Stochastic Optimization

Report

Assignment 2024/25

Di Battista Simona — 302689
Rostagno Andrea — 349152

June 15, 2025

Contents

1 Introduction

Stochastic optimization addresses decision-making problems under uncertainty, where parameters not known in advance are modeled as random variables. Unlike deterministic optimization, stochastic optimization aims to identify optimal decisions by considering the probabilistic distribution of future scenarios. In this project, we specifically focus on two classical stochastic problems: the Newsvendor Problem and the Assemble-to-Order (ATO) problem. The Newsvendor Problem involves determining the optimal quantity of newspapers to order under uncertain demand, to maximize expected profits. In contrast, the ATO problem addresses component inventory and assembly decisions to satisfy uncertain demand for multiple products.

To analyze these problems, scenarios were generated through a predefined scenario generation code provided beforehand. Subsequently, to reduce computational complexity, were applied two scenario reduction methods:

- K-means clustering;
- an heuristic method, based on Wasserstein distance.

Finally, were compared the results to evaluate effectiveness and stability of each strategy.

2 Problems explanation

In this section, are presented the two stochastic optimization problems analyzed in this report: the Newsvendor Problem and the Assemble-to-Order (ATO) Problem. Both are classical examples of two-stage stochastic programs, characterized by decisions that must be made before uncertain demand is revealed. First each problem is introduced and mathematically formulated, then a schematic representations is given to aid understanding. Finally, these problems are solved and analyzed, using different scenario generation and reduction methods.

2.1 Newsvendor Problem

The newsvendor problem is a classical example in stochastic optimization used to determine the optimal inventory level when demand is uncertain. In our analysis, a vendor must decide the optimal number of newspapers to buy at the beginning of a day, without knowing the exact daily demand, with the goal of maximizing total expected profit.

2.1.1 Problem Formulation

Formally, the problem is formulated as follows:

$$\max_{x} \mathbb{E}[p \min(D(\omega), x) - cx]$$

with:

- x: decision variable representing the number of newspapers to buy;
- $D(\omega)$: a random variable modeling daily demand, characterized by discrete scenarios d_s, each with probability π_s;
- p: selling price per newspaper.
- c: cost per newspaper.

In our specific implementation:

- $c = 1$;
- $p = 10$.

2.1.2 Optimization Model

The model is implemented using the following integer linear programming formulation:

$$
\begin{aligned}
\max \quad & p \sum_{s \in S} \pi_s y_s - cx & \\
\text{s.t.} \quad & y_s \leq x, & \forall s \in S \\
& y_s \leq d_s, & \forall s \in S \\
& x, y_s \geq 0 \quad \text{and integers}, & \forall s \in S
\end{aligned}
$$

where y_s represents the actual number of newspapers sold if scenario s is realized.

2.2 Assemble-to-Order (ATO) Problem

The Assemble-to-Order (ATO) problem addresses decision-making in manufacturing systems, where final products are assembled from a set of pre-produced components once customer orders are realized. This two-stage stochastic program involves:

- **first stage**: decide the quantities of components to produce.
- **second stage**: determine the assembly quantities of final products, once demand is known.

2.2.1 Problem Formulation

The mathematical formulation of the ATO problem is given by:

$$
\begin{aligned}
\max \quad & -\sum_{i \in \mathcal{I}} C_i x_i + \mathbb{E}\left[\sum_{j \in \mathcal{J}} P_j y_j(\omega)\right] & \\
\text{s.t.} \quad & \sum_{i \in \mathcal{I}} T_{im} x_i \leq L_m, & \forall m \in \mathcal{M} \\
& y_j(\omega) \leq d_j(\omega), & \forall j \in \mathcal{J}, \forall \omega \in \Omega \\
& \sum_{j \in \mathcal{J}} G_{ij} y_j(\omega) \leq x_i, & \forall i \in \mathcal{I}, \forall \omega \in \Omega \\
& x_i, y_j(\omega) \geq 0, & \forall i \in \mathcal{I}, j \in \mathcal{J}, \omega \in \Omega
\end{aligned}
$$

with:

- x_i: decision variable representing the amount of component $i \in \mathcal{I}$ (set of components) to produce;

- $y_j(\omega)$: amount of item $j \in \mathcal{J}$ (set of final items) assembled after demand realization;
- $d_j(\omega)$: stochastic demand for item j in scenario $\omega \in \Omega$.
- C_i: cost of component i;
- P_j: selling price of item j;
- L_m: availability of machine m;
- T_{im}: time required to produce component i on machine m;
- G_{ij}: amount of component i required to assemble item j (Gozinto factor);
- $\mathcal{M}$: set of machines.

In our specific implementation, we considered the example of the pizza maker, with six hours of work available, two different types of pizzas to be able to produce and the following ingredients on hand: dough, tomato sauce, vegetables. The parameters in the optimization problem are set as follows:

- $C = [1, 1, 3]$.
- $P = [6, 8.5]$.
- $T = [0.5, 0.25, 0.25]$.
- $L = 6.0$ hours.
- Gozinto matrix: $G = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$.

These parameters and constraints form the basis of our computational experiments.

3 The class ScenarioTree

The scenario tree is a fundamental tool in stochastic optimization used to represent and manage uncertainty through a structured set of possible future scenarios. In this study, scenario trees are generated using a two-step process involving initial probabilistic models and a specialized Python class called `ScenarioTree`. This chapter contains the aspects of the above class most relevant to the development of the assignment.

3.1 Initial Probabilistic Model

Initially, scenarios are generated using a stochastic model defined by the class `EasyStochasticModel`. This class uses a multivariate normal distribution characterized by specified averages and variances; a fixed number of scenarios are sampled according to the probability distribution

$$\text{Obs} \sim \mathcal{N}(\mu, \Sigma),$$

which parameters depend on the problem under consideration.

3.2 Scenario Tree Generation

The `ScenarioTree` class is responsible for constructing and managing the tree structure. The tree is built iteratively, where each node generates child nodes according to the stochastic model described above. Each node has attributes:

- **obs**: observation values at the current node;
- **prob**: conditional probability of reaching the current node from its parent;
- **path_prob**: cumulative probability from the root node to the current node.

Formally, for each node at stage t, child nodes are generated as follows:

$$\text{obs}_j^{(t+1)} \sim \text{StochModel}\Big(\text{obs}^{(t)}\Big), \quad j = 1, \ldots, \text{branching\_factor}_t$$

The tree generation continues until the predefined depth (planning horizon) is reached, creating a comprehensive structure of all possible demand outcomes and associated probabilities. In this study, two-stage problems were analyzed and scenario trees of depth one were generated.

MAGARI METTERLO NELLA CAPTION DELL'IMMAGINE: the resulting scenario tree can be visualized graphically, clearly illustrating the branching structure, node values, and probabilities. This visual aid significantly enhances the interpretation of how uncertainty evolves over time and assists in understanding the implications of different scenario reduction methods presented later in the report.

4 Results with all scenarios

In this section, are reported the results obtained by solving the stochastic optimization problems considering the full set of demand scenarios, without applying any reduction technique. This approach provides a benchmark solution, capturing the entire variability of the underlying random variables. The outcomes presented here will be useful as a reference for evaluating the accuracy and computational efficiency of the scenario reduction methods discussed in the following sections.

4.1 Newsvendor Problem

The following results summarize the optimization outcomes of the Newsvendor problem when considering all initially generated scenarios. From an initial fixed root, N nodes sampled from a multivariate normal of size 50 were generated; each node was generated from the following distribution:

$$\mathcal{N}_{50}(\mu, \Sigma),$$

with $\mu = [25, \dots, 25]$ and covariance matrix $\Sigma = 200 * I_{50}$. The code generates N samples of size 50 to obtain N realizations of the expected value of profit, in order to calculate statistics and draw results and conclusions about the solution of the problem. The choice to place the initial number of scenarios equal to 50 was related to the use of a limited Gurobi license, while the choice of μ and Σ parameters was dictated by the possibility of generating data that encapsulated some variability. Of course, the scenario tree generated through this procedure returns demand values as floating-point numbers. However, since the Newsvendor problem inherently deals with discrete units (we are talking about newspapers), these continuous observations must be rounded to the nearest integer and constrained to be non-negative, ensuring practical and realistic demand scenarios. This rounding and aggregation procedure is implemented in the code using the

provided `aggregate_discrete_demands` function, which combines scenarios with identical integer demands, summing their probabilities.
In summary, the newsvendor problem is solved N times for each generated set, which as a result of aggregation will have cardinality less than or equal to 50. This does not conflict with the estimates to be obtained, but rather is representative of the variability of the demand type (the same number of scenarios may not always occur). Table (4.1) shows an example of the output of a sample generated through the scenario tree, whose inputs were aggregated as necessary.

Demand (d)	**Probability (π)**	**Demand (d)**	**Probability (π)**	**Demand (d)**	**Probability (π)**
0	0.06	19	0.08	39	0.06
1	0.02	21	0.02	42	0.02
4	0.02	23	0.04	43	0.02
5	0.02	24	0.02	46	0.04
7	0.02	25	0.04	48	0.02
10	0.02	29	0.04	56	0.04
12	0.02	30	0.04	57	0.02
13	0.06	31	0.02	59	0.02
14	0.02	33	0.02	**Total**	**1.00**
15	0.04	35	0.04		
16	0.04	38	0.04		
17	0.02				

Table 1: An example of discrete demand values and aggregated probabilities used in the Newsvendor model. First, an initial set with cardinality equal to 50 of equiprobabilistic scenarios was generated. Following the aggregation operation, the scenarios are reduced to 31, with probabilities no longer all equal.

The plot in (Figure 1) illustrating an example of scenario tree with the generated demand values, their probabilities, and the structure of uncertainty before aggregation.

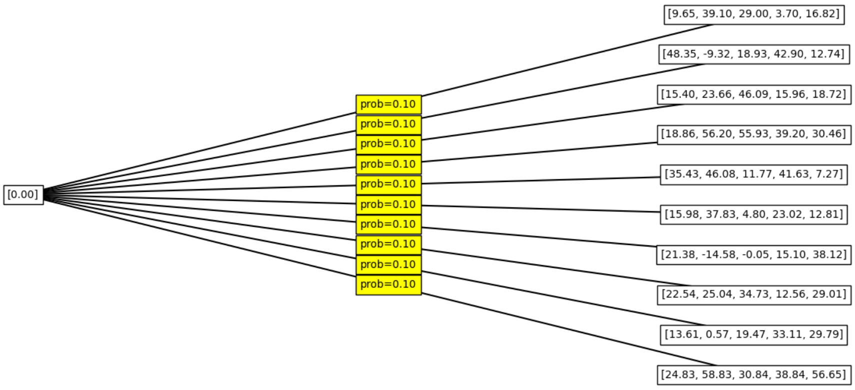

Figure 1: Visualization of an example of scenario tree, used in the analysis of the Newsvendor problem. The tree is composed of 10 equiprobabilistic sets, each of dimension 5 sampled from a multivariate normal distribution of dimension 5 with parameters $\mu = 25, \sigma^2 = 200$, before rounding and aggregation.

..Results..

$\bar{p}_N$	s_N^2	t_S
1	3	0.16

4.2 ATO Problem

This subsection presents the results of the Assemble-to-Order (ATO) optimization model using the complete set of scenarios generated by our stochastic model.

In this case N sets of observations sampled from a multivariate normal of size 50 were generated for each type of pizza considered in the problem. Specifically, the parameters $\mu_1 = 90, \Sigma_1 = 200 * I_{50}$ were considered for the first typology, while the parameters $\mu_2 = 100, \Sigma_2 = 2000 * I_{50}$ were considered for the second one. Even in this case this procedure is used to obtain samples of the expected value of profit, to calculate statistics useful for drawing conclusions about the problem. The choice to place the number of scenarios equal to 50 was related to the use of a limited Gurobi license, while the choice of μ and Σ parameters was dictated by the possibility of generating data that encapsulated some variability. The generated demands, initially provided as continuous values from a multivariate normal distribution, must be discretized to integer values due to the practical constraints of real-world production systems. Specifically, demands are rounded to the nearest multiple of ten, ensuring realistic and manageable scenarios. This procedure is implemented by the provided function `aggregate_vectorial_demands`, which aggregates the probabilities of scenarios with identical discretized demands. In summary, also the ATO problem is solved N times for each generated set, which as a result of aggregation will have cardinality less than or equal to 50. This does not conflict with the estimates to be obtained, but rather is representative of the variability of the demand type (the same number of scenarios may not always occur). Table (4.2) shows an example of the output of a sample generated through the scenario tree, whose inputs were aggregated as necessary.

Demand (Item 1, Item 2)	π	**Demand (Item 1, Item 2)**	π	**Demand (Item 1, Item 2)**	π
`[60, 190]`	0.02	`[80, 270]`	0.02	`[100, 100]`	0.02
`[70, 160]`	0.04	`[80, 280]`	0.02	`[100, 150]`	0.02
`[80, 120]`	0.02	`[90, 90]`	0.02	`[100, 160]`	0.04
`[80, 140]`	0.04	`[90, 150]`	0.02	`[100, 180]`	0.04
`[80, 150]`	0.02	`[90, 160]`	0.02	`[100, 200]`	0.04
`[80, 170]`	0.02	`[90, 170]`	0.02	`[100, 210]`	0.02
`[80, 190]`	0.04	`[90, 180]`	0.04	`[100, 230]`	0.02
`[80, 210]`	0.04	`[90, 200]`	0.02	`[100, 260]`	0.02
`[80, 220]`	0.02	`[90, 220]`	0.04	`[100, 270]`	0.04
`[80, 240]`	0.02	`[90, 240]`	0.04	`[110, 170]`	0.02
`[80, 250]`	0.04	`[90, 290]`	0.02	`[110, 240]`	0.04
`[80, 260]`	0.04	`[90, 310]`	0.02	`[110, 300]`	0.02
				Total	**1.00**

Table 2: An example of discrete demand values and aggregated probabilities used in the ATO model. First, an initial set with cardinality equal to 50 of equiprobabilistic scenarios (each scenario is a two-dimensional vector) was generated. Following the aggregation operation, the scenarios are reduced to 36, with probabilities no longer all equal.

The plot in Figure (2) illustrating an example of scenario tree for an ATO problem with the generated demand values, their probabilities, and the structure of uncertainty before aggregation.

..Results..

$\bar{p}_N$	s_N^2	t_S
1	3	0.16

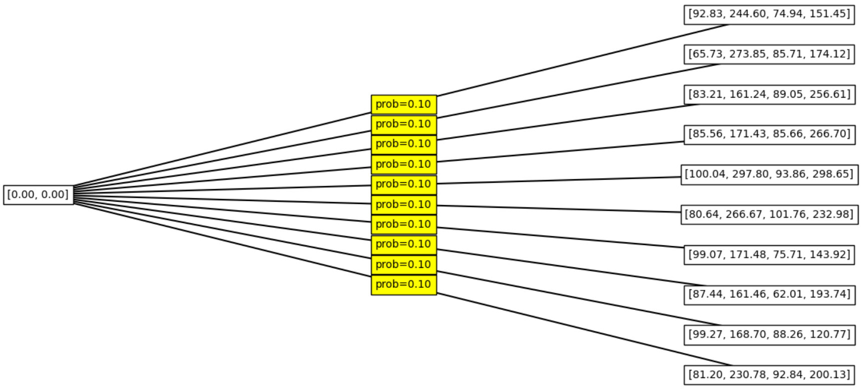

Figure 2: Visualization of an example of scenario tree, used in the analysis of the ATO problem. The tree is composed of 10 equiprobabilistic sets, each of dimension 4, composed of two scenarios for Item 1 and two scenarios for Item 2. They are sampled respectively from a multivariate normal distribution of dimension 2 with parameters $\mu_1 = 90, \Sigma_1 = 200 * I_2$, and $\mu_2 = 100, \Sigma_2 = 2000 * I_2$ before rounding and aggregation.

5 Results with K-means reduction

In this section, we present the results obtained by applying the K-means clustering method to reduce the number of scenarios and computational complexity generated in both the Newsvendor and Assemble-to-Order (ATO) problems. K-means is a clustering algorithm that partitions the original set of scenarios into k clusters by minimizing the weighted sum of squared distances between scenario values and their respective cluster centroids. Specifically, each original scenario is assigned to a cluster based on its proximity to the cluster's centroid, taking into account the scenario probabilities as weights. After the clustering process, the centroids become representative of the reduced scenarios, and the probabilities of these scenarios are recalculated by summing the probabilities of all original scenarios within each cluster.
K-means clustering were implemented to reduce the number of original scenarios generated for both problems. The algorithm was applied to each of the N samples, to reduce the numerosity of each sample to k, with $k \in [1, 15]$. The upper bound of the interval was chosen such that the number of clusters obtained was significantly smaller than the original number of scenarios. Each resulting cluster, in fact, is represented by its centroid, which acts as a reduced scenario, with the new scenario probability obtained by summing the probabilities of the original scenarios assigned to that cluster. Furthermore, for each of the N samples, the SSE trend graph was plotted to identify the appropriate number of points (clusters) to represent each initial set, according to the algorithm.

Below, we present detailed analyses of the performance of this scenario reduction method in the two considered optimization problems.

5.1 Newsvendor Problem

In order to analyze the newsvendor problem with scenario reduction, $\forall k \in [1, 15]$ the following were given in the table (5.1):

- $\bar{p}_N$: estimate of the expected value of profit, obtained by considering the sample mean computed using the N values from solving the problem for each sample;
- s^2_N : sample variance of the expected values of the profit;
- t_R : average computation time for reducing to k scenarios;
- t_S : average computation time for solving the Newsvendor problem with k scenarios;

# Cluster (k)	$\bar{p}_N$	s^2_N	t_R	t_S
1	3	0.16	0	0
2	18	0.40	0	0
3	34	0.26	0	0
4	45	0.10	0	0
5	57	0.08	0	0
6	57	0.08	0	0
7	57	0.08	0	0
8	57	0.08	0	0
9	57	0.08	0	0
10	57	0.08	0	0
11	57	0.08	0	0
12	57	0.08	0	0
13	57	0.08	0	0
14	57	0.08	0	0
15	57	0.08	0	0

DA TENERE SE E' VERO: these results demonstrate how scenario reduction via K-means effectively simplifies the scenario representation while preserving the essential characteristics needed to achieve reliable decision-making outcomes in stochastic optimization contexts.

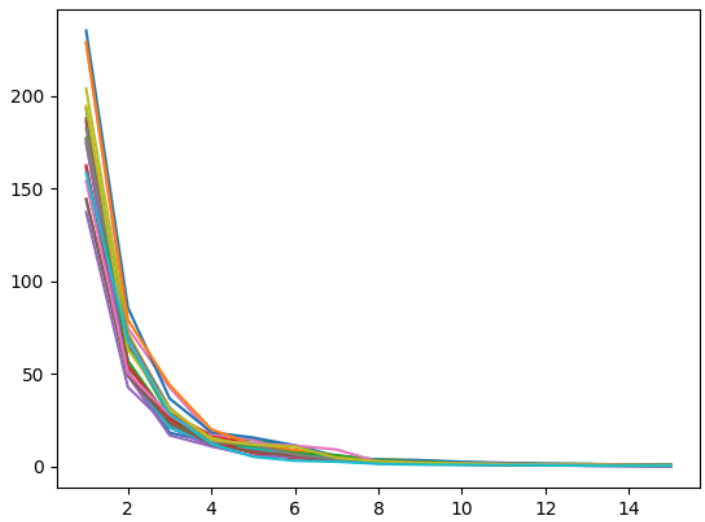

Figure 3: SSE trend graph for $k \in [1, 15]$ in case of Newsvendor problem. The figure shows that the appropriate number of points to which to reduce the numerosity of each set is around 3 and 4.

5.2 ATO Problem

In this subsection, we present the results obtained for the Assemble-to-Order (ATO) problem after applying scenario reduction via K-means clustering, to identify representative scenarios in a two-dimensional space (corresponding to the two final products). In order to analyze the newsvendor problem with scenario reduction, $\forall k \in [1, 15]$ the following were given in the table (5.2):

- $\bar{p}_N$: estimate of the expected value of profit, obtained by considering the sample mean computed using the N values from solving the problem for each sample;
- s_N^2 : sample variance of the expected values of the profit;
- t_R : average computation time for reducing to k scenarios;
- t_S : average computation time for solving the Newsvendor problem with k scenarios;

# Cluster (k)	$\bar{p}_N$	s_N^2	t_R	t_S
1	3	0.16	0	0
2	18	0.40	0	0
3	34	0.26	0	0
4	45	0.10	0	0
5	57	0.08	0	0
6	57	0.08	0	0
7	57	0.08	0	0
8	57	0.08	0	0
9	57	0.08	0	0
10	57	0.08	0	0
11	57	0.08	0	0
12	57	0.08	0	0
13	57	0.08	0	0
14	57	0.08	0	0
15	57	0.08	0	0

DA TENERE SE VERO: the scenario reduction via K-means preserved the essential characteristics of the demand distribution, allowing us to significantly reduce the problem size while maintaining optimality in the solution.

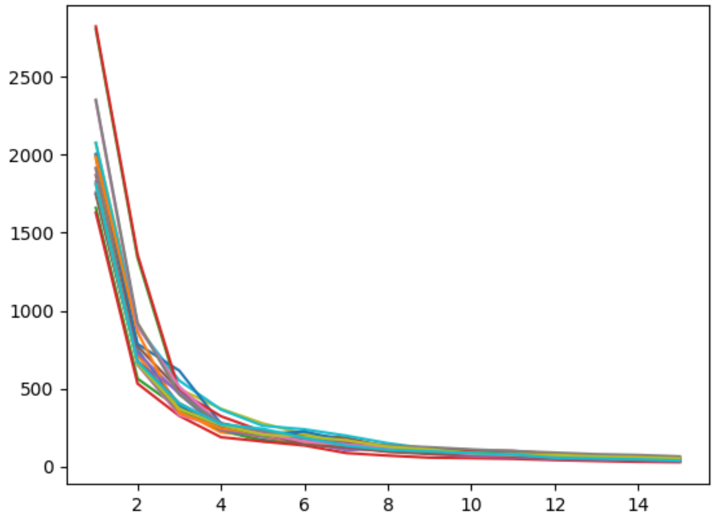

Figure 4: SSE trend graph for $k \in [1, 15]$ in case of ATO problem. The figure shows that the appropriate number of points to which to reduce the numerosity of each set is around 3 and 4.

6 Results with Wasserstein distance–based reduction

In this section, we present the scenario reduction method based on the Wasserstein distance, as applied to both the Newsvendor and Assemble-to-Order (ATO) problems. The Wasserstein distance, also known as the Earth Mover's Distance, is a mathematical metric used to quantify the dissimilarity between two probability distributions on a given metric space. In the context of scenario reduction, it measures the minimum "cost" required to transform the original probability distribution of scenarios into a reduced one, where the "cost" is defined as the amount of probability mass to move times the distance it is moved.

Formally, let $\mu = (\mu_1, \ldots, \mu_m)$ and $\nu = (\nu_1, \ldots, \nu_n)$ be two discrete probability distributions on points $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$. The p-Wasserstein distance is defined as:

$$W_p(\mu, \nu) = \left(\min_{\gamma \in \Gamma(\mu,\nu)} \sum_{i=1}^{m} \sum_{j=1}^{n} \|x_i - y_j\|^p \gamma_{ij} \right)^{1/p}$$

where γ_{ij} is the transport plan representing the amount of mass moved from x_i to y_j, and $\Gamma(\mu, \nu)$ is the set of admissible transport plans (satisfying mass conservation constraints).

In our scenario reduction approach, we use an exact mixed-integer programming (MILP) formulation to select a subset of k representative scenarios from the original m scenarios, such that the Wasserstein distance between the original and reduced distributions is minimized. The core optimization problem, implemented in our code, is as follows (for the unidimensional case):

$$
\begin{aligned}
\min_{\gamma,z,\nu} \quad & \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}\gamma_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{m} \gamma_{ij} = \mu_i \qquad \forall i = 1,\dots,m \\
& \sum_{i=1}^{m} \gamma_{ij} = \nu_j \qquad \forall j = 1,\dots,m \\
& \nu_j \le z_j \qquad \forall j = 1,\dots,m \\
& \sum_{j=1}^{m} z_j = k \\
& \sum_{j=1}^{m} \nu_j = 1 \\
& z_j \in \{0,1\}, \quad \gamma_{ij}, \nu_j \ge 0
\end{aligned}
$$

where:

- $c_{ij} = |x_i - x_j|^p$ is the cost of moving mass from scenario i to scenario j (L^p norm);
- γ_{ij} is the amount of probability mass transported from i to j;
- z_j is a binary variable indicating whether scenario j is selected in the reduced set;
- ν_j is the probability assigned to scenario j in the reduced distribution.

This model ensures that exactly k scenarios are selected ($\sum_j z_j = k$), the reduced probabilities sum to 1, and the transportation of probability mass is minimized according to the cost matrix C. The same approach is extended to the multidimensional case (for ATO) using the appropriate vector norms for the cost computation.

The algorithm was applied to each of the N samples, to reduce the numerosity of each sample to k, with $k \in [1, 15]$. The upper bound of the interval was chosen such that the number of clusters obtained was significantly smaller than the original number of scenarios. The Wasserstein reduction technique, by construction, selects scenarios that preserve the probabilistic structure of the original distribution as faithfully as possible, yielding a reduced scenario set that guarantees a minimal loss of information with respect to the original distribution. In the following subsections, we detail the results obtained by applying this method to our stochastic optimization problems.

6.1 Newsvendor Problem

For the Newsvendor problem, we applied the scenario reduction method based on the Wasserstein distance, which is designed to preserve the probabilistic structure of the original set of scenarios as closely as possible. The resulting reduced problems are summarized in Table 6.1, where $\forall k \in [1, 15]$ the following were given:

- $\bar{p}_N$: estimate of the expected value of profit, obtained by considering the sample mean computed using the N values from solving the problem for each sample;
- s_N^2 : sample variance of the expected values of the profit;
- t_R : average computation time for reducing to k scenarios;
- t_S : average computation time for solving the Newsvendor problem with k scenarios;

# Cluster (k)	$\bar{p}_N$	s_N^2	t_R	t_S
1	3	0.16	0	0
2	18	0.40	0	0
3	34	0.26	0	0
4	45	0.10	0	0
5	57	0.08	0	0
6	57	0.08	0	0
7	57	0.08	0	0
8	57	0.08	0	0
9	57	0.08	0	0
10	57	0.08	0	0
11	57	0.08	0	0
12	57	0.08	0	0
13	57	0.08	0	0
14	57	0.08	0	0
15	57	0.08	0	0

DA TENERE SE E' VERO: These results demonstrate the effectiveness of Wasserstein reduction: although the number of scenarios is significantly decreased, the essential statistical features of the original demand distribution are maintained. The reduced scenario set still allows the optimization model to find a solution that is both robust and close to the original optimal profit.

6.2 ATO Problem

In this subsection, we report the results for the Assemble-to-Order (ATO) problem using scenario reduction via the Wasserstein distance. The original multidimensional demand scenarios were reduced to $k \in [1, 15]$ representative scenarios by solving the exact MILP formulation that minimizes the Wasserstein distance, ensuring that the probabilistic and structural characteristics of the original distribution are preserved as faithfully as possible.

The resulting reduced problems are summarized in Table6.2, where $\forall k \in [1, 15]$ the following were given:

- $\bar{p}_N$: estimate of the expected value of profit, obtained by considering the sample mean computed using the N values from solving the problem for each sample;
- s_N^2 : sample variance of the expected values of the profit;
- t_R : average computation time for reducing to k scenarios;
- t_S : average computation time for solving the Newsvendor problem with k scenarios;

# Cluster (k)	$\bar{p}_N$	s_N^2	t_R	t_S
1	3	0.16	0	0
2	18	0.40	0	0
3	34	0.26	0	0
4	45	0.10	0	0
5	57	0.08	0	0
6	57	0.08	0	0
7	57	0.08	0	0
8	57	0.08	0	0
9	57	0.08	0	0
10	57	0.08	0	0
11	57	0.08	0	0
12	57	0.08	0	0
13	57	0.08	0	0
14	57	0.08	0	0
15	57	0.08	0	0

DA TENERE SE CORRETTO: This outcome highlights the robustness and accuracy of the Wasserstein-based scenario reduction, which manages to preserve the key features of the stochastic demand while significantly reducing computational complexity.

7 Efficiency

In this section, we compare the computational efficiency of the different scenario reduction methods applied to both the Newsvendor and ATO problems. We report and discuss the execution times of each algorithm, summarizing the results through graphical visualizations.

Figure 5 shows the total computational times required for the two problems, including both scenario reduction (with K-means and Wasserstein) and the subsequent optimization with the reduced scenarios.

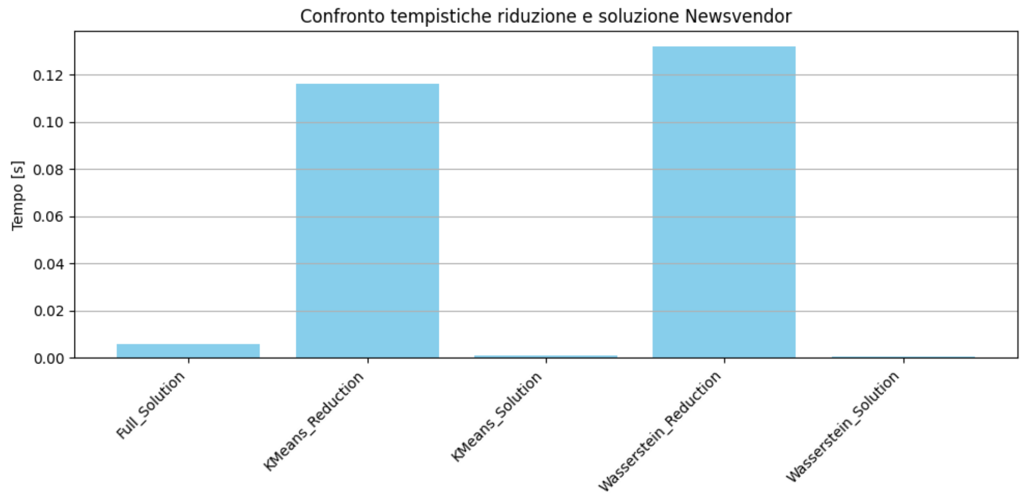

Figure 5: Total execution times for Newsvendor problem and ATO problem, for each value of $k \in [1, 15]$. $\rightarrow$ inserisci figura giusta

Problem	**Method**	**Total Time [s]**
Newsvendor	Full	0.0058
Newsvendor	K-means	0.1167
Newsvendor	Wasserstein	0.1324
ATO	Full	0.0059
ATO	K-means	0.1207
ATO	Wasserstein	0.1984

Table 3: Comparison between total execution times for Newsvendor problem and ATO problem with full scenarios, reduced scenarios via Kmeans and reduced scenarios via Wasserstein-based heuristic. $\rightarrow$ inserisci valori giusti.

8 Discussion

Modifica in base ai risultati.

From the chart, we observe that the time required to solve the full problem without reduction (`Full_Solution`) is significantly lower than the time spent in the reduction phases (`KMeans_Reduction` and `Wasserstein_Reduction`), which dominate the total computational cost. Conversely, the actual optimization on the reduced scenario sets (`KMeans_Solution` and `Wasserstein_Solution`) is almost negligible in terms of time.

This result highlights that, for the Newsvendor problem with a moderate number of scenarios, the bottleneck is the reduction procedure itself—especially when using methods such as K-means or Wasserstein MILP—while the optimization stage becomes extremely fast once the scenario set is reduced. Therefore, the choice of the reduction algorithm and its implementation has a direct impact on the overall computational efficiency of the workflow.

As with the Newsvendor case, the time required for the scenario reduction phases (`KMeans_Reduction` and `Wasserstein_Reduction`) is substantially higher than the time needed to solve the full problem without reduction (`Full_Solution`) or to solve the optimization problem on the reduced scenario sets (`KMeans_Solution` and `Wasserstein_Solution`), which are both negligible in terms of computational cost.

It is particularly evident that the Wasserstein reduction, involving the solution of a mixed-integer programming problem in a multidimensional space, is the most computationally demanding step. However, once the reduced scenario set is obtained, the final optimization becomes extremely efficient.

This analysis confirms that, for moderate instance sizes, the major computational effort is concentrated in the scenario reduction phase—especially for methods based on mathematical programming—while the actual optimization benefits greatly from working on a smaller scenario set. Thus, efficiency considerations must take into account not only the quality of the reduced scenarios, but also the time required for the reduction procedure itself.

The analyses presented in this report highlight important aspects regarding both the quality of solutions and the computational efficiency of different scenario reduction techniques when applied to the Newsvendor and Assemble-to-Order (ATO) problems.

From a solution perspective, all scenario reduction methods—K-means and Wasserstein distance—were able to preserve the essential structure of the original problems. In both cases, the optimal decisions (order quantity for Newsvendor, component production plan for ATO) and the expected profit obtained with the reduced scenario sets were virtually identical to those derived from the full set of scenarios. This demonstrates the effectiveness of both K-means and Wasserstein-based reductions in maintaining solution quality while decreasing the scenario space.

When considering computational efficiency, however, the results highlight important differences. The computational cost of the actual optimization, both for the full and reduced scenario sets, is negligible for both problems. The majority of the total computation time is instead spent on the scenarios reduction phase itself, with the Wasserstein MILP approach being the most time-consuming, especially for the multidimensional ATO case. K-means, while still significantly slower than the direct optimization, consistently requires less time than the Wasserstein-based reduction.

Comparing the two problems, the trends remain similar: the scenario reduction phase dominates the total computational time–especially as the dimensionality of the problem increases–but both reduction methods succeed in dramatically simplifying the scenario set without any meaningful loss in solution quality. In summary, the choice of the reduction technique may be guided by the available computational resources and the problem's dimensionality: K-means offers faster execution and satisfactory accuracy for most practical purposes, while the Wasserstein approach is preferable when maximum fidelity to the original probability distribution is required, at the expense of increased computation time.