SORBONNE UNIVERSITY

M1 - MOGPL

MOGPL Project: Robust Optimization in Total Uncertainty

Authors:
Sama SATARIYAN
Ines RAHAOUI

December 2, 2024

Contents

Project Description and Objectives

The project focuses on robust optimization under total uncertainty, where scenarios impact the evaluation of solutions. Classical optimization seeks the optimal solution for a given objective function, but robust optimization considers all possible scenarios to ensure solutions perform well even in worst-case conditions.

The objective function $f(x)$ is optimized over a feasible set X, with evaluations depending on the scenario $s \in S$, a finite set of scenarios. Each solution $x \in X$ is evaluated as a vector $z(x) = (z_1(x), \ldots, z_n(x))$, where $z_i(x) = f(x, s_i)$ represents the value in scenario s_i.

The problem explores four robustness criteria:

- **Maximin**: Maximizes the worst-case performance.
- **Minmax Regret**: Minimizes the maximum regret over all scenarios.
- **MaxOWA (Ordered Weighted Average)**: Maximizes a weighted average, emphasizing weaker outcomes.
- **MinOWA of Regrets**: Minimizes a weighted sum of sorted regrets.

The aim is to linearize these criteria and apply them to the robust optimization problems described below. In this project, an Academic License for the Gurobi solver is used to solve the optimization problems.

Example 1

We consider a project selection problem among a list of 10 available projects, given that we have a budget of 100 Keuros. The cost of each project is known with certainty, but the utility of the project varies depending on the scenario considered. Here, we distinguish between two scenarios s_1, s_2 concerning the evolution of the socio-economic environment. The utility of the 10 projects in these two scenarios, as well as the costs of the projects, are given in the table below. The goal is to find the best subset of projects to select, taking into account both scenarios (the utilities are assumed to be additive across the projects).

Projects	1	2	3	4	5	6	7	8	9	10
Utility in s_1	70	18	16	14	12	10	8	6	4	2
Utility in s_2	2	4	6	8	10	12	14	16	18	70
Cost (in Keuros)	60	10	15	20	25	20	5	15	20	60

Example 2

We consider a problem with two scenarios $S = \{s_1, s_2\}$ and two instances provided below, where the cost vectors labeling the arcs represent the travel times in scenarios s_1 and s_2.

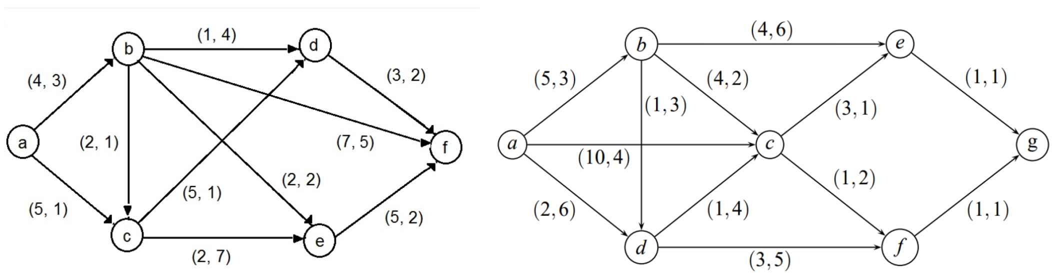

FIGURE 1: Two instances of the shortest path problem with 2 scenarios

Chapter 1

Example 1: Knapsack Problem

1 Linearization of Maximin and Minimax Regret

1.1 Maximin Criterion, Formulation

The first criterion, Maximin criterion, is formulated as:

$$\begin{gathered} max(g(x) = \min_i(z_i(x))) \\ i = 1, \ldots, 10; j = 1, 2 \end{gathered} \tag{1.1}$$

The goal is to maximize the worst-case scenario value $g(x)$ under a set of constraints, ensuring robustness in uncertain conditions. In order to do so, the function `ex1_maximin_approach` is implemented. It uses the Gurobi solver to determine the optimal subset of projects to select under uncertain conditions.

Problem Setup

A Gurobi model named `1_MaxiMin` is created. Binary decision variables $x_1, x_2, \ldots, x_{10}$ are defined. A continuous variable g is introduced, representing the worst-case scenario evaluation (the objective to maximize).

Objective Function

The objective function is set to maximize g, which represents the minimum evaluation across all scenarios: maximize g

Constraints

1. **Scenario Constraints:** The value of g is constrained to be less than or equal to the evaluation of the selected projects in each scenario:

$$g \leq 70x_1 + 18x_2 + 16x_3 + 14x_4 + 12x_5 + 10x_6 + 8x_7 + 6x_8 + 4x_9 + 2x_{10} \tag{1.2}$$

$$g \leq 2x_1 + 4x_2 + 6x_3 + 8x_4 + 10x_5 + 12x_6 + 14x_7 + 16x_8 + 18x_9 + 70x_{10} \tag{1.3}$$

2. **Budget Constraint:** The total cost of the selected projects must not exceed the available budget of 100 K€:

$$60x_1 + 10x_2 + 15x_3 + 20x_4 + 25x_5 + 20x_6 + 5x_7 + 15x_8 + 20x_9 + 60x_{10} \leq 100 \tag{1.4}$$

Optimization, Solution and the Result

As we see in the output part, the optimal value is $x^* = [0, 1, 1, 1, 0, 0, 1, 1, 1, 0]$ $(z_1, z_2) = (66, 66)$, so $g^* = 66$ as the objective value at the optimum level.

1.2 Minimax Regret Criterion, Formulation

For this criterion, formulated as:

$$\begin{aligned} &min(g(x) = \max_i r(x, s_i)); \\ &r(x, s_i) = z_i^* - z_i(x), \\ &z_i^* = \max_{y \in X} z_i(y) \\ &i = 1, \dots, 10; j = 1, 2 \end{aligned} \tag{1.5}$$

We first need to get z_1^* and z_2^*, as optimal outputs according to each of the scenarios. This one aims to minimize the maximum regret across all scenarios. The function `ex1_minmax_regret_approach` is implemented to solve this using the Gurobi solver.

The function `ex1_minmax_regret_approach_optz1` computes the optimal value z_1^*, which is the maximum utility achievable in the first scenario (s_1), subject to a budget constraint. The optimization problem is formulated as follows. For the objective we have:

$$maxz_1^* = 70x_1 + 18x_2 + 16x_3 + 14x_4 + 12x_5 + 10x_6 + 8x_7 + 6x_8 + 4x_9 + 2x_{10}, \tag{1.6}$$

For the constraints we have:

$$60x_1 + 10x_2 + 15x_3 + 20x_4 + 25x_5 + 20x_6 + 5x_7 + 15x_8 + 20x_9 + 60x_{10} \leq 100, \tag{1.7}$$

The output of this function includes $x_{s_1}^* = [1, 1, 1, 0, 0, 0, 1, 0, 0, 0]$ and $z_1^* = 112$

The function `ex1_minmax_regret_approach_optz2` computes the optimal value z_2^*, which is the maximum utility achievable in the second scenario (s_2), subject to the same budget constraint. The optimization problem is formulated as follows. For the objective we have:

$$maxz_2^* = 2x_1 + 4x_2 + 6x_3 + 8x_4 + 10x_5 + 12x_6 + 14x_7 + 16x_8 + 18x_9 + 70x_{10}, \tag{1.8}$$

For the constraints we have:

$$60x_1 + 10x_2 + 15x_3 + 20x_4 + 25x_5 + 20x_6 + 5x_7 + 15x_8 + 20x_9 + 60x_{10} \leq 100, \tag{1.9}$$

The output of this function includes $x_{s_2}^* = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1]$ and s_2 is $z_2^* = 118$

Problem Setup

A Gurobi model named `1_Minimax_Regret` is created. Binary decision variables $x_1, x_2, \dots, x_{10}$ represent whether a project is selected ($x_i = 1$) or not ($x_i = 0$). A continuous variable r is introduced, representing the maximum regret, which is the objective to minimize.

Objective Function

The objective function is set to minimize r, which is the maximum regret across all scenarios:

$$\text{minimize } r$$

Constraints

1. **Regret Constraints:** The regret r is constrained to be greater than or equal to the regret in each scenario:

$$r \geq z_1^* - (70x_1 + 18x_2 + 16x_3 + 14x_4 + 12x_5 + 10x_6 + 8x_7 + 6x_8 + 4x_9 + 2x_{10}) \tag{1.10}$$

$$r \geq z_2^* - (2x_1 + 4x_2 + 6x_3 + 8x_4 + 10x_5 + 12x_6 + 14x_7 + 16x_8 + 18x_9 + 70x_{10}) \tag{1.11}$$

2. **Budget Constraint:** The total cost of the selected projects must not exceed the available budget of 100 K€:

$$60x_1 + 10x_2 + 15x_3 + 20x_4 + 25x_5 + 20x_6 + 5x_7 + 15x_8 + 20x_9 + 60x_{10} \leq 100 \tag{1.12}$$

Optimization, Solution and the Result

The model is solved using Gurobi, and the output provides the optimal selection of projects (x^*), the utilities in both scenarios ($z_1(x^*), z_2(x^*)$), and the minimized maximum regret value ($g(x^*) = r$).

$$x^*[0, 1, 1, 0, 0, 1, 1, 1, 1, 0]$$

$$(z_1, z_2) = (62, 70)$$

$$(r_1, r_2) = (50, 42) \text{ therefore } g(x)^* = 50$$

1.3 Minimax Regret and Maximin Criteria - Analysis of Solutions

Putting x'^* (Minimax Regret optimum), x^* (Maximin optimum), x_1^* and x_2^* in the z_1-z_2 plane, we can see the results in the figure 1.1.

Connecting x_1^* and x_2^*, it is evident that the optima obtained from Maximin and Minimax Regret are not located on this line.

1.4 Generalization of the Knapsack Problem

For extending Example 1 to a robust knapsack problem involving p projects and n scenarios, we analyze how the resolution time depends on n and p. For this, 10 random instances are generated for each combination of n and p, where: $n \in \{5, 10, 15\}$ (number of scenarios), and $p \in \{10, 15, 20\}$ (number of projects), while the total budget is set to 50% of the total cost of all p projects. and project costs and utilities are sampled as integers in the range $[1, 100]$.

With the help of the function `comparing_different_scenario_project_combinations` we analyzes the average resolution times for various combinations of scenarios and projects: three different numbers of scenarios ($n = 15, 10, 5$) and three different numbers of projects ($p = 10, 15, 20$), generating 10 instances for each combination. For

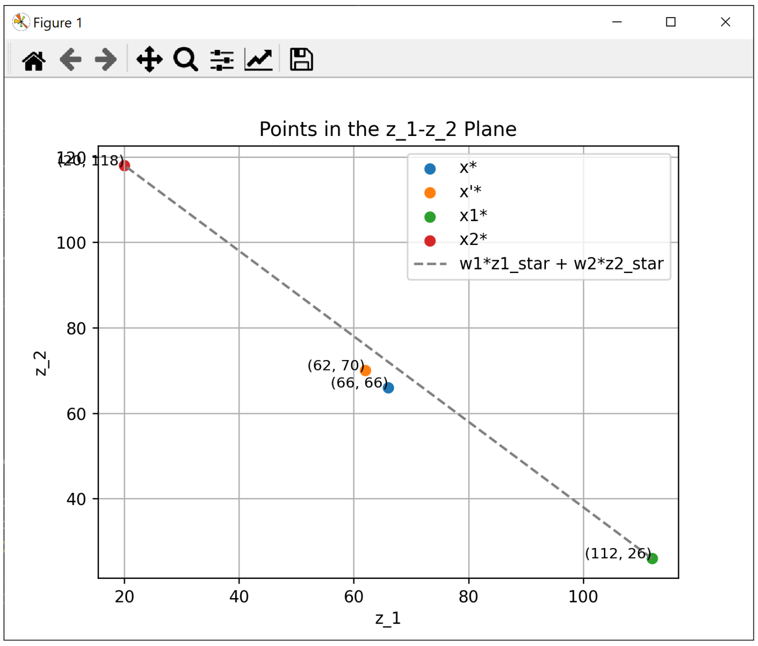

FIGURE 1.1: x^* and x'^* and Weighted sum of the Z_1^*, Z_2^* points

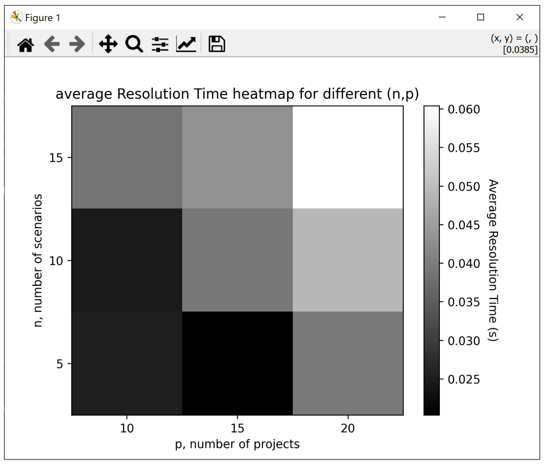

FIGURE 1.2: Average Resolution Time Heatmap

each (n, p) pair, we compute the average resolution time by calling the `resolution_time` function.

The heatmap in the Figure 1.2 is to visualize the relationship between scenarios, projects, and average resolution times, providing insights into how resolution time scales with different input sizes. We see that as the number of projects, scenarios, or both increase, the resolution time increases.

2 Linearization of the maxOWA Criteria

2.1 Optimal Value

Let $L(z)$ be the vector $(L_1(z), \ldots, L_n(z))$. Let us show that $L_k(z)$ is the optimal solution of the following linear program:

$$\begin{aligned} \min \sum_{i=1}^{n} a_{ik} z_i \\ \text{s.t.} \sum_{i=1}^{n} a_{ik} = k, \\ a_{ik} \in \{0,1\}, \quad \forall i = 1,\ldots,n \end{aligned} \tag{1.13}$$

The constraint number 1.13 amounts to taking the k smallest components of z_i. Since the components are sorted in ascending order, this is equivalent to taking the first k elements of z_i. $L_k(z)$, being the sum of the k smallest elements of z, is indeed the optimal solution.

2.2 Dual of the Linear Program

From the linear program, we obtain the following dual D_k:

$$\begin{aligned} \max\ kr_k - \sum_{i=1}^{n} b_{ik} \\ \text{s.c}: \quad r_k - b_{ik} \leq z_i, \quad \forall i = 1,\ldots,n, \\ r_k \in \mathbb{R}, \quad b_{ik} \geq 0, \quad \forall i = 1,\ldots,n. \end{aligned} \tag{1.14}$$

Solving the dual with Gurobi gives us the solution x^* =[2,6,11,17,25,34] with the vector $L[2,9,8,6,5,4]$.

2.3 Rewriting of the Equation

$$g(x) = \sum_{k=1}^{n} w_k z_{(k)}(x) \tag{1.15}$$

$$g(x) = w_1 z_{(1)} + (w_2 - w_1)L_2(z) + (w_3 - w_2)L_3(z) + \cdots + (w_n - w_{n-1})L_n(z)$$

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

2.4 MaxOWA On Example 1

We apply the given linear program to Example 1 and obtain:

$$\begin{aligned} \max\ w'_1 \left(1 \cdot r_1 - \sum_{i=1}^{2} b_{i1}\right) + w'_2 \left(2 \cdot r_2 - \sum_{i=1}^{2} b_{i2}\right) = r_1 - b_{11} - b_{21} + 2 \cdot r_1 - b_{12} - b_{22} \\ \sum_{i=1}^{10} c_i x_i \leq 100, \quad x_i \in \{0,1\} \\ r_1 - b_{i1} \leq z_i(x), \\ r_2 - b_{i2} \leq z_i(x), \\ i = 1,2 \quad b_{ik} \leq 0 \end{aligned} \tag{1.16}$$

The optimal solution is obtained by choosing projects 2, 3, 4, 7, 8, and 9, the value of the objective function is 198 with $z_1 = 18$ and $z_2 = 16$.

2.5 MinOWA of the Regrets

En se servant des réponses précédentes et en suivant la même démarche que pour maxOWA on trouve le programme linéaire suivant pour minOWA des regrets :

$$\min \sum_{k=1}^{n} w'_k r_k \tag{1.17}$$

$$r(x, s_i) = z_i^* - z_i(x)$$

$$r_1(x) \geq r_2(x) \geq \cdots \geq r_n(x), \quad \forall k \in \{1, \ldots, n\}.$$

$$r_k(x) \geq 0, \quad \forall k \in \{1, \ldots, n\}$$

Applying on the example 1, we obtain:

$$\min \sum_{k=1}^{n} w'_k r_k \tag{1.18}$$

$$\sum_{i=1}^{10} c_i x_i \leq 100, \quad x_i \in \{0, 1\}$$

$$r(x, s_1) = z_1^* - z_1(x)$$

$$r(x, s_2) = z_2^* - z_2(x)$$

$$r_1(x) \geq r_2(x)$$

$$r_k(x) \geq 0$$

We find that the optimal solution is obtained by selecting projects 2, 3, 5, 7, 8, and 9, and the value of the objective function is 150.

2.6 MinOWA VS MaxOWA - Comparison

Comparing the execution times of the two methods, we notice that the time difference is smaller when there are not many scenarios (5 scenarios). When increasing the number of scenarios to 5 or 10, the time difference becomes more significant, with minOWA taking three times longer to execute than maxOWA. This difference can be explained by the fact that minOWA needs to perform more calculations to compute the regrets.

Chapter 2

Example 2: Shortest Path Problem

1 Application to the Search for a Robust Path In a Graph

1.1 Shortest Path Linear Program Formulation

We are now looking for a linear program that allows us to find the fastest path from an initial vertex to a destination vertex in scenario s. Using the provided data, we obtain the following linear program.
Variables include binary x_{ij} representing the edge (i, j).

The objective function is:

$$\min \sum_{(i,j)\in A} t^s_{ij} \cdot x_{ij} \tag{2.1}$$

And the constraints :

$$\sum_{i:(i,k)\in A} x_{ik} - \sum_{j:(k,j)\in A} x_{kj} = \begin{cases} -1, & k = o, \\ 1, & k = d, \\ 0, & k \in A \setminus \{o, d\}, \end{cases} \tag{2.2}$$

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

1.2 4 Criteria Approaches for Both Instances - (Question 3.2 and 3.3 together)

We apply Maximin, Minimax Regret, MaxOWA and Min Regret OWA approaches for the two graphs shown at figure 1.1.

Minimax Regret Approach

First, we need to solve the problem for each scenario independently to get the z^*_i. For that we have the following modeling: The formulation of the model is quite like the maximin formulation explained in section 1.5.

Gurobi model named `ex3_1_opt_for_each_senario` is structured, and the objective is to minimize the sum

$$\sum_{(i,j)\in \text{Edges}} D_s[i, j] \cdot x[i, j] \tag{2.3}$$

, where $x[i, j]$ denotes the flow on edge (i, j).

Following the formulation 2.2, we have the three constraints: one for the flow of all the edges except the origin and the destination (enforcing that the total inflow into the node equals the total outflow from the node), one for the destination edge (sets node 5 as the termination point for exactly one unit of flow) and one for the origin edge (the source initiates one unit of flow into the network).

Upon solving the model, the function retrieves the optimal flow configuration, for each of the scenarios.
Here are the result of the two graphs and two scenarios:

TABLE 2.1: Optimal Paths and Solutions for Shortest Path Scenarios

Graph, Scenario	Optimal Path	Optimal Solution
left, 1	[(0, 1), (1, 3), (3, 5)]	8
left, 2	[(0, 2), (2, 3), (3, 5)]	4
right, 1	[(0, 3), (2, 5), (3, 2), (5, 6)]	5
right, 2	[(0, 2), (2, 4), (4, 6)]	6

After getting the minimum distance possible for each of the scenarios, we design the minimax regret model. The function `ex3_2_minimax_regret` as another Gurobi model is introduced.
For variables we have: binary variables $x[i,j]$ as well as continuous variable r.
We set the objective as :

$$\min r \tag{2.4}$$

On top of the three constraints explained before, we set two other constraints:

$$r \geq (max_y(D_{s1}[i,j] \cdot y[i,j])) - \sum(D_{s1}[i,j] \cdot x[i,j]); \quad \forall (i,j) \in A \tag{2.5}$$

$$r \geq (max_y(D_{s2}[i,j] \cdot y[i,j])) - \sum((D_{s2}[i,j] \cdot x[i,j]); \quad \forall (i,j) \in A \tag{2.6}$$

The result of the program for the left graph is as following. The optimization model successfully identified the optimal path with a minimax regret value of

$$r = 0.0 \quad \text{for the left graph} \tag{2.7}$$

$$r = 0.0 \quad \text{for the right graph} \tag{2.8}$$

The objective value achieved reflects the minimum possible regret across the considered scenarios. They are both zero, meaning that the model's solution perfectly matches to the scenario requirements, leading to no regret. The optimal path determined by the model for the left graph is:

- Edge (0, 1) with capacities $D_{s1} : 4$, $D_{s2} : 3$
- Edge (1, 3) with capacities $D_{s1} : 1$, $D_{s2} : 4$
- Edge (3, 5) with capacities $D_{s1} : 3$, $D_{s2} : 2$

And for the right graph is:

- Edge(0, 1) with capacities $D_{s1} : 5$, $D_{s2} : 3$
- Edge (1, 4) with capacities $D_{s1} : 4$, $D_{s2} : 6$
- Edge (4, 6) with capacities $D_{s1} : 1$, $D_{s2} : 1$

Therefore the optimal path is given by:

$$[(0,1),(1,3),(3,5)] \quad \text{for the left graph} \tag{2.9}$$

$$[(0,1),(1,4),(4,6)] \quad \text{for the right graph} \tag{2.10}$$

Maximin Approach

In this approach, what we did was following the exact formulation of the last example 1.1. For variables we have: binary variables $x[i,j]$ as well as continuous variable g.
We set the objective as :

$$\max g \tag{2.11}$$

On top of the three constraints explained before, we set two other constraints:

$$g \leq \sum((D_{s1}[i,j] \cdot x[i,j])); \quad \forall(i,j) \in A \tag{2.12}$$

$$g \leq \sum((D_{s2}[i,j] \cdot x[i,j])); \quad \forall(i,j) \in A \tag{2.13}$$

For the output, the solver gives us this path as the optimal path:

$$[(0,1),(1,2),(2,4),(4,5)] \quad \text{for the left graph} \tag{2.14}$$

The optimal path determined by the model for the left graph is:

- Edge (0, 1) with capacities $D_{s1}: 4$, $D_{s2}: 3$
- Edge (1, 2) with capacities $D_{s1}: 2$, $D_{s2}: 1$
- Edge (2, 4) with capacities $D_{s1}: 2$, $D_{s2}: 7$
- Edge (4, 5) with capacities $D_{s1}: 5$, $D_{s2}: 1$

And the optimal solution is:

$$g = 12.0 \tag{2.15}$$

$$[(0,1),(1,3),(2,4),(3,2),(4,6)] \quad \text{for the right graph} \tag{2.16}$$

The optimal path determined by the model for the left graph is:

- Edge (0, 1) with capacities $D_{s1}: 5$, $D_{s2}: 3$
- Edge (1, 3) with capacities $D_{s1}: 1$, $D_{s2}: 3$
- Edge (2, 4) with capacities $D_{s1}: 3$, $D_{s2}: 1$
- Edge (3, 2) with capacities $D_{s1}: 1$, $D_{s2}: 4$
- Edge (4, 6) with capacities $D_{s1}: 1$, $D_{s2}: 1$

And the optimal solution is:

$$g = 11.0 \tag{2.17}$$

MaxOWA approach

We use the same formulation explained at the equation 1.16.

$$[(0,1),(1,2),(2,4),(4,5)] \quad \text{for the left graph} \tag{2.18}$$

The optimal path determined by the model for the left graph is:

- Edge (0, 1) with capacities $D_{s1}: 4$, $D_{s2}: 3$
- Edge (1, 2) with capacities $D_{s1}: 2$, $D_{s2}: 1$

- Edge (2, 4) with capacities D_{s1} : 2, D_{s2} : 7
- Edge (4, 5) with capacities D_{s1} : 5, D_{s2} : 1

And the optimal solution is:

$$50.0 \tag{2.19}$$

$$[(0,1),(1,3),(3,5)] \quad \text{for the right graph} \tag{2.20}$$

The optimal path determined by the model for the right graph is:

- Edge (0, 1) with capacities D_{s1} : 4, D_{s2} : 3
- Edge (1, 3) with capacities D_{s1} : 1, D_{s2} : 3
- Edge (3, 2) with capacities D_{s1} : 1, D_{s2} : 4
- Edge (2, 4) with capacities D_{s1} : 3, D_{s2} : 1
- Edge (4, 6) with capacities D_{s1} : 1, D_{s2} : 1

And the optimal solution is:

$$40.0 \tag{2.21}$$

It's the same path than we found with maximin approach.

MinOWA approach

By using the linear program from question 2.5, we do not find feasible solutions with Gurobi. This can be explained by the fact that the minOWA criterion is highly penalizing for poor scenarios. However, when we remove the constraint on regrets, we find a feasible and optimal solution.

1.3 Generalization of the Shortest Path Problem

For extending Example 2 to a robust Shortest Path problem involving p nodes and n scenarios, we analyze how the resolution time depends on n and p. For this, 10 random instances are generated for each combination of n and p, where: $n \in \{10, 5, 2\}$ (number of scenarios), and $p \in \{10, 15, 20\}$ (number of nodes), while the distances are sampled as integers in the range $[1, 100]$ and the density of the arcs are sampled randomly fr As we see, the more scenarios we have, the higher the resolution time gets. Also, the bigger the graph is, the more the resolution time is.

For the Minimax Regret approache, the results are shown in the Figure 2.1 and table 2.2.

For the Maximin approach, the results are shown in the Figure 2.2 and table 2.2.

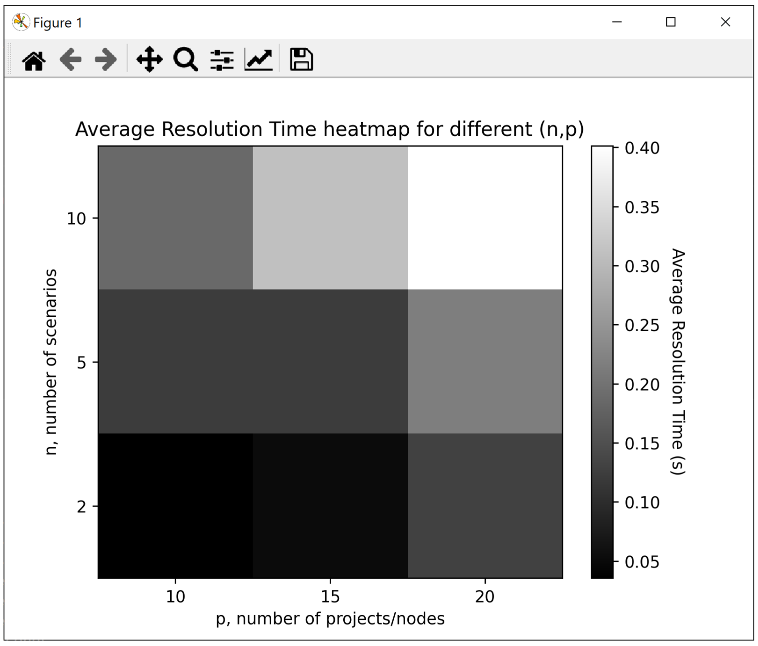

FIGURE 2.1: Average Resolution Time Heatmap - Minimax Regret

Scenarios (n)	Nodes (p)	Avg Time (s)
2	10	0.0511
2	15	0.0742
2	20	0.0979
5	10	0.1070
5	15	0.1580
5	20	0.2338
10	10	0.1893
10	15	0.2626
10	20	0.4299

TABLE 2.2: Average Resolution Times for Different Numbers of Scenarios and Nodes - Minmax regret

Scenarios (n)	Nodes (p)	Avg Time (s)
2	10	0.0182
2	15	0.0246
2	20	0.0646
5	10	0.0473
5	15	0.0451
5	20	0.1116
10	10	0.0481
10	15	0.0963
10	20	0.1626

TABLE 2.3: Average Resolution Times for Different Numbers of Scenarios and Nodes - Maximin

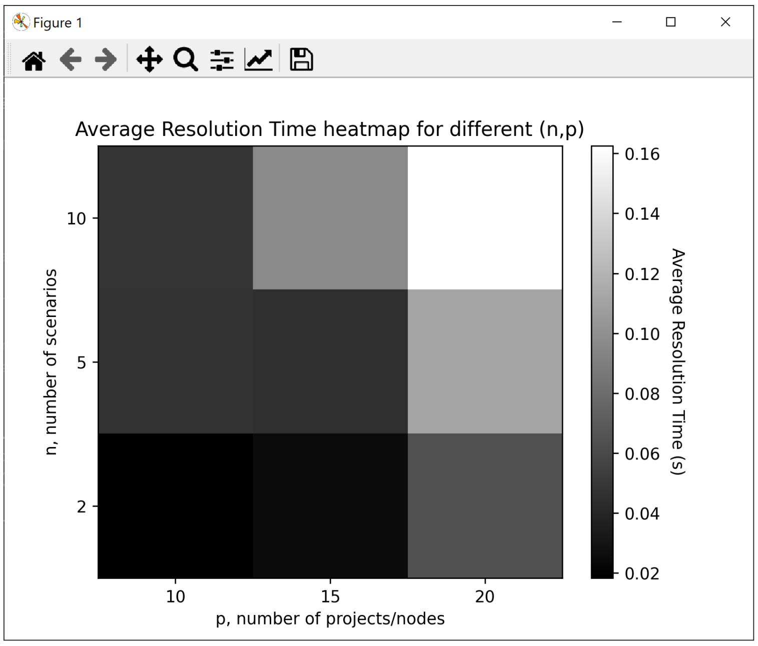

FIGURE 2.2: Average Resolution Time Heatmap - Maximin

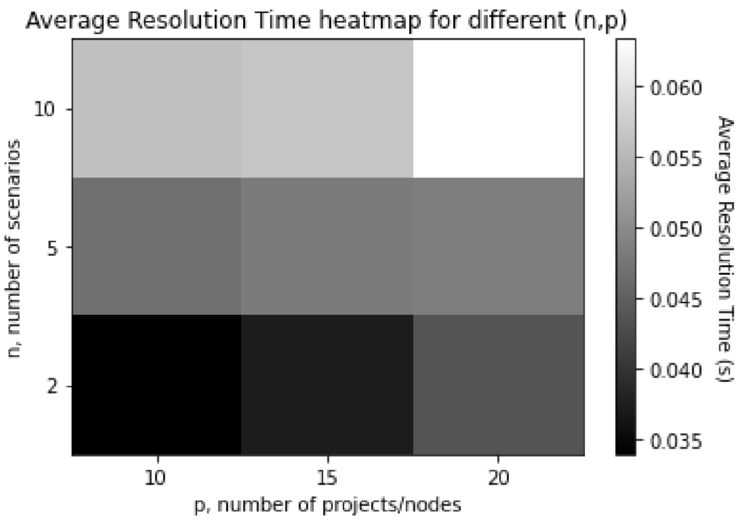

FIGURE 2.3: Average Resolution Time Heatmap - MaxOWA

Scenarios (n)	Nodes (p)	Avg Time (s)
2	10	0.0339
2	15	0.0372
2	20	0.0436
5	10	0.0469
5	15	0.0482
5	20	0.0484
10	10	0.0560
10	15	0.0566
10	20	0.0633

TABLE 2.4: Average Resolution Times for Different Numbers of Scenarios and Nodes - MaxOWA

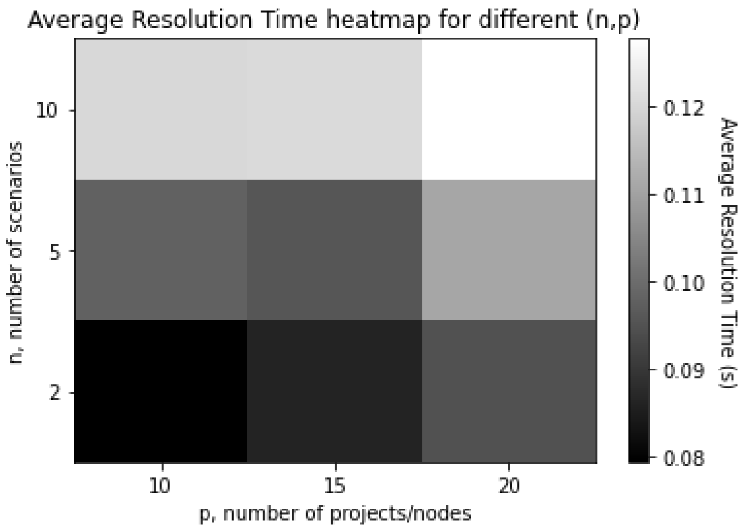

FIGURE 2.4: Average Resolution Time Heatmap - MinOWA

Scenarios (n)	**Nodes (p)**	**Avg Time (s)**
2	10	0.0795
2	15	0.0862
2	20	0.0947
5	10	0.0978
5	15	0.0958
5	20	0.1107
10	10	0.1203
10	15	0.1206
10	20	0.1277

TABLE 2.5: Average Resolution Times for Different Numbers of Scenarios and Nodes - MinOWA