**MAMLOUK** Haya [21107689] Sorbonne Université

**OZGENC** Doruk [21113927] 2024-2025

**MOGPL – Rapport de projet :**

**Optimisation robuste dans l’incertain total**

SUMMARY

INTROdUCTIOn explain the example sac s dos that we will use

1. **Linéarisation des critères maxmin et minmax regret**

Posons pour la suite les données suivantes :

* + - ***P***  : l’ensemble des projets
    - ***B***  : le budget
    - : utilité du projet *j* dans le scénario *i* avec ,
    - : coût du projet *j* avec

1. **Formulation du problème de maxmin**

Le critère **maxmin** cherche à maximiser la valeur minimale obtenue dans tous les scénarios possibles. En d'autres termes, on veut trouver une solution qui garantit le meilleur résultat possible dans le pire cas.

Pour maximiser la fonction g(x) = , nous introduisons une variable continue ***t*** pour représenter le minimum des . Le problème peut alors être formulé comme un programme linéaire :

**Variables :**

* : variable binaire,
* ***t***  : variable continue représentant le minimum des ) =

**Fonction objectif** :

**max *t***

**Contraintes :**

1. ***t*** ) =
2. ,

Cette modélisation a été implémentée dans un fichier Python (maxmin.py), utilisant le solveur Gurobi. Une solution optimale *x\**, maximisant ***t*** au sens du critère maxmin, a été déterminée. La solution obtenue est la suivante :

*x\**= (0, 1, 1, 1, 0, 0, 1, 1, 1, 0

La valeur de la fonction objective (maxmin) : 66

1. **Formulation du problème de minmax regret**

Le critère **minmax regret** cherche à minimiser le regret maximal parmi tous les scénarios possibles. Autrement dit, on cherche une solution qui réduit au minimum le "manque à gagner" dans le pire des cas.

Ce problème est résolu en deux étapes : la première consiste à calculer les regrets pour chaque scénario, et la seconde à maximiser le regret maximal à l'aide d'un programme linéaire.

**Étape 1 : Calcul des regrets**

Définissons les regrets *=) - ) =* ***) -*** où ) est la valeur maximale de sur tous les x. Ainsi, pour chaque scénario *i*, nous devons calculer).

**Étape 2 : Minimisation du regret maximal**

Pour minimiser la fonction g(x) = , introduisons une variable continue ***t*** pour représenter le maximum des regrets. Le problème peut alors être formulé comme un programme linéaire.

**Variables :**

* : variable binaire,
* ***t***  : variable continue représentant le maximum des

**Fonction objectif** :

**max *t***

**Contraintes :**

1. ***t*** *)* -
2. ,

Cette modélisation a été implémentée dans un fichier Python (minmaxRegret.py), en utilisant le solveur Gurobi. Gurobi nous a permis de trouveret en résolvant les programmes linéaires associés aux solutions optimales et respectivement. Par la suite, il a résolu le programme linéaire pour déterminer la solution optimale du regret minmax, fournissant les résultats suivants :

et

et

La valeur de la fonction objective (minmax regret) : 50

1. **Représentation des points dans le plan**

Nous souhaitons représenter dans le plan z1(x) et z2(x) les quatre points suivants :

*z(*)

*z(*(20, 118)

Les points et ne se trouvent pas sur la ligne reliant *z(*et *z(*. Cela démontre qu'aucune des solutions et n’aurait pu être obtenue en maximisant une moyenne pondérée (avec des coefficients positifs) entre *)* et ). Ces solutions sont spécifiques à leurs critères respectifs et ne peuvent être réduites à un compromis linéaire simple.

Si une solution optimale, telle que ou , pouvait être obtenue en maximisant une moyenne pondérée, cela signifierait qu’elle correspondrait à un compromis linéaire entre *)* et ), pondéré par des coefficients positifs. Cependant, les solutions maxmin et minmax regret , ne suivent pas cette logique.

Elles sont conçues pour répondre à des critères spécifiques :

* La solution maxmin vise à équilibrer les deux objectifs en minimisant le pire résultat possible.
* La solution minmax regret cherche à minimiser le regret maximal dans tous les scénarios possibles.

Ces critères ne sont pas alignés sur la logique de compromis linéaire (combinaison linéaire), car ils se concentrent sur des aspects plus complexes que la simple combinaison pondérée des deux objectifs. Ainsi, ces solutions ne peuvent être interprétées comme des résultats d'une optimisation basée sur une moyenne pondérée.