http://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0899-1499, EISSN 1526-5528

Submitted to *INFORMS Journal on Computing*

A Clustering-Based Uncertainty Set for Robust Optimization

Alireza Yazdani
Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, a.yazdani.esfidvajani@tue.nl

Ahmadreza Marandi
Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, a.marandi@tue.nl
Department of Management, University of British Columbia, Canada

Rob Basten
Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, r.j.i.basten@tue.nl

Lijia Tan
Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, l.tan1@tue.nl

Abstract. Robust optimization is an approach for handling uncertainty in optimization problems, in which the uncertainty set determines the conservativeness of the solutions. In this paper, we propose a data-driven uncertainty set using a type of volume-based clustering, which we call Minimum-Volume Norm-Based Clustering (MVNBC). MVNBC extends the concept of minimum-volume ellipsoid clustering by allowing clusters to be enclosed within regions defined by a given set of vector norms, while explicitly detecting outliers at a specified rate. We formulate MVNBC as a mixed-integer conic optimization problem to optimally assign data points to clusters by minimizing the total volume of the norm-based regions. To address computational complexities, we develop both an exact decomposition-based algorithm and an efficient iterative approximation algorithm. The exact algorithm employs generalized Benders decomposition to decompose the problem into an assignment master problem and conic subproblems. The approximation algorithm, inspired by insights from generalized Benders decomposition, iteratively reassigns data points to different clusters until guaranteed convergence. Through extensive numerical experiments, we first show that MVNBC effectively captures data patterns and identifies clusters with minimum total volume. Moreover, we demonstrate that robust solutions derived from our uncertainty set are less conservative than those obtained using uncertainty sets without clustering or those obtained using selected benchmarks from the literature.

Key words: Robust Optimization, Uncertainty Sets, Clustering, Data-driven Optimization, Conic Optimization, Generalized Benders Decomposition

1. Introduction

Uncertainty can significantly affect the performance of solutions obtained from optimization problems (Ben-Tal and Nemirovski 2002). Robust Optimization (RO) is a widely-used approach for addressing uncertainty in optimization problems, aiming to identify solutions safeguarded against all scenarios within a specified uncertainty set (Ben-Tal and Nemirovski 2000).

For a convex optimization problem affected by an uncertain parameter $\mathbf{u}$, the nominal problem assumes a fixed parameter estimate $\bar{\mathbf{u}}$ and reads as

$$\inf_{\mathbf{x}\in\mathbb{R}^n} \{g(\mathbf{x}) \mid f_j(\mathbf{x},\bar{\mathbf{u}}) \le 0,\ \forall j \in \{1,\dots,m\}\}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of decision variables, $g\colon \mathbb{R}^n \to \mathbb{R}$ is the convex objective function, and $f_j(\cdot,\bar{\mathbf{u}})\colon \mathbb{R}^n \to \mathbb{R}, j \in \{1,\dots,m\}$, are closed, proper, and convex functions.

The robust counterpart of Problem (1), given that the uncertain parameter lies within the set $\mathcal{U}$ reads as

$$\inf_{\mathbf{x}\in\mathbb{R}^n} \{g(\mathbf{x}) \mid f_j(\mathbf{x},\mathbf{u}) \le 0,\ \forall \mathbf{u} \in \mathcal{U},\ \forall j \in \{1,\dots,m\}\}. \tag{2}$$

A common criticism of RO is that it can yield overly conservative solutions (Bertsimas et al. 2011). The conservativeness of the RO solutions is controlled via the uncertainty set $\mathcal{U}$, as a larger uncertainty set typically increases conservativeness, ultimately affecting the objective value (Baron et al. 2011, Kang et al. 2023). Classical uncertainty sets, such as boxes (Soyster 1973), ellipsoids (Ben-Tal and Nemirovski 1998), or polytopes (Bertsimas and Sim 2004), are commonly used due to their ease of construction, and they result in computationally tractable reformulations of Problem (2). However, as these sets are constructed solely based on empirical averages of available data points as a nominal estimate, they often lead to overly conservative solutions. To address this limitation, researchers have begun constructing uncertainty sets directly from data, giving rise to a stream of literature on data-driven uncertainty sets.

Data-driven uncertainty sets generally lead to less conservative solutions than classical uncertainty sets (Bertsimas and Thiele 2006). There are various ways to construct data-driven uncertainty sets, including methods based on statistical tests, moment-based approaches, clustering, and deep learning. We discuss these in more detail in Section 2. This paper contributes to this research stream by proposing a new clustering-based uncertainty set.

Clustering is an unsupervised learning method that groups data into distinct clusters, placing data points with similar attributes together and separating those with differing attributes (Suyal and Sharma 2024). Clustering is recognized as one of the most explainable Machine Learning (ML) methods (Moshkovitz et al. 2020) because the clusters provide a direct and interpretable structure, assigning each data point clearly to a group, often defined by geometric or statistical properties. This ML method has recently gained attention in constructing data-driven uncertainty sets, where it is used to construct uncertainty sets that balance interpretability and computational efficiency. One such method is volume-based clustering, which defines clusters based on minimal enclosing regions.

We propose a data-driven uncertainty set based on a volume-based clustering method, which we call Minimum-Volume Norm-Based Clustering (MVNBC). MVNBC generalizes Minimum Volume Ellipsoid Clustering (MVEC) by not restricting cluster shapes to predefined ellipsoids. Instead, it allows clusters to be defined through regions characterized by different vector norms, providing greater flexibility to capture data patterns. We formulate the MVNBC problem as a Mixed-Integer Non-Linear Optimization (MINLO) problem that identifies optimal clusters by minimizing their multidimensional volume while explicitly controlling the rate of outliers. To improve computational tractability, we reformulate the MINLO problem as a Mixed-Integer Conic Optimization (MICO) problem. To address the computational complexity of this MICO problem, we develop an exact decomposition-based algorithm using generalized Benders decomposition (GBD). As the exact algorithm is not computationally efficient for large-scale instances, we further develop an approximation algorithm inspired by GBD that iteratively reassigns points to clusters, monotonically decreasing clusters' total volume during this process. We conduct numerical experiments to show the performance of our method and the corresponding algorithms. The first experiment demonstrates that the MVNBC method outperforms some typical clustering methods in finding clusters with minimum volumes. The second experiment confirms the effectiveness of our MVNBC-based constructed uncertainty sets in RO for a 2-product newsvendor problem. The third experiment shows that our uncertainty set provides about 10% better objective value than one of the recent data-driven uncertainty sets in the literature, and highlights the potential for our data-driven uncertainty set to be applied to a wide range of RO problems.

The remainder of the paper is organized as follows. In Section 2, we review relevant literature on data-driven uncertainty sets, minimum volume-based clustering approaches, and highlight our contributions. In Section 3, we outline the notation and provide essential mathematical preliminaries used throughout. In Section 4, we introduce the norm-based clustering optimization problem and present its MICO reformulation. In Section 5, we develop an exact solution algorithm for solving the proposed MICO formulation. In Section 6, we describe an efficient approximation algorithm designed to overcome the computational limitations of the exact algorithm. In Section 7, we provide the robust optimization reformulation of a convex optimization problem using the presented uncertainty set. We present numerical experiments demonstrating the effectiveness of our clustering method and its application in constructing uncertainty sets for RO in Section 8. Finally, in Section 9, we summarize our findings and suggest potential directions for future research.

2. Literature

In this section, we first review data-driven uncertainty sets in Section 2.1, covering classical approaches and ML-based methods. Next, we discuss minimum volume-based clustering methods and their limitations in Section 2.2. Finally, in Section 2.3, we outline our contributions that address these gaps through Minimum Volume Norm-Based Clustering (MVNBC).

2.1. Data-driven uncertainty sets

There are two major research streams on constructing data-driven uncertainty sets: those that construct classical uncertainty sets and ML-based ones. Classical uncertainty sets include shapes such as boxes, ellipsoids, and polytopes, and they rely on statistical assumptions or distributional knowledge about data. A seminal contribution in this stream is Bertsimas et al. (2018), which introduced uncertainty sets integrating statistical hypothesis testing and goodness-of-fit measures with prior assumptions on data distributions. Additional classical approaches, such as those surveyed by Gabrel et al. (2014) and Gorissen et al. (2015), focus on defining uncertainty sets using statistical confidence intervals and prior assumptions on distributions. Within this line of work, Zhang et al. (2017) proposed specific enhancements by adaptively adjusting classical box, ellipsoidal, and polyhedral uncertainty sets via data-driven confidence intervals, making uncertainty sets more responsive to data patterns. Despite their computational efficiency and interpretability, classical methods often yield overly conservative solutions when actual data patterns deviate significantly from their predefined convex shapes.

The second stream, ML-based methods, relies on learning techniques to capture complex data patterns without being constrained to predefined structures. One of the first such methods is kernel-based Support Vector Clustering (SVC) to construct uncertainty sets (Shang et al. 2017). A modification of SVC, proposed by Asgari et al. (2024), introduced position-regulated SVC to refine the uncertainty set structure. However, both methods suffer from scalability issues and struggle to represent multiple data patterns effectively. To address these limitations, Goerigk and Kurtz (2023) proposed deep Neural Networks (NN) to construct uncertainty sets, showing superior performance over SVC in detecting data patterns and identifying outliers. However, the resulting RO counterparts derived from NN-based uncertainty sets tend to be computationally expensive.

To enhance computational tractability, several studies integrated dimension reduction techniques into the construction of uncertainty sets. Ning and You (2018) utilized Principal Component Analysis (PCA) combined with Kernel Density Estimation (KDE) to construct uncertainty sets based on the principal components of the data distribution. Zhang et al. (2022) employed PCA with a cutting-plane method to construct polyhedral uncertainty sets. However, PCA-based methods often encounter the curse of dimensionality, and the cutting-plane methods may include unlikely scenarios within the uncertainty sets.

Recent studies integrate learning techniques into optimization frameworks, aiming to balance computational efficiency with robust decision-making. Wang et al. (2023) introduced a data-driven method that reshapes uncertainty sets based on observed performance across a parametric family of problems, ensuring the satisfaction of constraints. To mitigate computational challenges, Goerigk and Khosravi (2023) developed scenario reduction techniques that select the centers of clustering of K-means as probable scenarios, improving the solvability in RO problems. Loger et al. (2024) introduced an approximate kernel learning method to define uncertainty sets, offering a flexible and computationally efficient alternative.

Within this stream, clustering techniques have emerged prominently, particularly K-means clustering, for constructing data-driven uncertainty sets. De Rosa et al. (2024) analyzed the effectiveness of linear programming relaxations in K-means clustering, offering insights into the integration of clustering techniques in RO frameworks. Wang et al. (2024) proposed mean robust optimization, which utilizes K-means to partition data and define uncertainty sets that balance computational efficiency and conservatism. Neofytou et al. (2025) introduced a method that combines clustering with a Dirichlet process mixture model to construct uncertainty sets as the intersection of ℓ_1 and ℓ_∞ norms. Li et al. (2025) constructed the uncertainty set using a monolithic mixed-integer formulation as a union of multiple subsets to improve computational efficiency in predictive control applications.

In summary, clustering-based uncertainty sets effectively balance interpretability, robustness, and computational efficiency. However, existing clustering-based uncertainty sets typically rely on a fixed predefined geometric shape, such as the ellipsoid, limiting their flexibility in capturing complex data patterns. This limitation motivates the development of uncertainty sets that allow for more general shapes, enhancing their adaptability to various data distributions while maintaining computational tractability.

2.2. Minimum volume-based clustering

Clustering methods can broadly be classified into partitioning, hierarchical, and density-based methods (Han et al. 2012, pp. 383-385). Partitioning methods, such as K-means, segment datasets into K disjoint clusters, assigning each point uniquely to one group. These methods typically minimize an objective function, often related to distances between points and cluster centroids (Han et al. 2012, pp. 385-390). Hierarchical clustering groups data points into nested clusters, forming a tree-like structure by progressively merging or splitting clusters based on their similarity (Nielsen 2016). Density-based clustering identifies clusters as regions of high data density, separated by areas of lower density, effectively handling clusters of arbitrary shapes (Campello et al. 2020).

Among density-based clustering methods, volume-based clustering stands out due to its suitability for constructing uncertainty sets in robust optimization. Specifically, volume-based clustering directly aligns with the objective of minimizing uncertainty region volumes, thereby reducing conservativeness in optimization solutions. One classical method, known as Minimum Volume Ellipsoid Clustering (MVEC), defines clusters using the smallest volume ellipsoid enclosing all points in a cluster. Rosen (1965) formulated the problem of finding a single minimum-volume ellipsoid enclosing a given set of points as a convex optimization problem for pattern separation. Barnes (1982) later developed an algorithm to compute such an ellipsoid.

Computing minimum-volume ellipsoids involves solving optimization problems that minimize objective functions defined by determinants of matrices. Several algorithms have been proposed to efficiently approximate solutions, including heuristic algorithms such as simulated annealing, genetic algorithms, and tabu search (Woodruff and Rocke 1993), arithmetic-based approximations (Khachiyan 1996), iterative procedures (Candela 1996), convex optimization algorithms (Martínez-Rego et al. 2013), re-sampling algorithms (Van Aelst and Rousseeuw 2009), and randomized algorithms (Ahipaşaoğlu 2015).

Shioda and Tunçel (2007) introduced Minimum Volume Ellipsoid Clustering (MVEC), formulating it as a mixed-integer semidefinite optimization problem that minimizes the geometric mean of the volumes of ellipsoids covering the clusters. Their method achieves scale invariance and effectively handles asymmetric and unequal-sized clusters, though it does not explicitly address outlier detection.

Extending MVEC, Kumar and Orlin (2008) proposed a mixed-integer nonconvex optimization formulation capable of simultaneously identifying clusters and detecting outliers. Similarly, Martínez-Rego et al. (2013), aiming to improve computational tractability, developed a nonconvex formulation that minimizes the sum of squared ellipsoid volumes but does not explicitly incorporate outlier detection.

Building upon these works on MVEC, we introduce a clustering-based methodology specifically tailored to construct data-driven uncertainty sets. While MVEC and related methods predominantly utilize ellipsoidal shapes, our method generalizes uncertainty regions by allowing clusters to be defined through various norm-based shapes. This enhanced flexibility provides a more accurate representation of diverse data distributions compared to purely ellipsoidal methods, and enables decision-makers to formulate RO problems that remain computationally tractable. By explicitly incorporating outlier detection, our uncertainty set further balances robustness and conservativeness effectively in data-driven RO.

2.3. Our Contribution

In this paper, we provide a threefold contribution to the literature:

First, we introduce *Minimum Volume Norm-Based Clustering (MVNBC)*, a generalization of Minimum Volume Ellipsoid Clustering (MVEC). MVNBC formulates clustering as an optimization problem that incorporates a set of vector norms in $\mathbb{R}^d$ and explicitly accounts for outliers. To enhance computational tractability, we reformulate this problem as a Mixed-Integer Conic Optimization (MICO) problem, leveraging structured conic constraints, including an *exponential cone* and a *logdet cone* for each cluster, as well as either a *quadratic cone* or a *linear cone* for each data point-cluster pair.

Second, we develop two solution algorithms tailored to address the computational complexity associated with solving the MVNBC problem. First, we propose an *exact algorithm* based on *generalized Benders decomposition*, which decomposes the original problem into an assignment master problem and several conic subproblems, enhancing computational tractability. Recognizing the limitations of the exact algorithm for large-scale datasets, we next propose an *approximation algorithm*. This algorithm iteratively reassigns data points to clusters while progressively reducing the total volume, thereby substantially improving scalability and computational performance for larger problem instances.

Third, we conduct *comprehensive numerical experiments* to evaluate MVNBC's performance. Initially, we compare MVNBC against K-means and Gaussian Mixture Models (GMM) in terms of capturing underlying data patterns. Our results demonstrate that MVNBC provides superior pattern recognition. Subsequently, we leverage MVNBC to construct *data-driven uncertainty sets* tailored for RO. By varying the number of clusters and region shapes, we find configurations that significantly improve the quality of robust solutions when applied to a newsvendor problem. Finally, our experiments show that the MVNBC-based uncertainty sets achieve up to 10% improvement in the objective value, while considerably reducing computation times, compared to recent uncertainty sets proposed by Goerigk and Kurtz (2023).

3. Preliminaries

In this section, we introduce the notation, define essential mathematical concepts used throughout the paper, and review relevant results from the literature on calculating the volume of regions.

Throughout the paper, the set of real numbers is denoted by $\mathbb{R}$, while the set of strictly positive integers is denoted by $\mathbb{N}$. Other general sets are denoted by calligraphic fonts, e.g., $\mathcal{U}$. Matrices are represented by uppercase bold letters, such as $\mathbf{A}$ or $\mathbf{B}$, and vectors by lowercase bold letters, such as $\mathbf{x}$ or $\mathbf{y}$. For a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$, its transpose is denoted by $\mathbf{A}^T \in \mathbb{R}^{m\times n}$. For any positive integer n, we define the set $[n] := \{1, 2, \ldots, n\}$.

We define the p-norm (or ℓ_p-norm) of a vector $\mathbf{x} \in \mathbb{R}^d$ as $\|\mathbf{x}\|_p = \left(\sum_{i\in[d]} |x_i|^p\right)^{1/p}$, where $|\cdot|$ denotes the absolute value. In particular, we use the following norms:

- The ℓ_1-norm: $\|\mathbf{x}\|_1 = \sum_{i\in[d]} |x_i|$
- The ℓ_2-norm or Euclidean norm: $\|\mathbf{x}\|_2 = \sqrt{\sum_{i\in[d]} |x_i|^2} = \sqrt{\mathbf{x}^T\mathbf{x}}$
- The ℓ_∞-norm: $\|\mathbf{x}\|_\infty = \max_{i\in[d]} |x_i|$

For a $d\times d$ matrix $\mathbf{A}$, the induced ℓ_p matrix norm is defined as

$$\|\mathbf{A}\|_p = \sup\left\{\|\mathbf{A}\mathbf{x}\|_p \mid \|\mathbf{x}\|_p = 1\right\}.$$

The determinant of a symmetric matrix $\mathbf{A}$ is denoted by $\det(\mathbf{A})$. We use $\mathbf{A} \succeq 0$ and $\mathbf{A} \succ 0$ to denote that the matrix $\mathbf{A}$ is positive semidefinite and positive definite, respectively.

Given $\mathbf{T} \in \mathbb{R}^{d\times d}$, the *matrix transformation* associated to $\mathbf{T}$ is the transformation $\mathrm{F} : \mathbb{R}^d \to \mathbb{R}^d$ defined by $\mathrm{F}(\mathbf{x}) = \mathbf{T}\mathbf{x}$. For a vector $\mathbf{t} \in \mathbb{R}^d$, the *vector translation* associated to $\mathbf{t}$ is the translation $\mathrm{G} : \mathbb{R}^d \to \mathbb{R}^d$ defined by $\mathrm{G}(\mathbf{x}) = \mathbf{x} - \mathbf{t}$.

A *norm-based region* is defined as $\mathcal{S} = \left\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{T}(\mathbf{x}-\mathbf{t})\|_p \leq 1\right\}$, which is compact, where $\mathbf{T} \in \mathbb{R}^{d\times d}$ is an invertible transformation matrix and $\mathbf{t} \in \mathbb{R}^d$ is a translation vector. We denote by $\mathcal{B}^p$ the unit ball of the ℓ_p-norm, defined as $\mathcal{B}^p = \left\{\mathbf{x} \mid \|\mathbf{x}\|_p \leq 1\right\}$.

Next, we recall some definitions and theorems from the literature that are essential for this paper, specifically focusing on the relationship between linear transformations and the volumes of a given norm-based region.

Definition 1 (Volume of a Region, Bemporad et al. 2004). Let $\mathcal{S} \subset \mathbb{R}^d$ be a region. The *volume of* $\mathcal{S}$ is defined as $\mathrm{vol}(\mathcal{S}) = \int_{\mathcal{S}} \mathrm{d}x$, representing the Lebesgue measure of $\mathcal{S}$.

THEOREM 1 **(Volume Scaling under Linear Transformations, Kumar and Orlin 2008).** *Let $\mathbf{T}$ be a $d \times d$ matrix and $\mathrm{T}(\mathbf{x}) = \mathbf{T}\mathbf{x}$ be the associated matrix transformation. For any region $\mathcal{S} \subseteq \mathbb{R}^d$, it holds that*

$$\mathrm{vol}(\mathrm{T}(\mathcal{S})) = |\det(\mathbf{T})| \cdot \mathrm{vol}(\mathcal{S}).$$

Proof See Kumar and Orlin (2008) (Lemma 1) for the special case where $\mathcal{S}$ is an ellipsoid, and Appendix A.1 for the general case. □

COROLLARY 1 **(Volume of a Norm-Based Region).** *Let $\mathbf{T} \in \mathbb{R}^{d\times d}$ be a positive definite symmetric matrix and $\mathbf{t} \in \mathbb{R}^d$. For the region $\mathcal{S} = \left\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{T}(\mathbf{x}-\mathbf{t})\|_p \le 1\right\}$, the volume is given by*

$$\mathrm{vol}(\mathcal{S}) = \frac{(2\Gamma(1+\frac{1}{p}))^d}{\Gamma(1+\frac{d}{p})} \cdot \frac{1}{\det(\mathbf{T})}, \tag{3}$$

where $\Gamma(\cdot)$ denotes the gamma function, defined as

$$\Gamma(x) = \int_0^\infty s^{x-1} e^{-s} ds.$$

Proof See Appendix A.2. □

For more details regarding volumes of unit balls and the calculation of the gamma function, we refer the reader to Wang (2005).

4. Minimum volume norm-based uncertainty set

In this section, we formulate the problem of constructing a data-driven uncertainty set using volume-based clustering based on the available data. First, we provide the MVNBC formulation in Section 4.1 and then reformulate it as an MICO problem in Section 4.2.

4.1. Minimum Volume Norm-based Clustering

The volume-based clustering optimization problem is similar to an assignment problem with a non-linear objective function. It aims to find the optimal assignment of N different points, $\mathbf{a}^i$, $i \in [N]$, to multiple clusters, allowing some of them to be outliers. The objective function is to minimize the summation of the volumes of the norm-based regions. Similar to the formulation by Kumar and Orlin (2008) and Martínez-Rego et al. (2013) for the ℓ_2-norm, we formulate the problem for a general norm region as

$$\begin{aligned}
\inf_{\mathcal{S}_{k_p}, w^i, l^i_{k_p}} \quad & \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \mathrm{vol}(\mathcal{S}_{k_p}) && && \text{(4a)}\\
\text{subject to} \quad & \mathbf{a}^i \in \mathcal{S}_{k_p} \text{ if } l^i_{k_p} = 1, && \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, && \text{(4b)}\\
& w^i \le \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} l^i_{k_p} \le w^i \sum_{p\in\mathcal{P}} K_p, && \forall i \in [N], && \text{(4c)}\\
& \sum_{i\in[N]} w^i \ge (1-r)N, && && \text{(4d)}\\
& w^i \in \{0,1\}, && \forall i \in [N], && \text{(4e)}\\
& l^i_{k_p} \in \{0,1\}, && \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, && \text{(4f)}
\end{aligned}$$

where $\mathcal{P}$ is the desired set of norms, K_p is the number of ℓ_p-norm-based regions, for each $p \in \mathcal{P}$, $\mathcal{S}_{k_p}$ is the norm-based region defined as $\mathcal{S}_{k_p} := \{\mathbf{x} \in \mathbb{R}^d \mid \left\|\mathbf{T}_{k_p}(\mathbf{x} - \mathbf{t}_{k_p})\right\|_p \leq 1\}$ for all $p \in \mathcal{P}$ and $k_p \in [K_p]$, where $\mathbf{T}_{k_p} \in \mathbb{R}^{d\times d}$ is a positive definite transformation matrix and $\mathbf{t}_{k_p} \in \mathbb{R}^d$ is the translation vector, and $l^i_{k_p}$ is the membership indicator, a binary variable that equals 1 if the data point $\mathbf{a}^i$ is assigned to region $\mathcal{S}_{k_p}$, and 0 otherwise.

Problem (4) aims to minimize the total volume of regions. Constraint (4b) assigns the points to clusters, Constraint (4c) ensures that non-outlier data points are assigned to at least one cluster. Constraint (4d) ensures that at most r portion of the data points are treated as outliers. To define the assignment indicators, w^i, and the membership indicators, $l^i_{k_p}$, we use binary values, which are enforced by Constraints (4e)-(4f).

Our formulation is originally devised for minimizing Objective (4a), representing the aggregated volume of the clusters. It can, however, easily be revised to minimize the geometric mean, which is also considered in the literature. We emphasize that Shioda and Tunçel (2007) formulate the Minimum Volume Ellipsoid Clustering (MVEC) as a mixed-integer semidefinite optimization problem with $\mathcal{P} = \{2\}$, minimizing the geometric mean of the cluster volumes, expressed as $\sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \log(\mathrm{vol}(\mathcal{S}_{k_p}))$. Our formulation generalizes this by detecting outliers while considering regions defined by general vector norms rather than restricting them solely to ellipsoids. Thus, our model achieves greater flexibility while ensuring the tractability of the resulting robust counterpart. As we show in our numerical experiments, this flexibility provides computational advantages for the reformulation of the robust counterpart.

To improve tractability of (4), we now reformulate the problem as a Mixed-Integer Conic Optimization (MICO) problem.

4.2. Minimum Volume Norm-based Clustering Conic Formulation

This section provides an MICO problem to find the clusters with minimum volume. Clearly, Problem (4) is equivalent to the following optimization problem, where Constraint (4b) is reformulated to Constraints (5b)-(5c)

$$\begin{aligned}
\inf_{\mathcal{S}_{k_p}, l^i_{k_p}, \mathbf{T}_{k_p}, \mathbf{t}_{k_p}, w^i} \quad & \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \mathrm{vol}(\mathcal{S}_{k_p}) && && \text{(5a)}\\
\text{subject to} \quad & l^i_{k_p} \left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p \leq 1, && \forall i \in [N],\ k_p \in [K_p],\ p \in \mathcal{P}, && \text{(5b)}\\
& \mathbf{T}_{k_p} \succ 0, && \forall k_p \in [K_p],\ p \in \mathcal{P}, && \text{(5c)}\\
& \text{(4c)}, \ldots, \text{(4f)}.
\end{aligned}$$

Problem (5) is a mixed-integer non-convex optimization problem, which is computationally intractable. In the following theorem, we introduce a reformulation for this particular problem.

THEOREM 2 **(Conic Refomulation of MVNBC).** *The optimization problem (5) is equivalent to*

$$\begin{aligned}
\inf_{\theta_{k_p}, l^i_{k_p}, \mathbf{T}_{k_p}, \mathbf{t}_{k_p}, w^i} \quad & \sum_{p\in\mathcal{P}} \frac{(2\Gamma(1+\frac{1}{p}))^d}{\Gamma(1+\frac{d}{p})} \cdot \sum_{k_p\in[K_p]} \theta_{k_p} && && \text{(6a)}\\
\text{subject to} \quad & -\log\theta_{k_p} - \log\det(\mathbf{T}_{k_p}) \le 0, && \forall k_p \in [K_p],\ p\in\mathcal{P}, && \text{(6b)}\\
& \mathbf{T}_{k_p} \succ 0, && \forall k_p \in [K_p],\ p\in\mathcal{P}, && \text{(6c)}\\
& \theta_{k_p} > 0, && \forall k_p \in [K_p],\ p\in\mathcal{P}, && \text{(6d)}\\
& l^i_{k_p} \left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p \le 1, && \forall i\in[N],\ k_p \in [K_p],\ p\in\mathcal{P}, && \text{(6e)}\\
& (4c),\ldots,(4f).
\end{aligned}$$

Proof Since $\mathcal{S}_{k_p} := \{\mathbf{x}\in\mathbb{R}^d \mid \left\|\mathbf{T}_{k_p}(\mathbf{x}-\mathbf{t}_{k_p})\right\|_p \le 1\}$ for all desired norm $p\in\mathcal{P}$ and all $k_p\in[K_p]$, using Corollay 1, we rewrite Objective (5a) as

$$\begin{aligned}
\sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \mathrm{vol}(\mathcal{S}_{k_p}) &= \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} \frac{1}{\det(\mathbf{T}_{k_p})} \cdot \mathrm{vol}(\mathcal{B}^p)\\
&= \sum_{p\in\mathcal{P}} \mathrm{vol}(\mathcal{B}^p) \sum_{k_p\in[K_p]} \frac{1}{\det(\mathbf{T}_{k_p})}.
\end{aligned}$$

Let θ_{k_p} be a new variable for each region $k_p\in[K_p]$. Using epigraph reformulation, Problem (5) reads as

$$\begin{aligned}
\inf_{\theta_{k_p}, l^i_{k_p}, \mathbf{T}_{k_p}, \mathbf{t}_{k_p}, w^i} \quad & \sum_{p\in\mathcal{P}} \mathrm{vol}(\mathcal{B}^p) \sum_{k_p\in[K_p]} \theta_{k_p} && && \text{(7a)}\\
\text{subject to} \quad & \frac{1}{\det(\mathbf{T}_{k_p})} \le \theta_{k_p}, && \forall k_p \in [K_p],\ p\in\mathcal{P}, && \text{(7b)}\\
& \mathbf{T}_{k_p} \succ 0, && \forall k_p \in [K_p],\ p\in\mathcal{P}, && \text{(7c)}\\
& \theta_{k_p} > 0, && \forall k_p \in [K_p],\ p\in\mathcal{P}, && \text{(7d)}\\
& l^i_{k_p} \left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p \le 1, && \forall i\in[N],\ k_p \in [K_p],\ p\in\mathcal{P}, && \text{(7e)}\\
& (4c),\ldots,(4f).
\end{aligned}$$

Since $\log(\cdot)$ is a strictly increasing function, we replace Constraint (7b) with

$$-\log\det(\mathbf{T}_{k_p}) \le \log\theta_{k_p}, \quad \forall k_p\in[K_p],\ p\in\mathcal{P}. \tag{8}$$

□

Theorem 2 shows how an optimal clustering is obtained by reformulating the $\det(\cdot)$ function into a conic form. More explicitly, introducing new variable τ_{k_p}, Constraint (6b) can be further reformulated as

$$\begin{aligned}
-\log\det(\mathbf{T}_{k_p}) &\le \tau_{k_p}, \quad \forall k_p\in[K_p],\ p\in\mathcal{P}, && \text{(9a)}\\
\tau_{k_p} &\le \log\theta_{k_p}, \quad \forall k_p\in[K_p],\ p\in\mathcal{P}. && \text{(9b)}
\end{aligned}$$

Thus, given $\mathcal{P} \subseteq \{1, 2, +\infty\}$, Constraint (9a) is equivalent to $(-\tau_{k_p}, 1, \mathbf{T}_{k_p})$ belonging to the *logdet* cone, and Constraint (9b) is equivalent to $(\theta_{k_p}, 1, \tau_{k_p})$ belonging to the *exponential* cone.

Despite having the cones in Problem (6), the problem is not yet an MICO problem due to the products of $l^i_{k_p}$ and $\left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p$ in Constraint (6e). We know that if $\left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p$ is bounded from above, then the product has a mixed-integer convex quadratic reformulation, due to $l^i_{k_p}$ being binary. However, $\left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p$ may not necessarily be bounded. To see this, let $l^i_{k_p}$ be 1 for only one i and k_p. Let $\mathbf{T}_{k_p}$ be a diagonal matrix where all the diagonal elements are $\chi > 0$, and $\mathbf{t}_{k_p} = \mathbf{a}^i$. Therefore, Constraint (6e) holds for this solution. Furthermore, increasing χ decreases the objective function, implying the norm is unbounded. Even though $\left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p$ cannot be bounded from above, in the next theorem, we provide a necessary condition on an optimal clustering, helping us to bound $\left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p$.

THEOREM 3 **(Bounding $\left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p$ for optimal clustering).** *Let $0 < \omega < \frac{1}{2}$ be given. Also, let $\left(\theta^*_{k_p}, l^{i*}_{k_p}, \mathbf{T}^*_{k_p}, \mathbf{t}^*_{k_p}, w^{i*}\right)$ be an optimal solution of Problem (6). For any $p \in \mathcal{P}$ and $k_p \in [K_p]$, if $\theta^*_{k_p} > 0$, then $\left\|{\mathbf{T}^*_{k_p}}^{-1}\right\|_p > \gamma$, where*

$$\gamma := \omega \cdot \min_{p \in \mathcal{P}} \min_{\substack{i,j \in [N] \\ i \neq j}} \left\|a^i - a^j\right\|_p .$$

Proof Suppose, for contradiction, that there exist $p \in \mathcal{P}$ and $k_p \in [K_p]$ such that $\theta^*_{k_p} > 0$ and $\left\|{\mathbf{T}^*_{k_p}}^{-1}\right\|_p \leq \gamma$. Let us denote by $\mathcal{B}^p_\kappa(\mathbf{a})$ the ball $\{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x} - \mathbf{a}\|_p \leq \kappa\}$. Given this notation, we first show that for any $\mathbf{a} \in \mathbb{R}^d$, the ball $\mathcal{B}^p_\gamma(\mathbf{a})$ contains at most one data point. To see this, let us assume the contrary and that there are two points i and j in the ball. So, we have

$$\|\mathbf{a}^i - \mathbf{a}\|_p \leq \gamma, \quad \|\mathbf{a}^j - \mathbf{a}\|_p \leq \gamma.$$

Therefore,

$$\|\mathbf{a}^i - \mathbf{a}^j\|_p \leq \|\mathbf{a}^i - \mathbf{a}\|_p + \|\mathbf{a} - \mathbf{a}^j\|_p \leq 2\gamma < \|\mathbf{a}^i - \mathbf{a}^j\|_p,$$

which is a contradiction, where (from left) the first inequality is the triangle inequality, the second one is because both i-th and j-th data points are in the ball, and the last one is because of the definition of γ.

We now show that the region $\left\|\mathbf{T}^*_{k_p}(\mathbf{x} - \mathbf{t}^*_{k_p})\right\|_p \leq 1$ is contained in the ball $\mathcal{B}^p_\gamma(\mathbf{t}^*_{k_p})$. We know

$$\sup_{\substack{\mathbf{x} \in \mathbb{R}^d \\ \left\|\mathbf{T}^*_{k_p}(\mathbf{x} - \mathbf{t}^*_{k_p})\right\|_p \leq 1}} \left\|\mathbf{x} - \mathbf{t}^*_{k_p}\right\|_p = \sup_{\|\mathbf{y}\| \leq 1} \left\|{\mathbf{T}^*_{k_p}}^{-1}\mathbf{y}\right\|_p \leq \left\|{\mathbf{T}^*_{k_p}}^{-1}\right\|_p \leq \gamma,$$

which shows that the cluster is contained in the ball $\mathcal{B}^p_\gamma(\mathbf{t}^*_{k_p})$. Therefore, the cluster contains at most one data point since the ball has this feature. If the cluster does not contain any data point, then clearly its volume, and hence $\theta^*_{k_p}$, is 0, which contradicts the assumption. So, let $\mathbf{a}^i$ be the point in the cluster. Then, setting $\mathbf{T}_{k_p}$ to be a diagonal matrix where all diagonal elements are $\chi > 0$, and $\mathbf{t}_{k_p} = \mathbf{a}^i$, we have the same cluster of data, which is a singleton, with $\theta_p = \frac{1}{\chi^d}$. Since increasing χ does not change the cluster while reducing the objective function, we have $\theta^*_{k_p} = 0$. This contradicts the assumption, which concludes the proof. □

Theorem 3 asserts that for clusters with positive volume, we have a lower bound on the norm of the inverse of the transformation matrix. Since for any induced matrix norm $\|\cdot\|_p$, we have $\frac{1}{\|\mathbf{A}\|_p} \leq \|\mathbf{A}^{-1}\|_p$, we conclude that we are looking for transformation matrices $\mathbf{T}_{k_p}$, whose norm does not exceed $\frac{1}{\gamma}$. Moreover, without loss of generality, we can assume that $\|\mathbf{a}^i - \mathbf{t}_{k_p}\|_p < 1$. Therefore, we limit ourselves to the feasible points where $\left\|\mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p})\right\|_p$ is bounded from above by $\frac{1}{\gamma}$. Therefore, Problem (6) has an outer approximation (enlarging the feasible region) that is an MICO problem.

We emphasize that this new approximation and Problem (6) extends the mixed-integer nonconvex optimization problem (MINOP) introduced by Shioda and Tunçel (2007), which was limited to $p = 2$.

While there are solvers that can tackle MICO problems, our formulation present computational challenges due to the large number of cones, which grows with the number of clusters, and the increasing number of constraints, which depend on the number of data points. Consequently, in the next section, we present an exact solution algorithm based on GBD to solve this problem efficiently.

5. Exact Solution Algorithm

In this section, we present an exact solution algorithm based on generalized Benders decomposition (GBD) (Geoffrion 1972) for solving the Minimum Volume Norm-Based Clustering (MVNBC) problem formulated in Problem (6).

GBD is an optimization technique that decomposes complex optimization problems into a simpler master problem and a set of manageable subproblems, significantly enhancing computational efficiency. It is highly effective for handling Mixed-Integer Nonlinear Optimization (MINLO) problems. Specifically, the GBD framework identifies complicating variables that, when fixed, reduce the complexity of the remaining optimization problem (Geoffrion 1972).

The computational difficulties of the Minimum Volume Norm-Based Clustering (MVNBC) problem stem from both conic constraints and binary variables. We identify the binary assignment variables (l, w) as the complicating variables. The master problem assigns data points to clusters, providing binary decisions that inform the subsequent solution of the subproblems. Each subproblem, given these assignments, computes the minimum-volume norm-based region enclosing the assigned points. By iteratively solving these subproblems, we derive valid generalized Benders cuts that progressively refine the feasible region of the master problem until convergence.

In Section 5.1, we present the decomposition formulations for the master problem and the subproblems of Problem (6). Section 5.2 details the derivation of GBD cuts and a linear subgradient for them. Finally, in Section 5.3, we discuss the feasibility and convergence of the algorithm.

5.1. Decomposition Reformulation

Isolating (l, w) as complicating variables in the master problem, by removing Constraints (6b)–(6e), the MVNBC problem reduces to a Mixed-Integer Linear Optimization (MILO) assignment problem

$$\inf_{\substack{\phi_{k_p}\geq 0\\ l^i_{k_p}, w^i}} \quad \sum_{p\in\mathcal{P}} \frac{(2\Gamma(1+\frac{1}{p}))^d}{\Gamma(1+\frac{d}{p})} \cdot \sum_{k_p\in[K_p]} \phi_{k_p} \tag{10a}$$

$$\text{subject to} \quad w^i \leq \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} l^i_{k_p} \leq w^i \sum_{p\in\mathcal{P}} K_p, \qquad \forall i\in[N], \tag{10b}$$

$$\sum_{i\in[N]} w^i \geq (1-r)N, \tag{10c}$$

$$\phi_{k_p} \geq \theta_{k_p}(l), \qquad \forall i\in[N],\ k_p\in[K_p], \tag{10d}$$

$$w^i \in \{0,1\}, \qquad \forall i\in[N], \tag{10e}$$

$$l^i_{k_p} \in \{0,1\}, \qquad \forall i\in[N],\ k_p\in[K_p],\ p\in\mathcal{P}, \tag{10f}$$

where the convex function $\theta_{k_p}(l)$ in Constraint (10d) gives the minimum volume of norm-based region for any given l that is feasible according to other constraints.

As $\phi_{k_p}\geq 0$, removing Constraint (10d) results in the objective value of 0, and Problem (10) serves as a feasibility problem. For any feasible l, $\theta_{k_p}(l)$ acts as a lower bound for ϕ_{k_p}. Each $\theta_{k_p}(l)$ is the objective function of a conic subproblem to find the minimum-volume norm-based region given current binary assignments from the master problem. These subproblems can be solved independently for each cluster. To efficiently formulate the subproblems, for a given $\bar{l}$, we define $\mathcal{C}_{k_p} =: \{i\in[N] \mid \bar{l}^i_{k_p} > 0\}$ as the set of points assigned to cluster k_p, so for $k_p\in[K_p]$ and $p\in\mathcal{P}$, the subproblem is

$$\theta_{k_p}(\bar{l}) = \inf_{\substack{\tau_{k_p}\geq 0\\ \theta_{k_p}, \mathbf{T}_{k_p}, \mathbf{t}_{k_p}}} \quad \theta_{k_p} \tag{11a}$$

$$\text{subject to} \quad \tau_{k_p} \leq \log\theta_{k_p}, \tag{11b}$$

$$-\tau_{k_p} \leq \log\det(\mathbf{T}_{k_p}), \tag{11c}$$

$$\bar{l}^i_{k_p} \left\| \mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p}) \right\|_p \leq 1, \qquad \forall i\in\mathcal{C}_{k_p}, \tag{11d}$$

$$\mathbf{T}_{k_p} \succ 0, \qquad \forall k_p\in[K_p],\ p\in\mathcal{P}, \tag{11e}$$

$$\theta_{k_p} > 0, \qquad \forall k_p\in[K_p],\ p\in\mathcal{P}, \tag{11f}$$

which can be solved efficiently using a conic solver like MOSEK (MOSEK ApS 2025).

5.2. Cut Generation and Subgradients

Each solution of the subproblem provides dual variables used to derive generalized Benders cuts, which serve as linear outer approximations of the convex function $\theta_{k_p}(l)$, improving the master problem's accuracy and guiding its search toward optimality. Specifically, these cuts are constructed as follows:

$$\theta_{k_p}(l) \geq \theta_{k_p}(\bar{l}) + \sum_{i\in[N]} {s^*}^i_{k_p}(l^i_{k_p} - \bar{l}^i_{k_p}), \tag{12}$$

where ${s^*}^i_{k_p}$ is a subgradient of $\theta_{k_p}(\cdot)$ at $\bar{l}$.

Let $\mathbf{T}^*_{k_p}$ and $\mathbf{t}^*_{k_p}$ denote the optimal solution of the Subproblem (11) given $\bar{l}^i_{k_p}$, and let ${u^*}^i_{k_p}$ denote the optimal dual variable associated with the Constraint (11d). By leveraging Lagrangian duality, the Karush-Kuhn-Tucker (KKT) conditions, and as Slater's conditions hold for Problem (11), we can compute a subgradient as (see Fischetti et al. (2017))

$$\begin{aligned} {s^*}^i_{k_p} &= \frac{\partial \det(\mathbf{T}^*_{k_p})}{\partial l} + {u^*}^i_{k_p} \frac{\partial \left(l^i_{k_p} \left\| \mathbf{T}^*_{k_p}(\mathbf{a}^i - \mathbf{t}^*_{k_p}) \right\|_p \right)}{\partial l} \\ &= 0 + {u^*}^i_{k_p} \left\| \mathbf{T}^*_{k_p}(\mathbf{a}^i - \mathbf{t}^*_{k_p}) \right\|_p \\ &= {u^*}^i_{k_p} \left\| \mathbf{T}^*_{k_p}(\mathbf{a}^i - \mathbf{t}^*_{k_p}) \right\|_p . \end{aligned}$$

Substituting ${s^*}^i_{k_p}$ in Problem (12), the generalized Benders cuts can be explicitly written as

$$\phi_{k_p} \geq \theta_{k_p}(\bar{l}) + \sum_{i \in [N]} {u^*}^i_{k_p} \left\| \mathbf{T}^*_{k_p}(\mathbf{a}^i - \mathbf{t}^*_{k_p}) \right\|_p (l^i_{k_p} - \bar{l}^i_{k_p}). \tag{13}$$

For a detailed description of computational aspects, including solver implementation and practical refinements to improve efficiency, we refer the reader to Appendix B.

5.3. Feasibility and Convergence

In each iteration of the algorithm, if the number of data points assigned to a cluster is too low, the volume of the corresponding norm-based region collapses to zero, making it infeasible to define a region with $\mathbf{T} \succ 0$. Therefore, the solution obtained from the master problem must ensure that every cluster contains enough points to maintain a positive volume. This guarantees that the subproblem remains well-defined and that the generated cuts contribute effectively to guiding the optimization process.

To ensure convergence, we define an optimality gap $\epsilon > 0$. At each iteration, the master problem provides a valid lower bound on the objective function, while the best-known solution from the subproblems serves as an upper bound. The algorithm terminates when the gap between these bounds falls below ϵ, ensuring that additional iterations will not yield an improvement exceeding the specified tolerance.

As the subproblems are convex in its decision variables for fixed master variables, the iterative refinement ensures that the master problem asymptotically recovers the optimal solution. The finite ϵ-convergence property (Theorem 2.5 in Geoffrion (1972)) guarantees that for any $\epsilon > 0$, the algorithm terminates in a finite number of iterations, provided that the generated cuts sufficiently approximate the projection of the feasible region. Additionally, as the master problem is a mixed-integer formulation with a finite number of possible solutions, the algorithm achieves exact convergence within a finite number of steps (Theorem 2.4 in Geoffrion (1972)).

While this algorithm guarantees convergence, its computational complexity increases with the number of data points, clusters, and problem dimensions. As a result, solving the problem exactly may become impractical for large-scale instances. In the next section, we introduce an approximation algorithm that leverages insights from GBD to provide computationally efficient solutions while maintaining high-quality results.

6. Approximation Algorithm

In this section, we introduce an approximation algorithm designed to approximate Problem (6). The closest algorithm in the literature is that of Martínez-Rego et al. (2013), which is developed specifically for the ℓ_2-norm and is highly sensitive to the initial solution. However, our approximation algorithm extends to general vector norms and mitigates sensitivity to the initial solution.

While the GBD framework provides an exact solution for MVNBC, it faces scalability challenges in high-dimensional settings or when the number of data points and clusters increases. The subproblems involve solving complex conic optimization problems with exponential and logdet cones, making them computationally demanding as the dimensionality grows. Furthermore, the master problem becomes increasingly difficult to solve due to the rapid growth in binary decision variables, leading to impractical solution times for large-scale problems.

To address these computational difficulties, we propose an iterative approximation algorithm inspired by GBD cut generation. The algorithm iteratively refines clustering assignments, but instead of solving the master problem in each iteration, it employs a heuristic reassignment algorithm that efficiently reduces the total volume of clusters while maintaining feasibility. This algorithm is particularly effective in balancing computational efficiency and clustering quality, as it captures key insights from the GBD framework without incurring its full computational cost.

Our algorithm consists of two key components. The reassignment algorithm iteratively removes and reassigns a subset of points to refine the clustering structure, ensuring a progressive reduction in total volume. The exploration mechanism dynamically adjusts the reassignment rate to prevent premature convergence to poor local minima while maintaining computational efficiency. Together, these components provide a practical alternative to scale the clustering method to larger datasets and higher-dimensional spaces.

First, in Section 6.1, we provide an overview of the approximation algorithm. Next, in Section 6.2, we introduce the reassignment algorithm, which iteratively refines the assignment of points to clusters based on boundary points and overlapping regions. Finally, in Section 6.3, we present the exploration mechanism, which adjusts parameters dynamically to balance local refinement and global exploration, ensuring improved convergence.

6.1. An overview of our solution algorithm

The intuition behind this algorithm comes from analyzing Problem (13), where the dual variable u^{*i} is strictly positive only for data points assigned to clusters and located on their respective boundaries. This observation suggests that boundary points contribute significantly to the total volume, making them key candidates for reassignment. To systematically adjust assignments, we define the norm-based distance of point $\mathbf{a}^i$ to cluster k_p in Section 6.2.

The algorithm operates in two main stages: a reassignment algorithm, which iteratively removes and reassigns points to improve clustering, and an exploration mechanism, which adjusts the number of points considered for reassignment to balance local refinement with global adjustments.

At each iteration, the algorithm first updates the transformation matrices and translation vectors defining the cluster regions by solving the minimum-volume subproblems, Problem (11), based on the current assignment of points. Once these region parameters are determined, the algorithm identifies critical points for reassignment, focusing on those near the regions' boundaries. The selected points are then removed from their current clusters and reassigned in a way that minimizes total volume while preserving the feasibility of the Problem (10). Finally, the exploration mechanism adjusts the reassignment rate, which initially results in more considerable modifications and progressively refines the clusters as the algorithm converges.

6.2. Reassignment algorithm

In this algorithm, we focus on points that are either on the boundary of their assigned cluster or are assigned to multiple clusters simultaneously. The algorithm refines the cluster regions by adjusting their assignments while maintaining feasibility.

Boundary points Boundary points are those that lie close to the edge of their assigned region. Their reassignment can significantly affect the shape and size of the clusters. Inspired by the structure of the GBD cuts in Problem (13), we observe that boundary points are those where the dual variable u^{*i} is strictly positive, meaning that they are actively constraining the solution. To formalize this, we define the norm-based distance of a point $\mathbf{a}^i$ to cluster k_p as

$$\mathrm{dist}(\mathbf{a}^i, k_p) = \left\| \mathbf{T}_{k_p}(\mathbf{a}^i - \mathbf{t}_{k_p}) \right\|_p . \tag{14}$$

For a specific $\mathbf{a}^i$, let $(p, k_p^{*i}) \in \arg\min_{k_p \in [K_p], p \in \mathcal{P}} \mathrm{dist}(\mathbf{a}^i, k_p)$. We then select η data points that have the smallest values of $|\mathrm{dist}(\mathbf{a}^i, k_p^{*i}) - 1|$ for each iteration and label them as points in the boundary zone. The value of η is dynamically determined by the exploration mechanism.

Overlapping points Some points may satisfy the membership conditions of multiple clusters, leading to the potential for a reduction of total volume. A point $\mathbf{a}^i$ is classified as overlapping if

$$\mathrm{dist}(\mathbf{a}^i, k_p^1) \leq 1 \quad \text{and} \quad \mathrm{dist}(\mathbf{a}^i, k_p^2) \leq 1,$$

for two distinct clusters k_p^1 and k_p^2. These points are potential candidates for reassignment to improve the total volume.

Reassigning Procedure Once boundary and overlapping points are identified, the reassignment procedure removes them from their initial clusters and reassigns them to reduce total volume. Given an initial solution with membership indicators ${l^*}^i_{k_p}$, the following steps are performed: Once the key points for reassignment are identified, the clustering structure is updated by removing these points from their original clusters and reassigning them in a way that reduces the total volume. The steps are as follows:

1. **Initialize shrunk membership indicators**: Define

$$l'^i_{k_p} := {l^*}^i_{k_p}, \quad \forall i \in [N],\, k_p \in [K_p],\, p \in \mathcal{P}.$$

2. **Shrink regions**: Remove points in the boundary and overlap zones by setting $l'^i_{k_p} = 0$.
3. **Solve subproblems for shrunk regions**: Update the transformation matrices and translation vectors by solving the subproblems (11) for all clusters given l'.
4. **Update point assignments**: Recalculate membership indicators based on the updated cluster shapes

$$l'^i_{k_p} = \begin{cases} 1, & \text{if } \operatorname{dist}(\mathbf{a}^i, k_p) \le 1, \\ 0, & \text{otherwise.} \end{cases}$$

5. **Check feasibility constraints**: Compute the proportion of points that remain unassigned

$$r' = \frac{\left|\left\{i \in [N] \mid \sum_{p\in\mathcal{P}} \sum_{k_p \in [K_p]} l'^i_{k_p} = 0\right\}\right|}{N}.$$

If $r' \le r$, accept the updated regions. Otherwise, proceed to the next step.

6. **Expand the best cluster**: Select the cluster k'_p to which assigning $\frac{r'-r}{2}$ unassigned data points minimizes the total volume increment and reassign additional points by solving the corresponding subproblem.
7. **Update** l^*: Based on the new assignment from the previous step, update the clustering assignment.
8. **Iterate until convergence**: Continue to refine the clusters by iterating through steps 1–7 until the convergence criteria are met.

By iteratively shrinking and expanding the regions, the reassignment algorithm approximates the optimal assignments while ensuring that the outlier constraint is satisfied. The number of points selected for reassignment, η, plays a critical role in the algorithm's performance. Therefore, we develop an exploration algorithm to adjust the value of η for the reassignment algorithm.

6.3. Exploration mechanism

The value of η directly impacts the convergence behavior of the algorithm. A small constant value for η can cause the algorithm to get stuck in local optima, and a large value can lead to meaningless approximations, disrupting stable cluster structures and reducing the quality of the solution. To address this, we introduce an exploration mechanism that dynamically adjusts η throughout the iterative process.

This mechanism allows the algorithm to balance exploration (making large adjustments early in the process) and exploitation (fine-tuning cluster assignments in later iterations). Initially, the algorithm performs more aggressive point reassignments to explore different clustering structures. As the iterations progress, η is gradually reduced to refine the solution and stabilize the clusters.

Adaptive Control of η To regulate the reassignment intensity, we define a phased decreasing schedule for η. Let the total number of iterations allowed for the algorithm be $\mathcal{T}$, and let $iter$ be the current iteration number. We divide the total iterations into phases, each consisting of Ψ iterations. The algorithm undergoes $\mathcal{T}/\Psi$ phases, with ψ denoting the phase index.

At the start of each phase ψ, η is initialized to its maximum allowable value, $\eta_{\max}$, and is then progressively reduced to

$$\eta_{\psi} := \frac{\eta_{\max}}{2^{\psi-1}}. \tag{15}$$

As ψ increases, η is halved, ensuring a gradual transition from aggressive exploration to fine-tuned exploitation. Within each phase, the specific value of η at iteration $iter$ is determined as

$$\eta = \lfloor (1 - \frac{(iter-1) \mod \Psi}{\Psi}) \times \eta_{\psi} \rfloor, \tag{16}$$

where $\lfloor \cdot \rfloor$ is the floor function.

This procedure ensures that η starts at a high value at the beginning of each phase and gradually decreases within that phase.

Exploration Procedure Having established the adaptive adjustment of η, the exploration mechanism follows the steps below to refine the clustering solution:

1. **Initialize clustering parameters**: Start with the current best transformation matrices $\mathbf{T}^*_{kp}$, translation vectors $\mathbf{t}^*_{kp}$, and the total volume Obj^*.
2. **Check termination criteria**: If $iter = \mathcal{T}$, terminate the algorithm and return the current best solution. Otherwise, increment the iteration count and compute η for the current phase using (15) and (16).
3. **Perform reassignment step**: Using the reassignment algorithm in Section 6.2, update cluster assignments by identifying and reassigning η points.
4. **Evaluate solution improvement**: If the new total volume Obj is lower than Obj^*, update the best-known clustering solution

$$\mathbf{T}^*_{kp} \leftarrow \mathbf{T}_{kp}, \quad \mathbf{t}^*_{kp} \leftarrow \mathbf{t}_{kp}, \quad \mathrm{Obj}^* \leftarrow \mathrm{Obj}.$$

5. **Repeat until convergence**: Continue to adjust η and refine the clusters until the solution does not change during a complete phase.

7. Robust Optimization Reformulation based on Minimum-Volume Norm-Based Clustering Uncertainty Set

Having a solution algorithm to solve Problem (6), we can construct our uncertainty set using available data. In this section, we provide the mathematical reformulation of Problem (2) using the presented uncertainty set.

The uncertainty set derived from Problem (6) is

$$\mathcal{U} = \bigcup_{p\in\mathcal{P}} \bigcup_{k_p\in K_p} \mathcal{U}_{k_p}, \tag{17}$$

where $\mathcal{U}_{k_p} = \{\mathbf{u} \mid \|\mathbf{T}^*_{k_p}(\mathbf{u} - \mathbf{t}^*_{k_p})\|_p \le 1\}$. Implementing (17) as the uncertainty set for Problem (2), the robust counterpart reads as

$$\inf_{\mathbf{x}\in\mathbb{R}^n} \left\{ g(\mathbf{x}) \mid f_j(\mathbf{x},\mathbf{u}) \le 0,\ \forall \mathbf{u} \in \mathcal{U}_{k_p}, k_p \in [K_p], p \in \mathcal{P}, j \in [m] \right\}. \tag{18}$$

Substituting $\mathbf{z} = \mathbf{T}^*_{k_p}(\mathbf{u} - \mathbf{t}^*_{k_p})$, for a given $j \in [m]$, we can reformulate each constraint of Problem (18) as

$$f_j(\mathbf{x}, \mathbf{t}^*_{k_p} + \mathbf{T}^{*\,-1}_{\ k_p}\mathbf{z}) \le 0,\ \forall \mathbf{z} \in \mathcal{Z}_{k_p}, k_p \in [K_p], p \in \mathcal{P}, \tag{19}$$

where $\mathcal{Z}_{k_p} = \{\mathbf{z} \mid \|\mathbf{z}\|_p \le 1\}$. Using Theorem 2 in Ben-Tal et al. (2015) and Theorem 2 in Bertsimas et al. (2004), each constraint in Problem (19) reads as

$$\mathbf{t}^{*T}_{\ k_p}\boldsymbol{\zeta}_{k_p} + \left\|(\mathbf{T}^{*\,-1}_{\ k_p})^T\boldsymbol{\zeta}_{k_p}\right\|_{q_p} - f_{j_*}(\mathbf{x},\boldsymbol{\zeta}_{k_p}) \le 0,\ \forall k_p \in [K_p], p \in \mathcal{P}, \tag{20}$$

where $\boldsymbol{\zeta}_{k_p} \in \mathbb{R}^d$ is a new decision variable for any $k_p \in [K_p]$ and $p \in \mathcal{P}$, $\|.\|_{q_p}$ is the dual norm of any $p \in \mathcal{P}$, and $f_{j_*}(\mathbf{x},\cdot)$ is the concave congugate function.

EXAMPLE 1 (LINEAR CONSTRAINT IN THE OPTIMIZATION VARIABLE). Consider a linear constraint as

$$\mathbf{u}^T\mathbf{x} - \beta \le 0,\ \forall \mathbf{u} \in \mathcal{U} \tag{21}$$

where $\mathbf{u} \in \mathbb{R}^d$ is the coefficient vector of the variables, $\beta \in \mathbb{R}$ is the right-hand side parameter, and $\mathcal{U}$ is the uncertainty set defined in (17) with $\mathcal{P} = \{2.0\}$. Note that for $f(\mathbf{x},\mathbf{u}) = \mathbf{u}^T\mathbf{x} - \beta$, we have

$$f_*(\mathbf{x},\zeta) = \begin{cases} \beta, & \text{if } \mathbf{x} = \zeta, \\ -\infty, & \text{if } \mathbf{x} \neq \zeta. \end{cases}$$

So, using Constraint (20), the Constraint (21) reads as

$$\mathbf{t}^{*T}_{\ k_p}\mathbf{x} + \|\mathbf{T}^{*\,-T}_{\ k_p}\mathbf{x}\|_2 \le \beta,\ \forall k_p \in [K_p], p \in \mathcal{P}.$$

□

Reformulating a general RO problem using the MVNBC uncertainty set, the next step is to evaluate the performance of our method in numerical experiments.

8. Numerical Experiments

In this section, we evaluate the effectiveness and efficiency of our proposed method in comparison to existing ones. To ensure transparency and facilitate further research, we provide our implementation in a public repository.[1]

All experiments were conducted using Julia 1.11 and JuMP 1.24.0, with conic subproblems (11) solved using MOSEK 11.0 (MOSEK ApS 2025) and the master problem (10) solved using Gurobi 12.0 (Gurobi Optimization, LLC 2025). K-means clustering was implemented using the *Clustering.jl* package[2], and Gaussian Mixture Model (GMM) clustering was performed with *GaussianMixtures.jl*.[3] All experiments were executed on a high-performance computing (HPC) cluster, specifically on a Genoa partition compute node with 12 tasks, each allocated 16 CPU cores.

As a preliminary step, we assessed the performance of existing solvers on Problem (6) using the *Pajarito.jl* package (Coey et al. 2020). However, even for small instances, the solver failed to produce reasonable solutions. Due to this limitation, we report results only for our proposed algorithms alongside established benchmarks for clustering and uncertainty set construction.

The remainder of this section presents a comprehensive evaluation of our method across different settings. Section 8.2 evaluates both exact and approximation algorithms for solving a newsvendor problem. Finally, Section 8.3 compares the robust solutions achieved by our uncertainty set with the ones proposed by Goerigk and Kurtz (2023).

8.1. Minimum-Volume Noem-Based Clustering as a Clustering Method

In this section, we compare the clustering performance of our proposed method, MVNBC, against two conventional methods: K-means and the Gaussian Mixture Model (GMM).

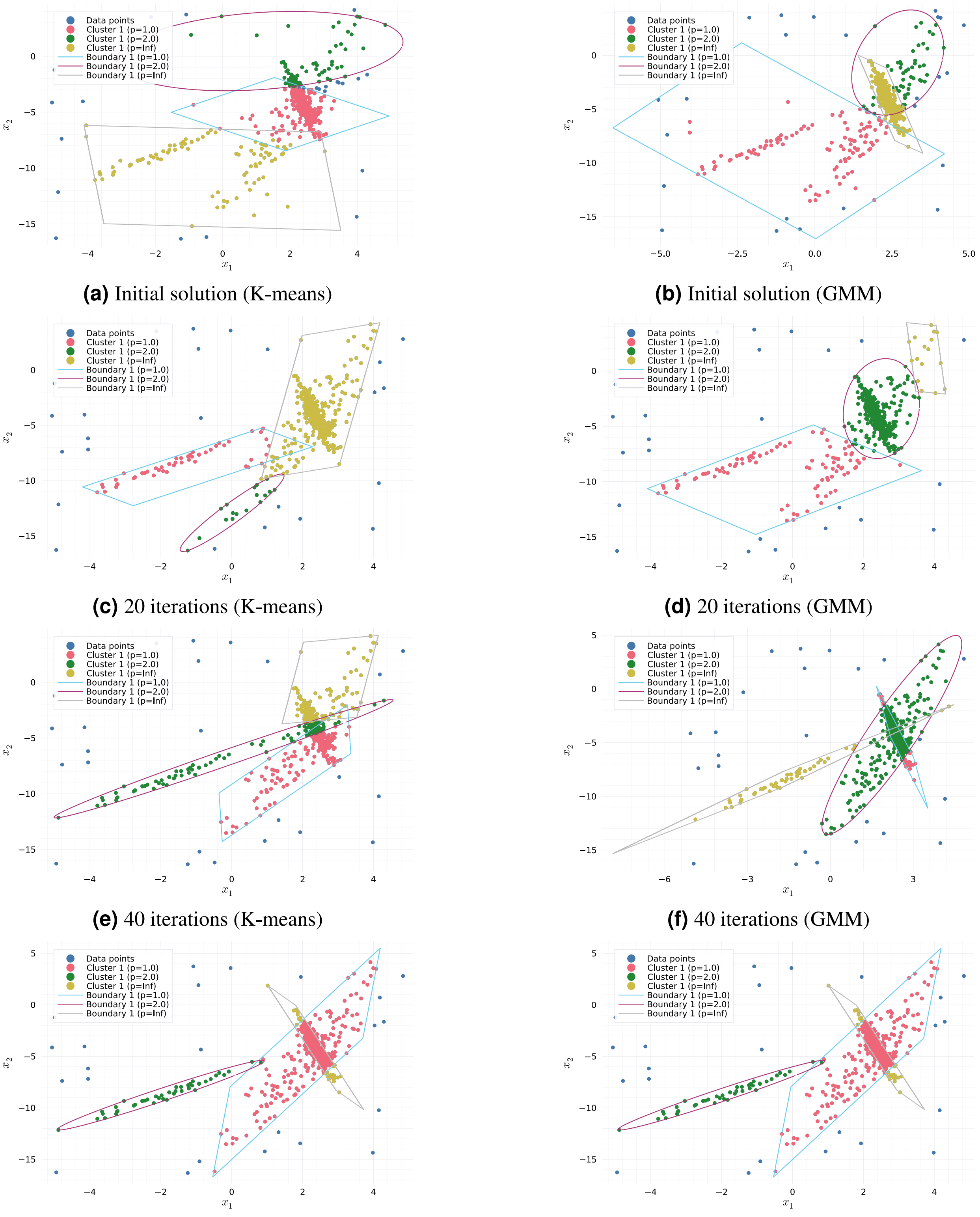
Data generation We generate 30 synthetic datasets using a multi-cluster Gaussian mixture model. Each dataset consists of 500 data points, distributed across 3 clusters with varying scaling factors, rotation matrices, and centers. The scaling factors are drawn from the normal distribution $\mathcal{N}(0.5, 1)$, and a random rotation matrix is applied to each cluster to induce correlations across dimensions. Cluster centers are sampled uniformly from the range $[-6, 6]^2$, and a Gaussian cloud of data points is generated around each center. To introduce outliers, we generate an additional 25 points uniformly over the bounding range of the clusters.

Visual results We fix the number of clusters to $K = 3$ for all methods to ensure a fair comparison under the known ground truth. Experiments are conducted using one cluster per member of $\mathcal{P} = \{1.0, 2.0, \infty\}$. Since K-means and GMM are not designed to detect outliers, we define outliers as the points with the highest values of $\text{dist}(\mathbf{a}^i, k_p^*)$ (as introduced in Section 6.2). Subproblems are solved for each clustering

[1] The code repository is available at https://github.com/alirezaSfid/MVNBC_Optimization. It can be used for further experimentation and insight.

[2] https://github.com/JuliaStats/Clustering.jl

[3] https://github.com/davidavdav/GaussianMixtures.jl



(a) Initial solution (K-means)

(b) Initial solution (GMM)

(c) 20 iterations (K-means)

(d) 20 iterations (GMM)

(e) 40 iterations (K-means)

(f) 40 iterations (GMM)

(g) 80 iterations (K-means)

(h) 80 iterations (GMM)

Figure 1 Clustering performance comparison: The left column presents results using K-means as the initial solution, while the right column shows results using GMM.

Table 1 **Comparison of K-means, GMM, and MVNBC in Terms of Total Volume Reduction and Computational Efficiency**

Method	MVNBC Iterations	Normalized Total Volume (Std. Dev.)	rel-Gap (%)	Time (s)
K-means	Initial Solution	1.00 (0.00)	0.0	2
	20 Iterations	0.33 (0.19)	67.0	1599
	40 Iterations	0.26 (0.15)	73.5	1503
	80 Iterations	**0.25 (0.14)**	74.6	2108
GMM	Initial Solution	1.18 (0.33)	-18.4	35
	20 Iterations	0.31 (0.20)	69.3	1652
	40 Iterations	0.27 (0.15)	73.3	1791
	80 Iterations	**0.25 (0.14)**	74.9	2588

membership index (l_k^i). For MVNBC, we consider two initialization strategies: one based on K-means and the other on GMM. Due to the computational intractability of solving large instances to optimality, we report results obtained using the approximation algorithm.

Among the 30 datasets, one instance with a more complex structure is selected for visual illustration. Figure 1 displays the clustering performance of our approximation algorithm over multiple iterations. The top row shows the initial clusterings from K-means (Figure 1a) and GMM (Figure 1b). The subsequent rows display the clustering results after 20 (Figures 1c, 1d), 40 (Figures 1e, 1f), and 80 iterations (Figures 1g, 1h) for both initialization strategies. As shown in this figure, K-means and GMM are ineffective at capturing the underlying data patterns. For MVNBC, while the quality of clusters is initially influenced by the starting solution, both initialization paths converge to similar results as the algorithm proceeds.

Algorithm effectiveness and computational efficiency We compare the clustering methods in terms of both computational time and the total volume of the resulting clusters. Since datasets exhibit different geometries, we normalize the total volume by dividing it by the volume obtained from the K-means initialization. We also define the *relative gap (rel-Gap)* as the percentage reduction in total volume compared to K-means; higher values indicate greater improvement.

Table 1 reports average performance across the 30 datasets. While MVNBC requires more computational time, it achieves a substantial reduction—up to 90% in total volume compared to K-means and GMM. The effectiveness of the approximation algorithm is consistent across both initializations. Furthermore, the decreasing standard deviation with increased iterations suggests that MVNBC yields more stable and consistent solutions. This highlights its reliability in effectively capturing underlying patterns in the data.

8.2. Newsvendor Problem

In this section, we assess the performance of our uncertainty set on a two-item newsvendor problem, following the experimental setup of Wang et al. (2024). We conduct experiments on two groups of datasets: (i) 30 small-scale datasets, for which both the exact and approximation algorithms are implemented, and (ii)

30 medium-scale datasets, where only the approximation algorithm is implemented due to the intractability of the exact algorithm for larger instances.

Problem description At the beginning of each day, the newsvendor must decide the order quantity $x \in \mathbb{R}^d$, where $d = 2$, for two products. Each unit is purchased at cost $h = (4, 5)$ and sold at price $c = (5, 6.5)$. The actual demand $u \in \mathbb{R}^d$ is uncertain, and only the minimum of the available stock and realized demand can be sold.

Rewriting the objective function, the problem can be reformulated as minimizing a maximum of affine functions under uncertainty

$$\min_{x} \quad h^T x - c^T \min\{x, u\}$$

where the minimum is taken component-wise.

Rewriting the objective, the problem can be expressed as minimizing a maximum-of-affine uncertain function

$$g(x,u) = h^T x + \max\{-c_1x_1 - c_2x_2, -c_1x_1 - c_2u_2, -c_1u_1 - c_2x_2, -c_1u_1 - c_2u_2\}. \tag{22}$$

The uncertain demand vector u is assumed to lie within a union of norm-based uncertainty sets

$$\mathcal{U} = \bigcup_{p\in\mathcal{P}} \bigcup_{k_p \in K_p} \mathcal{U}_{k_p},$$

where each norm-based region is defined as

$$\mathcal{U}_{k_p} = \{\mathbf{u} \mid \|\mathbf{T}_{k_p}(\mathbf{u} - \mathbf{t}_{k_p})\|_p \le 1,\ 0 \le e_1^T \mathbf{u} \le 40,\ 0 \le e_2^T \mathbf{u} \le 40\},$$

with $e_1 = (1,0)^T$ and $e_2 = (0,1)^T$.

Thus, the resulting robust counterpart of the problem is

$$\min_{x} \sup_{u\in\mathcal{U}} \quad g(x,u),$$

whose reformulation is provided in Appendix C, and solved using MOSEK (MOSEK ApS 2025).

8.2.1. Small-size data set

In this experiment, we apply the clusters obtained using both our exact and approximation algorithms.

Data generation We generate 50 data points for each dataset, assuming that demand u is bounded within $[0, 40]^2$. Following Wang et al. (2024), demand is generated based on a log-normal distribution derived from a multivariate normal distribution with the following parameters

$$\mu = \begin{bmatrix} 3.0 \\ 2.8 \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} 0.3 & -0.1 \\ -0.1 & 0.2 \end{bmatrix}.$$

If a generated data point lies outside the region $[0, 40]^2$, we project it back onto the region.

To evaluate the robust solution, we generate 30 independent datasets (each with 50 data points) to construct uncertainty sets and generate 100 testing scenarios per dataset to simulate demand realizations.

Table 2 Relative optimality gap of total volume obtained using the approximation algorithm compared to the exact algorithm

$\mathcal{P}$	$r = 0.00$	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
$[1]$	0.00	0.01	0.01	0.01	0.01	0.02	0.01	0.03	0.03	0.02	0.01
$[2]$	0.00	0.00	0.00	0.01	0.02	0.02	0.01	0.02	0.03	0.01	0.02
$[\infty]$	0.00	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.02	0.03	0.03
$[1,2]$	0.00	0.00	0.01	0.00	0.00	0.04	0.07	0.09	0.08	0.10	0.28
$[1,\infty]$	0.00	0.00	0.00	0.04	0.00	0.07	0.10	0.17	0.10	0.23	0.17
$[2,\infty]$	0.01	0.00	0.00	0.00	0.05	0.02	0.05	0.10	0.10	0.11	0.09

Results We construct uncertainty sets for each of the 30 datasets using the exact algorithm introduced in Section 5 and the approximation algorithm from Section 6. Specifically, we evaluate three commonly used norms in the RO literature, namely $p = \{1, 2, \infty\}$, across a range of outlier rate $r \in \{0.0, 0.05, 0.10, \ldots, 0.45, 0.50\}$. We consider all combinations of norm sets $\mathcal{P} \in \{\{1\}, \{2\}, \{\infty\}, \{1,2\}, \{1,\infty\}, \{2,\infty\}\}$, with the number of clusters fixed at $K_p = 1$ for all $p \in \mathcal{P}$.

To assess the efficiency of the approximation algorithm, we compute the relative optimality gap between the total volume obtained using the exact algorithm (GBD) and the Approximation Algorithm (AA), defined as

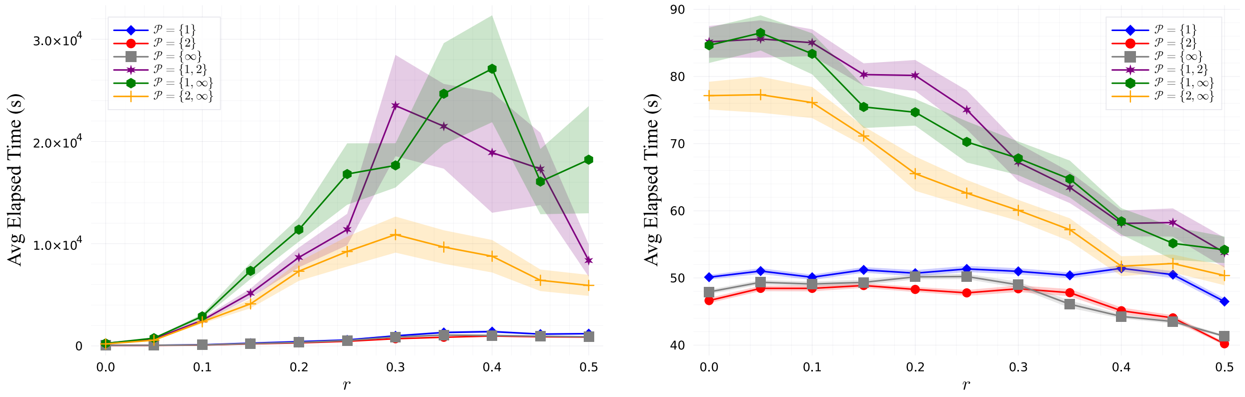
$$\text{Relative Optimality Gap} = \frac{\text{AA} - \text{GBD}}{\text{GBD}}.$$

Table 2 reports the relative for different values of r. The gap remains small at low values of r, demonstrating that the approximation algorithm effectively captures data patterns when sufficient points are included in the uncertainty sets. However, as r increases, more data points are excluded as outliers, substantially reducing the representativeness of the uncertainty set. This reduction increases sensitivity to individual data point assignments and diminishes the performance of the approximation algorithm. In particular, given the small dataset size (50 points), even minor changes in cluster assignments can cause significant variations in volume. Despite this, the approximation algorithm remains a practical alternative, especially for larger datasets where sensitivity to individual points diminishes.

Figure 2 shows the computational time required by the exact and approximation algorithms for various outlier rates and norm sets. As expected, the exact algorithm is significantly more computationally expensive, with a notable increase around $r = 0.3$, where identifying outliers becomes more complex. The approximation algorithm maintains a relatively stable and much lower runtime.

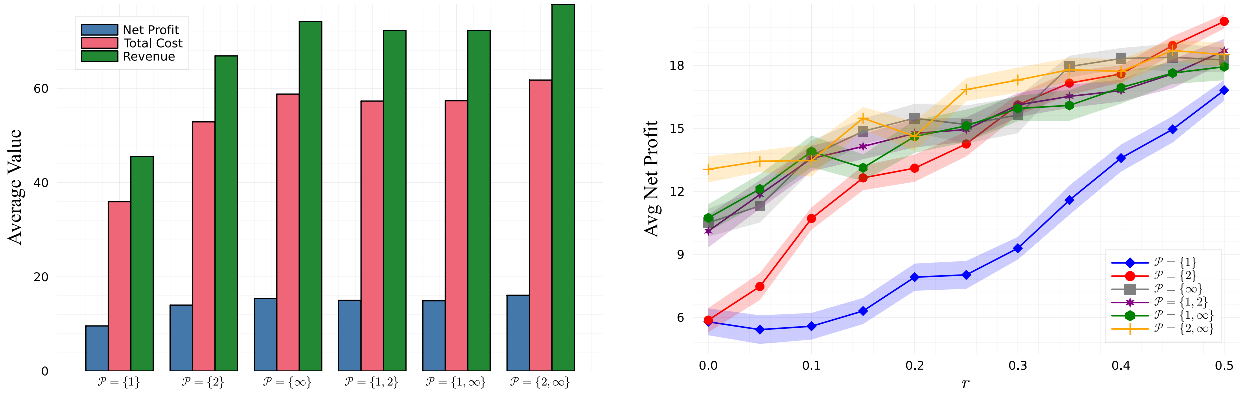
For the sake of exposition, the remainder of this section focuses on results obtained using the approximation algorithm. Results for the exact algorithm are provided in Appendix D.

Figure 3 illustrates the financial performance of robust solutions obtained using various uncertainty sets. As shown in Figure 3a, uncertainty sets constructed using multiple norms, particularly $\mathcal{P} = \{2, \infty\}$, results in robust solutions that yield the highest net profit on average. This result suggests that employing multiple



(a) Exact Solution Algorithm **(b)** Approximation Algorithm

Figure 2 **Computational efficiency across different r values for small-size data set. The shaded ribbons represent the ±1 SEM (standard error of the mean).**



(a) Average total cost, revenue, and net profit across different uncertainty sets defined by $\mathcal{P}$

(b) Average net profit across different outlier rate r

Figure 3 **Financial performance of Approximation Algorithm for small-size data set. The shaded ribbons represent the ±1 SEM (standard error of the mean).**

norms allows for a more expressive and better-fitting representation of uncertainty. Moreover, using two clusters improves the stability of results across datasets, more effectively capturing the underlying data patterns.

Figure 3b highlights how net profit varies with the outlier rate r. As expected, higher outlier rates yield less conservative solutions and, consequently, higher net profit. However, beyond a certain point, this trade-off diminishes as removing too many data points (interpreted as outliers) results in poorly structured uncertainty sets, reducing their reliability.

Overall, the results highlight the significant influence of both the choice of norm and the outlier rate on solution quality. The inferior performance of $\mathcal{P} = \{1\}$ underscores the importance of selecting appropriate norm-based uncertainty sets. In particular, $\mathcal{P} = \{2, \infty\}$ achieves a desirable trade-off between robustness and performance for these datasets.

8.2.2. Medium-size data set As discussed in the previous section, the exact algorithm effectively finds optimal clusters for small-scale instances but does not scale computationally. Therefore, in this section, we evaluate the performance of our approximation algorithm on medium-size instances. To assess the resulting robust solution, we generate 30 independent datasets, each consisting of 200 data points, used to construct uncertainty sets. For each dataset, we generate 200 testing scenarios representing possible demand realizations. To introduce outliers, we add 10 data points drawn uniformly from the bounding range of the previously generated points.

Data generation We generated demand using a mixture of two log-normal distributions, each derived from a multivariate normal distribution with the following parameters

$$\mu_1 = \begin{bmatrix} 3.5 \\ 2.8 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 0.01 & 0.05 \\ 0.05 & 0.01 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} 2.6 \\ 3.4 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 0.03 & -0.01 \\ 0.01 & 0.02 \end{bmatrix}.$$

As before, if a generated data point falls outside the range $[0,40]^2$, we project it onto the region.

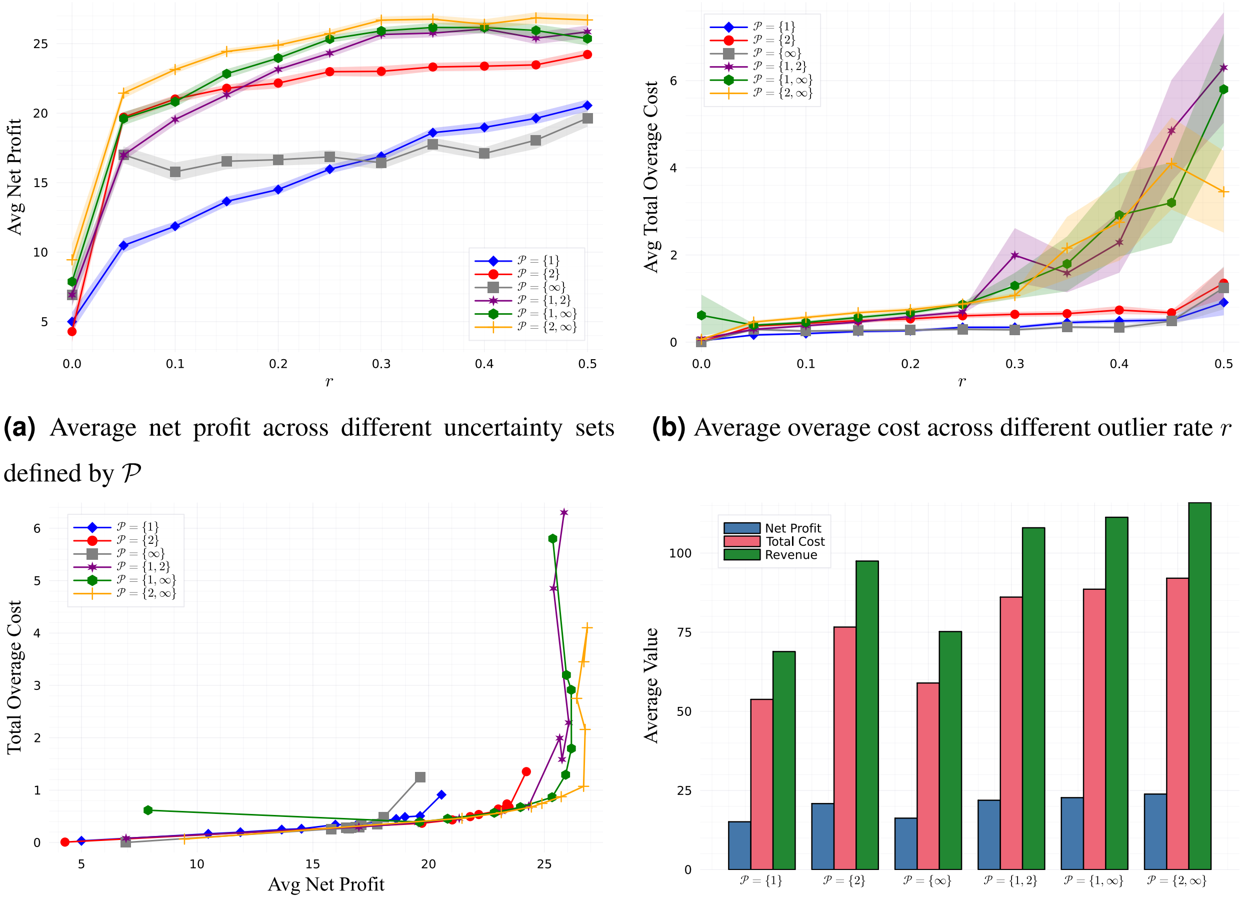
Results We construct uncertainty sets using the same experimental parameters as for the small-scale dataset. Specifically, we consider $p = \{1,2,\infty\}$, outlier rates $r \in \{0.0, 0.05, 0.10, \ldots, 0.50\}$, and norm sets $\mathcal{P} \in \{\{1\},\{2\},\{\infty\},\{1,2\},\{1,\infty\},\{2,\infty\}\}$. For each p, we fix the number of clusters at $K_p = 1$.

To evaluate performance, we analyze how the outlier rate r affects revenue and overage cost (i.e., the cost of ordering more than the realized demand). Figure 4a illustrates how revenue increases with r, as higher outlier rates result in less conservative uncertainty sets. This reduction in conservativeness allows the vendor to order more aggressively, capturing higher revenue. Notably, we observe a performance jump at $r = 0.05$ since excluding a small number of outliers significantly decreases the conservatives of the uncertainty sets. Furthermore, using multiple clusters improves performance, suggesting that additional flexibility in shaping the uncertainty set enhances the algorithm's ability to capture complex data patterns.

Figure 4b shows that for single-cluster uncertainty sets, overage cost remains relatively stable across different values of r. In contrast, two-cluster uncertainty sets exhibit increased overage cost as r increases, which is expected since removing more points leads to a less conservative set. However, at high outlier rates (e.g., r = 0.5), the uncertainty sets become less structured, reducing their effectiveness.

In Figure 4c, we analyze the trade-off between revenue and overage costs. As expected, uncertainty sets with two clusters outperform those with a single cluster in terms of revenue while maintaining a more controlled overage cost. In particular, the single-cluster uncertainty set with $\mathcal{P} = 2.0$ performs better than other single-cluster sets. However, all two-cluster uncertainty sets still lead to higher revenue, demonstrating the advantage of increased flexibility in capturing demand patterns. This result is presented in Figure 4d more clearly. We observe a hidden cap for the Net profit, and the vendor cannot make more profit than that at an average net profit of around 27.

This trade-off highlights an important characteristic of robust decision-making. A less conservative solution, reflected in increased overage costs, allows for higher revenue. However, an overage cost that is too



(a) Average net profit across different uncertainty sets defined by $\mathcal{P}$

(b) Average overage cost across different outlier rate r

(c) The trade-off between overage cost and net revenue

(d) Average total cost, revenue, and net profit across different uncertainty sets defined by $\mathcal{P}$

Figure 4 Financial performance of Approximation Algorithm for medium-size data set. The shaded ribbons represent the ±1 SEM (standard error of the mean).

high suggests an increased risk of exposure. We observe that two-cluster solutions not only achieve higher revenue but also provide a more balanced trade-off, ensuring that decisions remain relatively robust. Thus, employing multiple clusters mitigates conservativeness while maintaining solution robustness

Computational Efficiency We also assess the computational cost of the approximation algorithm by measuring the time required to construct uncertainty sets. Figure 5 presents the total runtime for different norm sets when GMM is used for initialization and the number of iterations is fixed at $\mathcal{T} = 100$. As expected, two-cluster uncertainty sets take approximately three times longer to construct than single-cluster sets. However, their superior performance justifies this additional computational effort.

8.3. Robust Optimization with a Big-size Data Set

In this section, we compare our uncertainty set with the one proposed in Goerigk and Kurtz (2023), using the same RO problem and datasets.

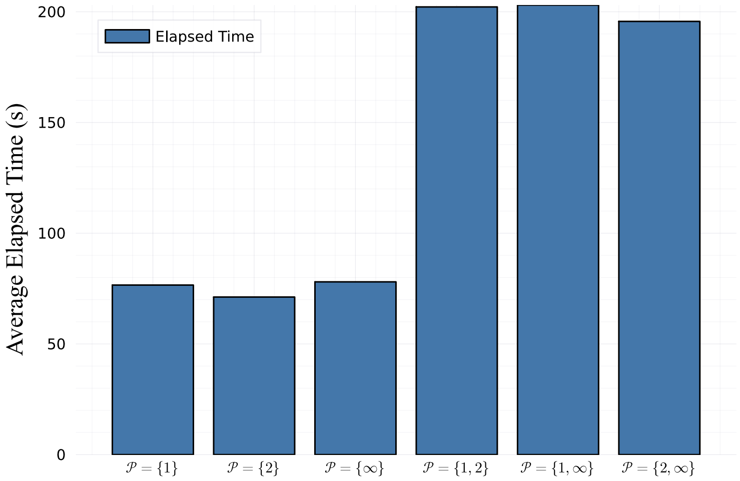


Figure 5 Average elapsed time to construct different uncertainty sets defined by $\mathcal{P}$ using Approximation Algorithm

Problem description We consider the following RO problem

$$\max_{\mathbf{x} \in \mathbb{R}^d} \quad \sum_{i \in [d]} x_i \tag{23a}$$

$$\text{subject to} \quad \mathbf{c}^T \mathbf{x} \leq 1000, \qquad \forall \mathbf{c} \in \mathcal{U}, \tag{23b}$$

$$\mathbf{x} \in [-1, 1]^d, \tag{23c}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the vector of decision variables, $\mathbf{c} \in \mathbb{R}^d$ is the vector of uncertain parameters, and $\mathcal{U}$ is the uncertainty set.

Data generation We use the datasets from Goerigk and Kurtz (2023), which include three types. The first, denoted *Gaussian*, is sampled from a multivariate normal distribution. The second, *Mixed Gaussian*, consists of samples from one of two independent Gaussian distributions, selected with equal probability. The third type, *Polyhedral*, is uniformly sampled from a polyhedron resembling a budgeted uncertainty set (Bertsimas and Sim 2004)

$$\Xi = \left\{ \boldsymbol{\xi} \in \mathbb{R}^d : \xi_i = \underline{\xi_i} + \overline{\xi_i} \varrho_i, \sum_{i=1}^{d} \varrho_i \leq \rho, \boldsymbol{\varrho} \in [0,1]^d \right\},$$

where the bounds $\underline{\xi_i}$ and $\overline{\xi_i}$ are randomly selected, and $\rho = d/2$.

For each of these three data types, 10 configurations were generated, yielding a total of 30 datasets. Each dataset includes a training set of 500 points with 5% noise uniformly sampled from $[0, 300]^d$ (with $d = 20$), and a test set containing 10,000 samples.

Results We construct uncertainty sets $\mathcal{U}$ using MVNBC with 1, 2, and 3 clusters when $p = 2$. We restrict this experiment to $\mathcal{P} = \{2\}$ to leverage the *MinimumVolumeEllipsoids.jl* package, which provides a significantly faster solution process than conic solvers such as MOSEK (MOSEK ApS 2025). Since subproblems must be solved repeatedly during the approximation algorithm, solving them for $p = 1$ or $p = \infty$ becomes computationally prohibitive.

For comparison, we also use the uncertainty sets constructed via the neural network (NN) method from Goerigk and Kurtz (2023). We test various outlier rates $r \in \{0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40\}$, and solve Problem (23) under each uncertainty set.

Table 3 reports the resulting objective values for all uncertainty sets and outlier rates. Higher values indicate better performance. MVNBC consistently outperforms NN by approximately 10%, which is a notable margin. Remarkably, even the most conservative MVNBC setting ($r = 0.10$) outperforms the least conservative NN setting ($r = 0.40$).

Table 3 Objective value comparison of MVNBC and NN approaches.

		Objective value ($\sum_{i=1}^{d} x_i$)			
Type	r	**MVNBC-1**	**MVNBC-2**	**MVNBC-3**	**NN**
Gaussian	0.10	9.58	9.56	9.57	8.95
	0.15	9.60	9.60	9.58	8.97
	0.20	9.63	9.63	9.63	9.01
	0.25	9.64	9.64	9.64	9.05
	0.30	9.66	9.62	9.65	9.08
	0.35	9.69	9.65	9.64	9.11
	0.40	9.70	9.66	9.64	9.14
Mixed Gaussian	0.10	8.81	8.83	8.83	7.82
	0.15	8.84	8.85	8.86	7.88
	0.20	8.87	8.89	8.88	7.92
	0.25	8.91	8.90	8.89	7.98
	0.30	8.91	8.92	8.92	8.04
	0.35	8.92	8.96	8.93	8.09
	0.40	8.92	8.97	8.95	8.14
Polyhedral	0.10	8.56	8.56	8.57	8.13
	0.15	8.59	8.61	8.56	8.14
	0.20	8.65	8.63	8.62	8.14
	0.25	8.65	8.63	8.60	8.16
	0.30	8.68	8.63	8.64	8.17
	0.35	8.68	8.64	8.62	8.19
	0.40	8.71	8.69	8.64	8.20

In terms of feasibility, we observe that Constraint (23b) is almost never violated across all out-of-sample test scenarios. Notably, the results do not show any preference for the number of clusters we used. This result is because of the structure of the problem and objective function. As the objective function in Objective (23a) is linear on $\mathbf{x}$, the worst-case scenario happens at the boundary of the uncertainty set in 1 cluster case, coinciding with the boundary of one of the clusters in 2 clusters case. Consequently, despite the differing representations of data patterns, the objective values obtained under both uncertainty sets are identical. This equivalence underscores the robustness of uncertainty sets constructed using 2 clusters compared with 1 cluster, indicating that both provide the same conservative solutions.

Table 4 Average training time and solution time of Problem (23)

	MVNBC-1		**MVNBC-2**		**MVNBC-3**		**NN**	
Data type	Training time (s)	Solution time (s)	Training time (s)	Solution time (s)	Training time (s)	Solution time (s)	Training time (s)	Solution time (s)
Gaussian	12.1	<0.1	88.0	<0.1	183.8	<0.1	2.8	102.3
Mix. Gaussian	10.4	<0.1	48.7	<0.1	122.5	<0.1	6.6	133.9
Polyhedral	20.3	<0.1	117.0	<0.1	180.2	<0.1	15.7	124.7

Computational efficiency For computational efficiency comparison, we provide the average training and solution times for different methods in Table 4. We utilized the pre-trained network in the repository of Goerigk and Kurtz (2023). We are using a different machine to conduct experiments, so we do not have the exact training time information for the NN case. However, we can estimate the training time by considering the correlation between training times and solution times reported in Goerigk and Kurtz (2023) and the solution time we achieved using our system. This allows us to extrapolate the training times for NN.

Training times for MVNBC-1 and NN require comparably little effort, with averages ranging from 12 to 21 seconds and 2 to 16 seconds, respectively. MVNBC-3 has the longest training times among all, as the optimization problem size is larger. There is a significant difference in solution times between NN and MVNBC. The robust counterpart achieved from MVNBC involves solving a second-order conic optimization problem (see Example 1), which can be solved quickly. On the other hand, the solution algorithm for solving the robust counterpart achieved from the NN uncertainty set generates scenarios iteratively. It is clear that MVNBC is much faster than NN by two orders of magnitude. Furthermore, solving time is more important than training time since we only need to train and construct uncertainty sets once based on historical data.

As demonstrated in Goerigk and Kurtz (2023), the NN-based uncertainty set outperforms the kernel one proposed in Shang et al. (2017) and the box uncertainty set in their experiments. Using the same datasets and computational environment, our numerical results show that our uncertainty set outperforms the NN-based uncertainty set in the problem under consideration. This suggests that our method provides a more effective uncertainty set within this specific setting.

9. Conclusion

We have proposed a norm-based clustering method called Minimum Volume Norm-Based Clustering (MVNBC) to minimize the volumes of norm-based regions defining clusters of data points. MVNBC effectively captures data patterns and identifies outliers at a specified rate. We formulated MVNBC as a Mixed-Integer Conic Optimization (MICO) problem, which includes a quadratic or linear cone for each data point and region, as well as exponential and logdet cones for each region. To solve this problem, we developed an exact solution algorithm based on generalized Benders decomposition (GBD), which exploits the problem structure by decomposing it into a master problem (handling binary assignments) and subproblems (optimizing the volume of norm-based regions). Furthermore, we introduced an iterative approximation algorithm that iteratively refines cluster assignments to minimize total volume, making the algorithm scalable to larger datasets.

Beyond clustering, we utilized the MVNBC-generated regions as a data-driven uncertainty set in RO. We formulated the robust counterpart for a general convex optimization problem and demonstrated that the reformulation remains computationally tractable and solvable with conic solvers.

To evaluate our method, we conducted three numerical experiments. First, we tested MVNBC as a clustering method and demonstrated its superior performance compared to K-means and GMM in capturing data patterns and minimizing total cluster volume. Second, we analyzed the performance of both the exact GBD-based algorithm and the approximation algorithm in solving the newsvendor problem, highlighting the advantages of multiple clusters and flexible region shapes in improving solutions. Finally, we applied MVNBC-based uncertainty sets to a benchmark RO problem (Goerigk and Kurtz 2023) and showed that it improved the objective value by 10% while significantly reducing computational time compared to the selected uncertainty set from the literature.

While our method shows strong performance, some challenges remain. The exact GBD-based algorithm becomes computationally expensive for high-dimensional and large-scale datasets, suggesting the need for further scalability improvements. Future research could explore alternative decomposition techniques, adaptive strategies to refine the approximation algorithm, and extensions to more RO problems. Additionally, applying MVNBC in dynamic settings, where uncertainty evolves over time, could open new avenues for robust decision-making.

Acknowledgments

We would like to express our sincere gratitude to the HPC Lab at Eindhoven University of Technology for providing access to the Snellius supercomputer.

References

Ahipaşaoğlu SD (2015) Fast algorithms for the minimum volume estimator. *Journal of Global Optimization* 62(2):351–370.

Asgari SD, Mohammadi E, Makui A, Jafari M (2024) Data-driven robust optimization based on position-regulated support vector clustering. *Journal of Computational Science* 76:102210.

Avdija D (2025) Gaussianmixtures.jl: Gaussian mixture models in julia. `https://github.com/davidavdav/GaussianMixtures.jl`, accessed: 2025-04-08.

Barnes ER (1982) An algorithm for separating patterns by ellipsoids. *IBM Journal of Research and Development* 26(6):759–764.

Baron O, Milner J, Naseraldin H (2011) Facility location: A robust optimization approach. *Production and Operations Management* 20(5):772–785.

Bemporad A, Filippi C, Torrisi FD (2004) Inner and outer approximations of polytopes using boxes. *Computational Geometry* 27(2):151–178.

Ben-Tal A, Den Hertog D, Vial JP (2015) Deriving robust counterparts of nonlinear uncertain inequalities. *Mathematical Programming* 149(1):265–299.

Ben-Tal A, Nemirovski A (1998) Robust convex optimization. *Mathematics of Operations Research* 23(4):769–805.

Ben-Tal A, Nemirovski A (2000) Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming* 88:411–424.

Ben-Tal A, Nemirovski A (2002) Robust optimization–methodology and applications. *Mathematical Programming* 92:453–480.

Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. *SIAM Review* 53(3):464–501.

Bertsimas D, Gupta V, Kallus N (2018) Data-driven robust optimization. *Mathematical Programming* 167:235–292.

Bertsimas D, Pachamanova D, Sim M (2004) Robust linear optimization under general norms. *Operations Research Letters* 32(6):510–516.

Bertsimas D, Sim M (2004) The price of robustness. *Operations Research* 52(1):35–53.

Bertsimas D, Thiele A (2006) *Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty*, chapter Chapter 4, 95–122. INFORMS TutORials in Operations Research (Catonsville, MD: INFORMS).

Campello RJGB, Kröger P, Sander J, Zimek A (2020) Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10(2):e1343.

Candela JA (1996) Exact iterative computation of the multivariate minimum volume ellipsoid estimator with a branch and bound algorithm. Prat A, ed., *COMPSTAT*, 175–180 (Heidelberg: Physica-Verlag HD).

Coey C, Lubin M, Vielma JP (2020) Outer approximation with conic certificates for mixed-integer convex problems. *Mathematical Programming Computation* 12(2):249–293.

De Rosa A, Khajavirad A, Wang Y (2024) On the power of linear programming for k-means clustering. *arXiv preprint arXiv:2402.01061* .

Fischetti M, Ljubić I, Sinnl M (2017) Redesigning benders decomposition for large-scale facility location. *Management Science* 63(7):2146–2162.

Gabrel V, Murat C, Thiele A (2014) Recent advances in robust optimization: An overview. *European Journal of Operational Research* 235(3):471–483.

Geoffrion AM (1972) Generalized benders decomposition. *Journal of Optimization Theory and Applications* 10:237–260.

Goerigk M, Khosravi M (2023) Optimal scenario reduction for one-and two-stage robust optimization with discrete uncertainty in the objective. *European Journal of Operational Research* 310(2):529–551.

Goerigk M, Kurtz J (2023) Data-driven robust optimization using deep neural networks. *Computers & Operations Research* 151:106087.

Gorissen BL, Yanıkoğlu İ, Den Hertog D (2015) A practical guide to robust optimization. *Omega* 53:124–137.

Gurobi Optimization, LLC (2025) Gurobi Optimizer Reference Manual. URL https://www.gurobi.com.

Han J, Kamber M, Pei J (2012) 8 - classification: Basic concepts. Han J, Kamber M, Pei J, eds., *Data Mining (Third Edition)*, 327–391, The Morgan Kaufmann Series in Data Management Systems (Boston: Morgan Kaufmann), third edition edition.

Kang Z, Marandi A, Basten RJ, de Kok T (2023) Robust spare parts inventory management. *Available at SSRN 4553430* .

Khachiyan LG (1996) Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research* 21(2):307–320.

Kumar M, Orlin JB (2008) Scale-invariant clustering with minimum volume ellipsoids. *Computers & Operations Research* 35(4):1017–1029.

Li Y, Yorke-Smith N, Keviczky T (2025) On data-driven robust optimization with multiple uncertainty subsets: Unified uncertainty set representation and mitigating conservatism. *arXiv preprint arXiv:2502.11867* .

Loger B, Dolgui A, Lehuédé F, Massonnet G (2024) Approximate kernel learning uncertainty set for robust combinatorial optimization. *INFORMS Journal on Computing* 36(3):900–917.

Margalit D, Rabinoff J, Rolen L (2019) *Interactive linear algebra* (Atlanta, GA, USA: School of Mathematics, Georgia Institute of Technology).

Martínez-Rego D, Castillo E, Fontenla-Romero O, Alonso-Betanzos A (2013) A minimum volume covering approach with a set of ellipsoids. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(12):2997–3009.

MOSEK ApS (2025) *MOSEK Optimizer API for C*. URL https://docs.mosek.com/latest/capi/index.html.

Moshkovitz M, Dasgupta S, Rashtchian C, Frost N (2020) Explainable k-means and k-medians clustering. III HD, Singh A, eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 7055–7065 (Virtual Conference: PMLR).

Neofytou A, Liu B, Akartunalı K (2025) Compact uncertainty sets for robust optimization based on bootstrapped dirichlet process mixture model. *Optimization* 1–44.

Nielsen F (2016) *Hierarchical Clustering*, 195–211 (Cham: Springer International Publishing).

Ning C, You F (2018) Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods. *Computers & Chemical Engineering* 112:190–210.

Rosen J (1965) Pattern separation by convex programming. *Journal of Mathematical Analysis and Applications* 10(1):123–134.

Shang C, Huang X, You F (2017) Data-driven robust optimization based on kernel learning. *Computers & Chemical Engineering* 106:464–479.

Shioda R, Tunçel L (2007) Clustering via minimum volume ellipsoids. *Computational Optimization and Applications* 37(3):247–295.

Soyster AL (1973) Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research* 21(5):1154–1157.

Suyal M, Sharma S (2024) A review on analysis of k-means clustering machine learning algorithm based on unsupervised learning. *Journal of Artificial Intelligence and Systems* .

Van Aelst S, Rousseeuw P (2009) Minimum volume ellipsoid. *Wiley Interdisciplinary Reviews: Computational Statistics* 1(1):71–82.

Wang I, Becker C, Van Parys B, Stellato B (2023) Learning decision-focused uncertainty sets in robust optimization. *arXiv preprint arXiv:2305.19225* .

Wang I, Becker C, Van Parys B, Stellato B (2024) Mean robust optimization. *Mathematical Programming* 1–43.

Wang X (2005) Volumes of generalized unit balls. *Mathematics Magazine* 78(5):390–395.

Woodruff DL, Rocke DM (1993) Heuristic search algorithms for the minimum volume ellipsoid. *Journal of Computational and Graphical Statistics* 2(1):69–95.

Zhang S, Jia R, He D, Chu F (2022) Data-driven robust optimization based on principle component analysis and cutting plane methods. *Industrial & Engineering Chemistry Research* 61(5):2167–2182.

Zhang Y, Feng Y, Rong G (2017) New robust optimization approach induced by flexible uncertainty set: Optimization under continuous uncertainty. *Industrial & Engineering Chemistry Research* 56(1):270–287.

Appendices

The appendix of this paper contains four sections. Section A provides mathematical proofs related to minimum volume transformations. In Section B, we provide implementation details of our generalized Benders decomposition (GBD) framework. Section C explicitly formulates the robust counterpart of the newsvendor problem discussed in the paper. Section D presents numerical results obtained from the exact algorithm (GBD) applied to the newsvendor problem.

A. Mathematical Supplements

A.1. Proof of Theorem 1

A brief proof of this theorem is provided in Section 4.3.3 Margalit et al. (2019). Here is a more elaborate proof for the theorem.

DEFINITION 2 (SEE DEFINITION 4.3.1 IN MARGALIT ET AL. (2019)). The **paralellepiped** determined by n vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\cdots$, $\boldsymbol{v}_n$ in $\mathbb{R}^n$ is the subset

$$P = \{a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \cdots + a_n\boldsymbol{v}_n | 0 \leq a_1, a_2, \ldots, a_n \leq 1\} \tag{24}$$

THEOREM 4 **(See Theorem 4.3.6 in Margalit et al. (2019)).** *Let* $\mathbf{v}_1$*,* $\mathbf{v}_2$*, …,* $\mathbf{v}_n$ *be vectors in* $\mathbb{R}^n$*, let* P *be the parallelepiped determined by these vectors, and let* $\mathbf{T}$ *be the matrix with rows* $\mathbf{v}_1$*,* $\mathbf{v}_2$*, …,* $\mathbf{v}_n$*. Then the absolute value of determinant of* $\mathbf{T}$ *is the volume of* P

$$|\det(\mathbf{T})| = \text{vol}(P) \tag{25}$$

Proof (See Section 4.3.2 from Margalit et al. (2019))

Proof of Theorem 1 Let $\mathcal{C}$ be the unit cube in d dimesnion, $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_d$ be the columns of $\mathbf{T}$, and P be the paralellepiped determined by these vectors, i.e. $\mathrm{T}(\mathcal{C}) = P$ and $\text{vol}(\mathrm{T}(\mathcal{C})) = |\det(\mathbf{T})|$ (from Theorem 4). Let $\varepsilon > 0$ and $\varepsilon\mathcal{C}$ be the cube with side lengths ε, i.e., the parallelepiped determined by the vectors $\varepsilon e_1, \varepsilon e_2, \ldots, \varepsilon e_d$ and εP defines similarly. We have

$$\text{vol}(\varepsilon P) = \varepsilon^d \text{vol}(P) = \varepsilon^d |\det(\mathbf{T})| \tag{26}$$

where $\text{vol}(P) = \text{vol}(\mathrm{T}(\mathcal{C}))$. The volume of $\varepsilon\mathcal{C}$ is ε^d, because we scaled each of the d standard vectors by a factor ε. We obtain that the volume of εP equals to $\varepsilon^d |\det(\mathbf{T})|$.
Moreover, for any $\mathbf{x} \in \mathbb{R}^d$

$$\mathrm{T}(\mathbf{x} + \varepsilon\mathcal{C}) = \mathrm{T}(\mathbf{x}) + \mathrm{T}(\varepsilon\mathcal{C}) = \mathrm{T}(\mathbf{x}) + \varepsilon P \tag{27}$$

Since a translation does not change the volumes, T scales the volume of a translate of $\varepsilon\mathcal{C}$ by $|\det(\mathbf{T})|$ (from Theorem 4), i.e.

$$\text{vol}(\mathrm{T}(\mathbf{x} + \varepsilon\mathcal{C})) = \text{vol}(\mathrm{T}(\varepsilon\mathcal{C})) = \varepsilon^d |\det(\mathbf{T})| \tag{28}$$

Furthermore, we know

$$\text{vol}(\mathcal{S}) = \oint_{\mathcal{S}} \mathrm{d}s = \int \cdots \int_{\mathbf{x} \in \mathcal{S}} \mathrm{d}x \tag{29}$$

where $\mathcal{S}$ is a region in $\mathbb{R}^d$ and

$$\mathrm{d}s = \mathrm{d}e_1 \cdot \mathrm{d}e_2 \cdot \cdots \cdot \mathrm{d}e_d = \lim_{\varepsilon \to 0} \varepsilon e_1 \cdot \varepsilon e_2 \cdot \cdots \cdot \varepsilon e_d = e_1 \cdot e_2 \cdot \cdots \cdot e_d \lim_{\varepsilon \to 0} \varepsilon^d. \tag{30}$$

where e_i be the vector of all zeros except the i-th component, which is 1. The translation of this subset ($\mathcal{S}$) by T is another subset ($\mathcal{V}$), which consists of translated $\mathrm{d}s$. The volume of this subset ($\mathcal{V}$) is calculated by

$$\mathrm{vol}(\mathcal{V}) = \oint_{\mathcal{V}} \mathrm{d}v = \int \cdots \int_{\mathbf{x} \in \mathcal{V}} \mathrm{d}x \tag{31}$$

where

$$\begin{aligned} \mathrm{d}v &= \mathrm{T}(\mathrm{d}s) && (32)\\ &= \mathrm{T}(e_1 \cdot e_2 \cdot \cdots \cdot e_d \lim_{\varepsilon \to 0} \varepsilon^d) && (33)\\ &= \lim_{\varepsilon \to 0} \varepsilon^d \times \mathrm{T}(e_1 \cdot e_2 \cdot \cdots \cdot e_d) && (34)\\ &= \lim_{\varepsilon \to 0} \varepsilon^d \times |\det(\mathbf{T})| \times e_1 \cdot e_2 \cdot \cdots \cdot e_d && (35)\\ &= |\det(\mathbf{T})| \times \mathrm{d}s. && (36) \end{aligned}$$

So,

$$\begin{aligned} \mathrm{vol}(\mathcal{V}) &= \oint_{\mathcal{V}} \mathrm{d}v = \oint_{\mathcal{S}} \mathrm{T}(\mathrm{d}s) = \oint_{\mathcal{S}} |\det(\mathbf{T})| \cdot \mathrm{d}s = |\det(\mathbf{T})| \cdot \oint_{\mathcal{S}} \mathrm{d}s && (37)\\ &= |\det(\mathbf{T})| \cdot \mathrm{vol}(\mathcal{S}) && (38) \end{aligned}$$

□

A.2. Proof of Corollary 1

Proof Given $\mathbf{x} \in \mathbb{R}^d$, let us set $\boldsymbol{y} = \mathbf{T}(\mathbf{x} - \mathbf{t})$. Since $\mathbf{T}$ is invertible, we have

$$\mathbf{x} = \mathbf{T}^{-1}\boldsymbol{y} + \mathbf{t}. \tag{39}$$

Therefore, $\mathcal{S}$ can be represented as

$$\mathcal{S} = \{\mathbf{T}^{-1}\boldsymbol{y} + \mathbf{t} \mid \|\boldsymbol{y}\|_p \le 1\} = \mathrm{T}(\mathcal{B}^p), \tag{40}$$

where $\mathrm{T}(\mathbf{x}) = \mathbf{T}^{-1}\mathbf{x} + \mathbf{t}$. As translation $\mathbf{t}$ does not change the volume, using Theorem 1, we have

$$\mathrm{vol}(\mathcal{S}) = |\det(\mathbf{T}^{-1})| \cdot \mathrm{vol}(\mathcal{B}^p) = \frac{1}{|\det(\mathbf{T})|} \cdot \mathrm{vol}(\mathcal{B}^p), \tag{41}$$

where $\mathcal{B}^p$ is the unit ball of ℓ_p-norm in $\mathbb{R}^d$, i.e., $\mathcal{B}^p = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_p \le 1\}$. So, the volume of the region $\mathcal{S}$ is inversely proportional to the determinant of $\mathbf{T}$. The fact that the volume of the unit ball $\mathcal{B}^p$ in $\mathbb{R}^d$ is equal to (See e.g., Wang (2005))

$$\mathrm{vol}(\mathcal{B}^p) = \frac{(2\Gamma(1 + \frac{1}{p}))^d}{\Gamma(1 + \frac{d}{p})}, \tag{42}$$

concludes the proof. □

B. Implementation Details of the Generalized Benders Decomposition Framework

This appendix provides implementation details of the generalized Benders decomposition (GBD) framework used to solve Problem (6), including computational setup, solver integration, and specific enhancements employed for computational efficiency.

B.1. Master Problem Setup and Solver Enhancements

We solve the master problem (10), formulated as a Mixed-Integer Linear Optimization (MILO), using Gurobi (Gurobi Optimization, LLC 2025). We utilize Gurobi's callback mechanism to dynamically handle binary variable assignments. Within each callback iteration, binary assignments are evaluated, and corresponding conic subproblems are solved. GBD cuts are then generated based on these subproblem solutions and dynamically added to the master problem using Gurobi's `addLazyConstraints`. This technique ensures only violated cuts are included, significantly reducing computational overhead.

To improve solver performance and avoid generating infeasible clusters, we refine the original feasibility region (Constraints (10b)–(10c)) as follows

$$\begin{aligned}
& w^i = \sum_{p\in\mathcal{P}} \sum_{k_p\in[K_p]} l^i_{k_p}, && \forall i \in [N], \\
& \sum_{i\in[N]} w^i = (1-r)N, && \\
& \sum_{i\in[N]} l^i_{k_p} \geq d+1, && \forall k_p \in [K_p],\, p \in \mathcal{P}.
\end{aligned}$$

These refinements ensure each cluster contains a sufficient number of points, eliminating the possibility of degenerate clusters and thus eliminating the need for feasibility cuts.

B.2. Solving Conic Subproblems and Warm-starting

The conic subproblems are solved using MOSEK (MOSEK ApS 2025), a specialized solver for conic optimization problems. These subproblems, which arise from fixed binary assignments provided by the master problem, are solved independently for each cluster and norm in parallel. After each subproblem solution, we retrieve optimal dual variables associated with Constraint (11d) to calculate subgradients used for GBD cut generation.

To leverage computational efficiency, we initialize each conic subproblem formulation only once at the beginning of the algorithm for each cluster k_p. Instead of reconstructing these formulations from scratch at each iteration, we update the existing formulations to reflect new binary assignments, $\bar{l}$. This strategy enables MOSEK (MOSEK ApS 2025) to utilize warm-start information from previous iterations, significantly reducing the computational time required for subsequent subproblem solutions.

B.3. Warm-start Initialization via Gaussian Mixture Models

We employ a warm-start initialization strategy using Gaussian Mixture Models (GMM), implemented via the `GaussianMixtures.jl` package (Avdija 2025), to obtain an initial clustering of data points. This clustering provides a high-quality initial solution, accelerating convergence by reducing the time spent exploring suboptimal clustering.

Additionally, we adopt experiment-specific tolerances for numerical convergence, ensuring they are tight enough to yield accurate results without incurring excessive computational cost. This adaptive technique strikes a balance between numerical precision and computational performance.

C. Robust Counterpart of the Newsvendor Problem

In this section, we provide the robust counterpart of Problem (22) in Section 8.2.

Using the uncertainty set provided by MVNBC, Problem (22) reads as

$$\min_{x,y,w^{11}_{k_p},w^{21}_{k_p},w^{12}_{k_p},w^{22}_{k_p},w^{13}_{k_p},w^{23}_{k_p}} \quad h^T x + y \tag{44a}$$

$$\text{subject to} \quad -c^T x \le y, \tag{44b}$$

$$\begin{aligned} & -e_1^T c e_1^T x - e_2^T c e_2^T \mathbf{T}_{k_p}^{-1}\mathbf{t}_{k_p} \\ & \quad + (\mathbf{T}_{k_p}[40,40]^T + \mathbf{t}_{k_p})^T I_2 w^{11}_{k_p} + \|w^{21}_{k_p}\|_q \le y, \qquad k_p \in [K_p],\ p \in \mathcal{P}, \end{aligned} \tag{44c}$$

$$w^{11}_{k_p} + w^{21}_{k_p} = -e_2^T c e_2^T \mathbf{T}_{k_p}^{-1}, \qquad k_p \in [K_p],\ p \in \mathcal{P}, \tag{44d}$$

$$I_2 w^{11}_{k_p} \ge 0, \qquad k_p \in [K_p],\ p \in \mathcal{P}, \tag{44e}$$

$$\begin{aligned} & -e_2^T c e_2^T x - e_1^T c e_1^T \mathbf{T}_{k_p}^{-1}\mathbf{t}_{k_p} \\ & \quad + (\mathbf{T}_{k_p}[40,40]^T + \mathbf{t}_{k_p})^T I_2 w^{12}_{k_p} + \|w^{22}_{k_p}\|_q \le y, \qquad k_p \in [K_p],\ p \in \mathcal{P}, \end{aligned} \tag{44f}$$

$$w^{12}_{k_p} + w^{22}_{k_p} = -e_1^T c e_1^T \mathbf{T}_{k_p}^{-1}, \qquad k_p \in [K_p],\ p \in \mathcal{P}, \tag{44g}$$

$$I_2 w^{12}_{k_p} \ge 0, \qquad k_p \in [K_p],\ p \in \mathcal{P}, \tag{44h}$$

$$-c^T \mathbf{T}_{k_p}^{-1}\mathbf{t}_{k_p} + (\mathbf{T}_{k_p}[40,40]^T + \mathbf{t}_{k_p})^T I_2 w^{13}_{k_p} + \|w^{23}_{k_p}\|_q \le y, \quad k_p \in [K_p],\ p \in \mathcal{P}, \tag{44i}$$

$$w^{13}_{k_p} + w^{23}_{k_p} = -c^T \mathbf{T}_{k_p}^{-1}, \qquad k_p \in [K_p],\ p \in \mathcal{P}, \tag{44j}$$

$$I_2 w^{13}_{k_p} \ge 0, \qquad k_p \in [K_p],\ p \in \mathcal{P}, \tag{44k}$$

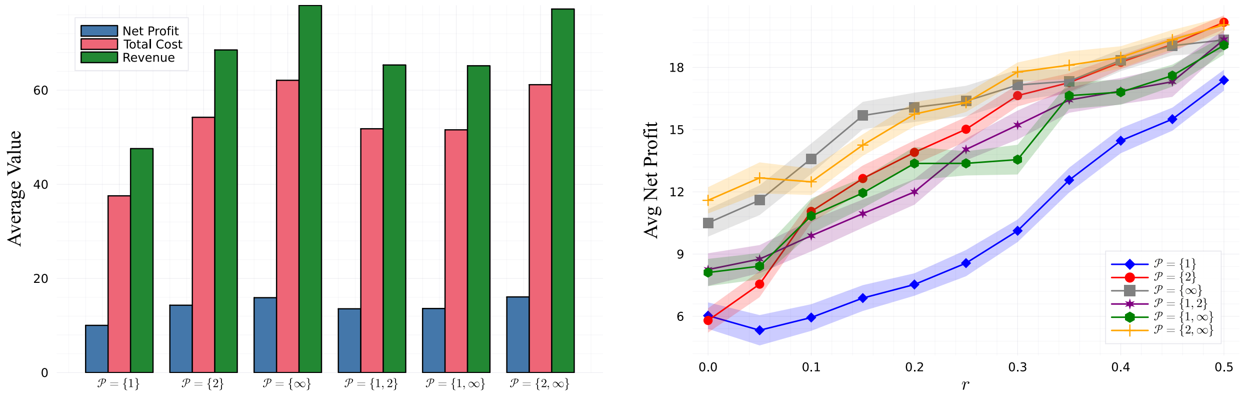
where w^{1j}_{kp}, w^{2j}_{kp} for $j \in \{1,2,3\}$ are auxiliary decision variables introduced to reformulate the robust constraints using conic representations, I_2 is the 2×2 identity matrix, and $q = 1 + \frac{1}{p-1}$ is the dual norm associated with the norm p. Problem (44) is a conic problem when $p \in \{1, 2, \infty\}$, and we solve it using the MOSEK (MOSEK ApS 2025).

D. Results for the Newsvendor Problem Using the Exact Algorithm

In this appendix, we present the results for the Newsvendor problem using the exact algorithm. These results complement the analysis in Section 8.2, where we focused on the solutions obtained by our Approximation Algorithm (AA).

Figure 6 shows the performance of various uncertainty sets in terms of cost, revenue, and net profit. Unlike the results from the approximation algorithm, the exact algorithm provides globally optimal clusters but at a significantly higher computational cost.

The exact algorithm (GBD) provides globally optimal clusters but comes at a significantly higher computational cost. The results confirm that using multiple norm sets ($\mathcal{P} = \{2, \infty\}$) leads to more effective uncertainty sets for RO. However, as seen in Figure 7, the exact algorithm becomes computationally infeasible for a higher number of clusters or data points, making the approximation algorithm a practical alternative for larger datasets.



(a) Average cost, revenue, and net profit across different uncertainty sets defined by $\mathcal{P}$

(b) Average net profit across different different outlier rate r

Figure 6 **Financial performance of Exact Solution Algorithm for a small-size data set. The shaded ribbons represent the ±1 SEM (standard error of the mean).**

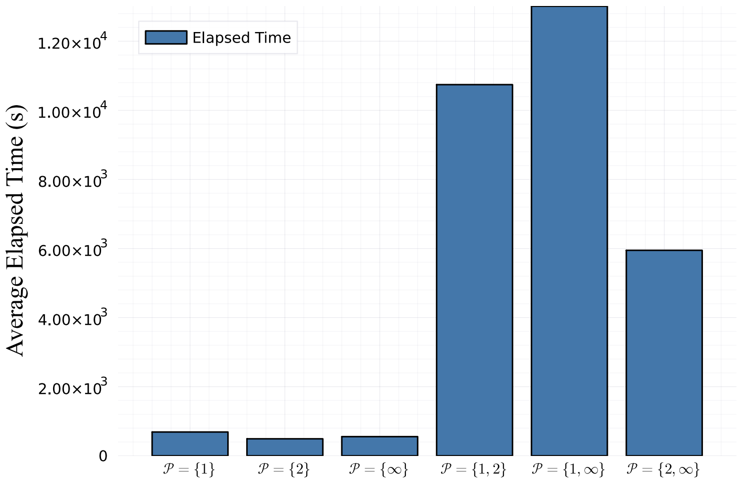


Figure 7 **Average elapsed time to construct different uncertainty sets defined by $\mathcal{P}$ using Exact Solution Algorithm**