ \text{sous contraintes} \quad & t \leq \sum_{i=1}^n v_{i,1} x_i \\ & t \leq \sum_{i=1}^n v_{i,2} x_i \\ & \sum_{i=1}^n w_i x_i \leq W \\ & x_i \in \{0, 1\}, \quad \forall i = 1, \dots, n \end{aligned}$$

Solution optimale trouvé avec Gurobi :

$$\begin{aligned} x^* &= [2, 3, 4, 7, 8, 9] \\ z(x^*) &= [66, 66] \end{aligned}$$

### 1.2 Programme linéaire *minmax regret*

$$\begin{aligned} \min \quad & t \\ \text{sous contraintes} \quad & t \geq z_1^* - \sum_{i=1}^n v_{i,1} x_i \\ & t \geq z_2^* - \sum_{i=1}^n v_{i,2} x_i \\ & \sum_{i=1}^n w_i x_i \leq W \\ & x_i \in \{0, 1\}, \quad \forall i = 1, \dots, n \end{aligned}$$

$z_i^*$  étant la meilleure valeur pour le scénario i

Solution optimale trouvé avec Gurobi :

$$\begin{aligned} x'^* &= [2, 3, 6, 7, 8, 9] \\ z(x_1^*, x_2^*) &= [112, 118] \\ z(x'^*) &= [62, 70] \end{aligned}$$

### 1.3 Représentation dans le plan

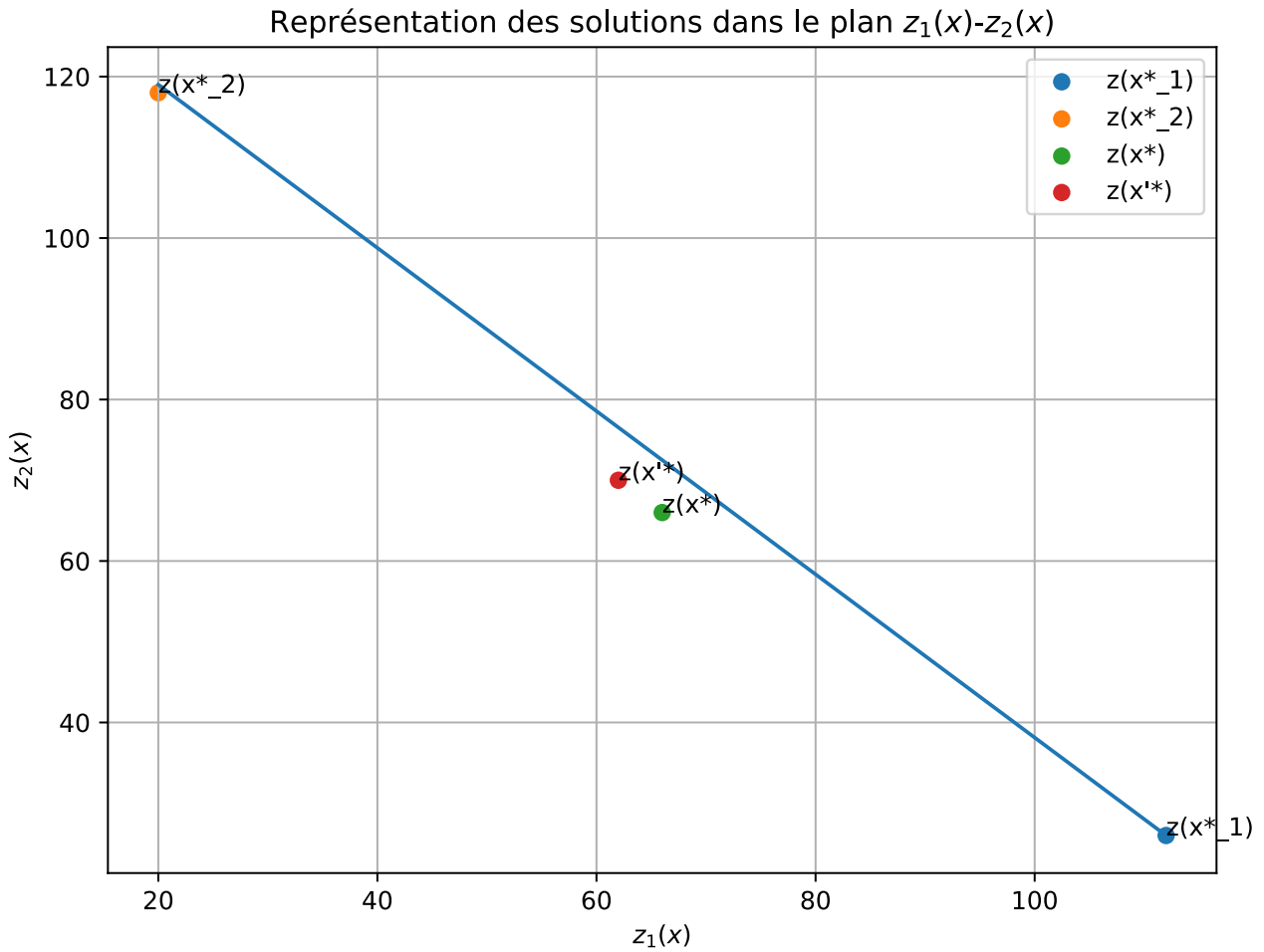


FIGURE 1.1 – Représentation des solutions  $z(x^*_1)$ ,  $z(x^*_2)$ ,  $z(x')$ ,  $z(x'^*)$  dans le plan  $z_1(x)$  et  $z_2(x)$

Conclusion : Les solutions trouvés dans la question 1 et 2 ne sont pas sur la ligne entre les variables  $z(x^*_1)$ ,  $z(x^*_2)$ , donc on ne peut pas trouver  $z(x')$  et  $z(x'^*)$  en ponderant les  $z(x^*_1)$ ,  $z(x^*_2)$  (voir figure 1.1).

### 1.4 Analyse de performance

Comme on observe dans la figure 1.2 regret prend plus de temps que maxMin, parce qu'il calcule la valeur optimale  $z^*$  pour chaque scénario d'abord. Le temps de calcul augmente pour les deux quand il y a plus des constraints (traduit par plusieurs scénarios). Les résultats trouvés qui montrent un temps supérieur de résolution pour  $p = 10$  à  $p = 20$  ne sont pas consistante avec ce qui est attendu et avec ce qui est observé dans l'exercice 2 et 3.

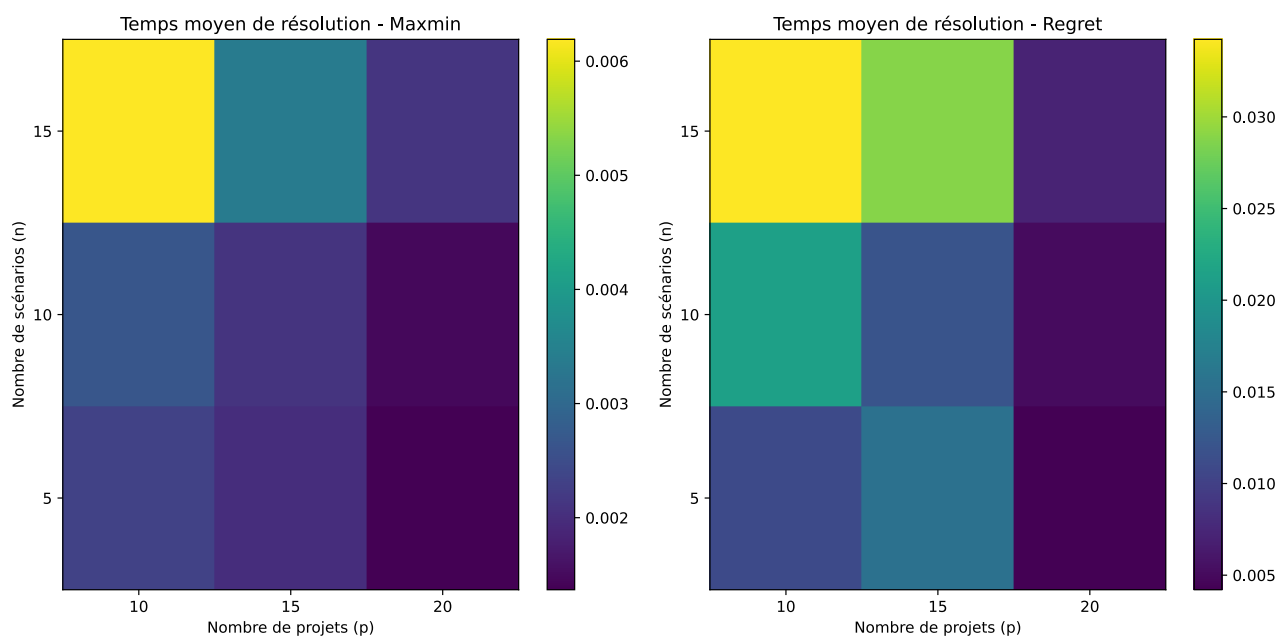


FIGURE 1.2 – Comparaison des temps d'exécution pour les modèles maxmin et minmax regret.

## Chapitre 2

# Linéarisation du critère *maxOWA*

### 2.1 $L_k(z)$ comme valeur optimale

**Explication :** Étant donné que le vecteur  $(z_{(1)}, \dots, z_{(n)})$  est ordonné en ordre croissant, la somme  $\sum_{i=1}^k z_{(i)}$  correspond à la somme des  $k$  plus petites valeurs de  $z_i$  et donc à la valeur optimale du programme linéaire.

### 2.2 Utiliser le dual pour trouver les composantes du $L(z)$

Primal :

$$\begin{aligned} \min \quad & \sum_{i=1}^n a_{ik} z_i \\ \text{s.c.} \quad & \sum_{i=1}^n a_{ik} = k \\ & 0 \leq a_{ik} \leq 1, \quad i = 1, \dots, n \end{aligned}$$

Dual :

$$\begin{aligned} \max \quad & k \cdot r_k - \sum_{i=1}^n b_{ik} \\ \text{s.c.} \quad & r_k - b_{ik} \leq z_i, \quad \forall i = 1, \dots, n \\ & b_{ik} \geq 0, \quad \forall i = 1, \dots, n \end{aligned}$$

Trouver  $L(z)$  avec  $z = [2, 9, 6, 8, 5, 4]$  (voir le code pour calcul par programmation linéaire) :

$$\begin{aligned} L_1(z) &= 2 \\ L_2(z) &= 6 \\ L_3(z) &= 11 \\ L_4(z) &= 17 \\ L_5(z) &= 25 \\ L_6(z) &= 34 \end{aligned}$$

### 2.3 Réécrire l'OWA

Montrer que

$$\begin{aligned} g(x) &= \sum_{i=1}^n w_i z_{(i)}(x) \\ &= \sum_{k=1}^n w'_k L_k(z(x)) \end{aligned}$$

**Explication :**

par définition :  $z_{(k)}(x) = L_k(z(x)) - L_{k-1}(z(x))$

$$\begin{aligned}
g(x) &= \sum_{i=1}^n w_i z_{(i)}(x) \\
&= w_1 z_{(1)} + w_2 z_{(2)} + \dots + w_n z_{(n)} \\
&= w_1 L_1(z(x)) + w_2 (L_2(z(x)) - L_1(z(x))) + \dots + w_n (L_n(z(x)) - L_{n-1}(z(x))) \\
&= (w_1 - w_2) L_1(z(x)) + (w_2 - w_3) (L_2(z(x))) + \dots + (w_{n-1} - w_n) L_n \\
&= \sum_{k=1}^n w'_k L_k(z(x))
\end{aligned}$$

## 2.4 Formulation *maxOWA* pour le problème SAD

$$\begin{array}{ll}
\max & \sum_{k=1}^n w'_k \left( k \cdot r_k - \sum_{i=1}^n b_{ik} \right) \\
\text{s.c.} & r_k - b_{ik} \leq z_i(x) \\
& b_{ik} \geq 0 \\
& \sum_{i=1}^n w_i x_i \leq W, \\
& x_i \in \{0, 1\}, \quad i = 1, \dots, n.
\end{array}$$

Solution optimale trouvé avec Gurobi :

$$\begin{aligned}
x^* &= [2, 3, 4, 7, 8, 9] \\
z(x^*) &= [66, 66]
\end{aligned}$$

## 2.5 Formulation linéaire pour le critère de *minOWA*

Primal :

$L_k(z) = \sum_{i=1}^k r(x, s_{(i)})$ , regret en ordre décroissant, donc :

$$\begin{array}{ll}
\max & \sum_{i=1}^n a_{ik} r(x, s_{(i)}) \\
\text{s.c.} & \sum_{i=1}^n a_{ik} = k \\
& 0 \leq a_{ik} \leq 1, \quad i = 1, \dots, n
\end{array}$$

$$r(x, s_{(i)}) = z_i^* - z_i(x)$$

Dual :

$$\begin{array}{ll}
\min & k \cdot r_k + \sum_{i=1}^n b_{ik} \\
\text{s.c.} & r_k + b_{ik} \geq r(x, s_{(i)}), \quad \forall i = 1, \dots, n \\
& b_{ik} \geq 0, \quad \forall i = 1, \dots, n
\end{array}$$

Formulation entière :

$$\begin{aligned}
& \min \sum_{k=1}^n w'_k \left( k \cdot r_k + \sum_{i=1}^n b_{ik} \right) \\
& \text{s.c.} \quad r_k + b_{ik} \geq r(x, s_{(i)}), \quad i = 1, \dots, n \\
& \quad \quad b_{ik} \geq 0, \quad i = 1, \dots, n \\
& \quad \quad \sum_{i=1}^n w_i x_i \leq W \\
& \quad \quad x_i \in \{0, 1\} \quad \forall i = 1, \dots, n
\end{aligned}$$

Solution optimale trouvé avec Gurobi :

$$\begin{aligned}
x'^* &= [2, 4, 5, 7, 8, 9] \\
z(x_1^*, x_2^*) &= [112, 118] \\
z(x'^*) &= [62, 70]
\end{aligned}$$

## 2.6 Analyse de performance

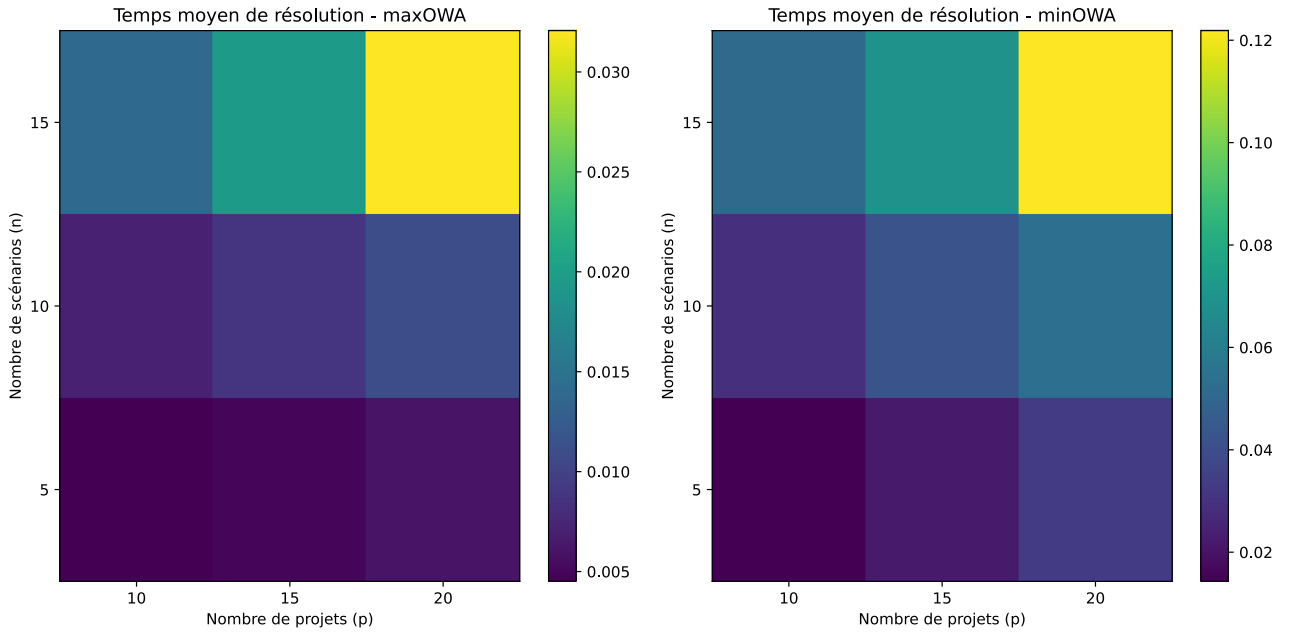


FIGURE 2.1 – Comparaison des temps d'exécution pour les modèles maxOWA et minOWA des regrets.

Comme dans l'exercice 1, on observe que minOWA regrets prend plus de temps que maxOWA. Le temps d'exécution augmente avec le nombre de variables (projets) et le nombre de contraintes (scénarios). En plus, on observe que le temps de résolution augmente plus en ajoutant une contrainte qu'en ajoutant une variable.

## Chapitre 3

# Application à la recherche d'un chemin robuste dans un graphe

### 3.1 Modélisation du problème de plus court chemin comme problème du flot de coût minimum

$G = (V, E)$ , source :  $s \in V$ , puit :  $t \in V$

$a(u, v)$  : coût pour passer de  $u$  à  $v$ ,  $f(u, v) = 1$  si  $(u, v) \in$  plus court chemin

Programme linéaire :

$$\begin{aligned} \min \quad & \sum_{(u,v) \in E} a(u,v)f(u,v) \\ \text{s.c.} \quad & \sum_{(u,v) \in E} f(u,v) = \sum_{(v,w) \in E} f(v,w) \quad \forall v \in V \setminus \{s, t\} \quad , \text{ contrainte de conservation du flux} \\ & \sum_{s,v \in E} f(s,v) \geq 1 \quad , \text{ contrainte de flux sortant de la source} \\ & f(u,v) \in (0, 1) \quad \forall (u,v) \in E \end{aligned}$$

### 3.2 Application aux exemples

Solution optimale trouvée avec Gurobi pour l'instance 1 :

$$\begin{aligned} x_1^* &= [a, b, d, f] \\ x_2^* &= [a, c, d, f] \\ z(x^*) &= [8, 4] \end{aligned}$$

Pour l'instance 2 :

$$\begin{aligned} x_1^* &= [a, d, c, f, g] \\ x_2^* &= [a, c, e, g] \\ z(x^*) &= [5, 6] \end{aligned}$$

### 3.3 Programmes linéaires pour trouver un chemin robuste

$x$  est une solution valide si :

$$\begin{aligned} \sum_{(u,v) \in E} f(u,v) &= \sum_{(v,w) \in E} f(v,w) \quad \forall v \in V \setminus \{s, t\} \quad , \text{ contrainte de conservation du flux} \\ \sum_{s,v \in E} f(s,v) &\geq 1 \quad , \text{ contrainte de flux sortant de la source} \\ f(u,v) &\in \{0, 1\} \quad \forall (u,v) \in E \end{aligned}$$



### ***minmax***

Comme on veut trouver le plus court chemin, on calcule le *min* des solutions maximales pour un chemin  $X$ .  
Programme linéaire pour  $n$  scénarios :

$$\begin{aligned}
& \min \quad t \\
& \text{s.c.} \quad t \geq \sum_{(u,v) \in E} a_i(u,v) f(u,v) \quad i \in \{1, \dots, n\} \\
& \quad x \in X \\
& \quad f(u,v) \in \{0, 1\} \quad \forall (u,v) \in E
\end{aligned}$$

### ***minmax* des regrets**

Programme linéaire pour  $n$  scénarios :

$$\begin{aligned}
& \min \quad t \\
& \text{s.c.} \quad t \geq \sum_{(u,v) \in E} (a_i(u,v) f(u,v)) - z_i^* \quad i \in \{1, \dots, n\} \\
& \quad x \in X \\
& \quad f(u,v) \in \{0, 1\} \quad \forall (u,v) \in E
\end{aligned}$$

### ***minOWA***

$$\begin{aligned}
& \min \sum_{k=1}^n w'_k \left( k \cdot r_k + \sum_{i=1}^n b_{ik} \right) \\
& \text{s.c.} \quad r_k + b_{ik} \geq z_i(x) \quad i = 1, \dots, n \\
& \quad x \in X \\
& \quad b_{ik} \geq 0, \quad i = 1, \dots, n
\end{aligned}$$

### ***minOWA* des regrets**

$$\begin{aligned}
& \min \sum_{k=1}^n w'_k \left( k \cdot r_k + \sum_{i=1}^n b_{ik} \right) \\
& \text{s.c.} \quad r_k + b_{ik} \geq r(x, s_{(i)}), \quad i = 1, \dots, n \\
& \quad b_{ik} \geq 0, \quad i = 1, \dots, n \\
& \quad x \in X
\end{aligned}$$

## **Résultats**

### **minMax et minMax des regrets**

Instance	Méthode	Chemin	Valeur de la fonction objectif	Vecteur <b>_z</b>
Instance 0	minmax	a,b,d,f	9	[8, 9]
	minMax des regrets	a,b,e,f	3	[11, 7]
Instance 1	minmax	a,b,e,g	10	[10, 10]
	minMax des regrets	a,b,e,g	5	[10, 10]

TABLE 3.1 – Comparaison des résultats des méthodes minmax et minMax des regrets.

Méthode	Instance	$k$	Chemin	Vecteur_z	Valeur de la fonction objectif
minOWA	0	2	a,b,d,f	[8, 9]	26
		4	a,b,d,f	[8, 9]	44
		8	a,b,d,f	[8, 9]	80
		16	a,b,d,f	[8, 9]	152
	1	2	a,b,e,g	[10, 10]	30
		4	a,b,e,g	[10, 10]	50
		8	a,b,e,g	[10, 10]	90
		16	a,b,e,g	[10, 10]	170
minOWA des regrets	0	2	a,b,e,f	[11, 7]	9
		4	a,b,e,f	[11, 7]	15
		8	a,b,e,f	[11, 7]	27
		16	a,b,e,f	[11, 7]	51
	1	2	a,d,f,g	[6, 12]	13
		4	a,b,e,g	[10, 10]	24
		8	a,b,e,g	[10, 10]	44
		16	a,b,e,g	[10, 10]	84

TABLE 3.2 – Comparaison des méthodes minOWA et minOWA des regrets.

Instance	$z^*$
0	[8, 4]
1	[5, 6]

TABLE 3.3 – Valeurs de  $z^*$  pour chaque instance

### minOWA et minOWA des regrets

On observe dans le tableau 3.2 qu'il y a un seul changement de chemin dans minOWA des regrets pour l'instance 1. Ce changement est due au fait que lorsque  $k = 2$  les regrets sont de  $[1, 6]$ , donc pour une faible valeur  $k$  le chemin  $a, d, f, g$  est optimale, mais si on augmente  $k$ , le vecteur des regrets  $z = [5, 4]$  correspondant au chemin  $a, b, e, g$  devient l'optimal. Donc pour des valeurs plus élevés de  $k$  on cherche un vecteur (de regrets) avec des valeurs qui sont assez proches.

## 3.4 Analyse de performance

On observe dans la figure 3.1 que les méthodes calculant le regret (minMaxRegret, minOWAregret) ont un temps moyen d'exécution plus important à cause du calcul de  $z^*$ . On observe que minOWA prend plus de temps que minMax et que minOWARegret prend plus de temps que minMaxRegret parce qu'on utilise plus de variables. Dans l'OWA il y a  $n$  variables  $r_k$  et  $n^2$  variables  $b_{ik}$  et pour maxMin il y a  $p$  variables. Pour les valeurs  $n$  et  $p$  prises,  $n^2 + n$  est toujours supérieur à  $p$ .

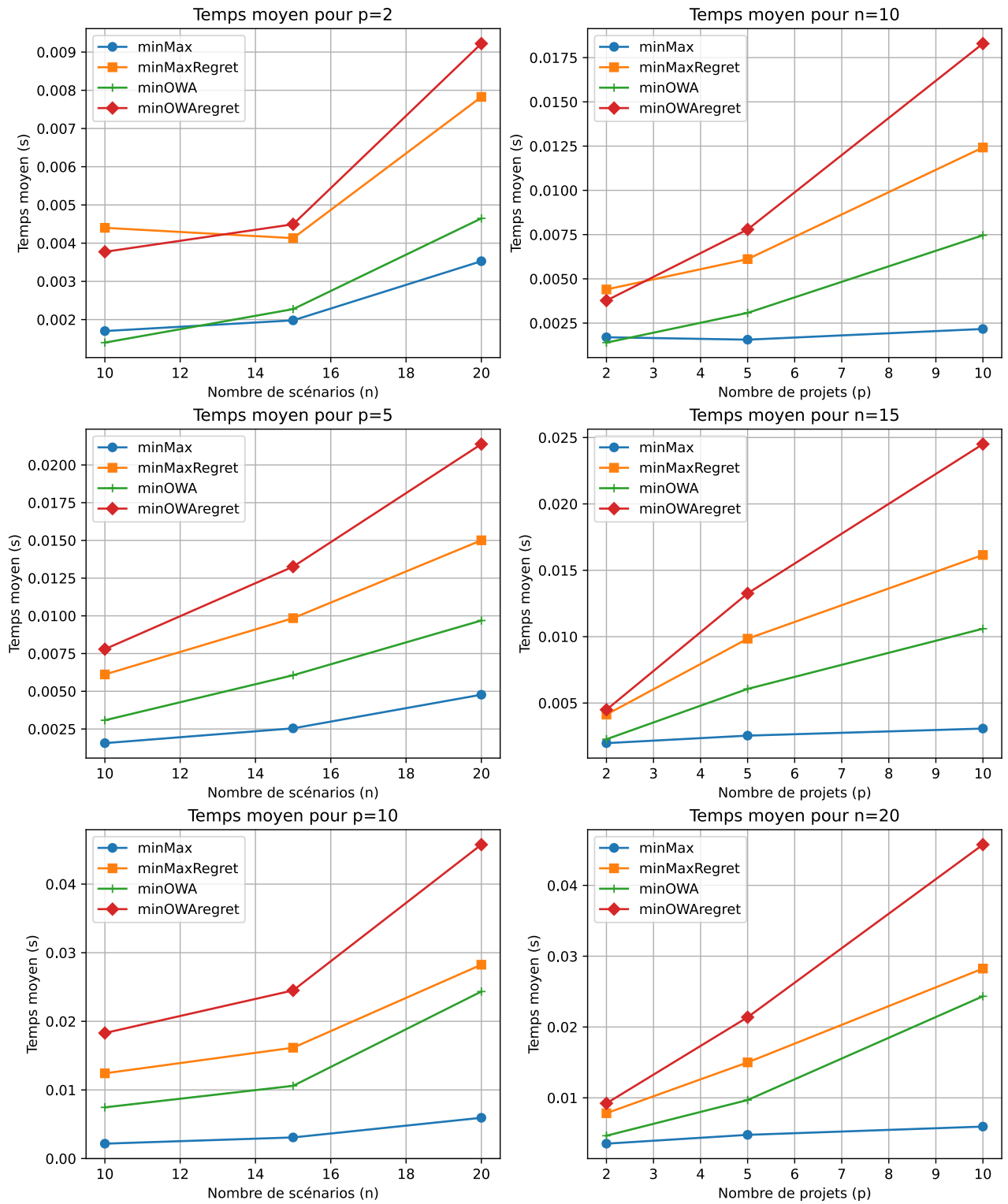


FIGURE 3.1 – Comparaison des temps d'exécution pour les modèles minMax, minMaxRegret, minOWA et minOWA des regrets.