Improving the Tractability of SVC-based Robust Optimization
Benoît Loger, Alexandre Dolgui, Fabien Lehuédé, Guillaume Massonnet

To cite this version:
hal-03832914v2

HAL Id: hal-03832914
https://hal.science/hal-03832914v2
Submitted on 3 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Improving the Tractability of SVC-based Robust Optimization

Benoît Loger* Alexandre Dolgui* Fabien Lehúédé*
Guillaume Massonnet*

*IMT Atlantique, LS2N, 4 rue Alfred Katler 44300 Nantes FRANCE
(e-mail: {benoit.loger, alexandre.dolgui, fabien.lehude, guillaume.massonnet}@imt-atlantique.fr).

Abstract: Support Vector Clustering (SVC) has been proposed in the literature as a data-driven approach to build uncertainty sets in robust optimization. Unfortunately, the resulting SVC-based uncertainty sets induces a large number of additional variables and constraints in the robust counterpart of mathematical formulations. We propose two methods to approximate the resulting uncertainty sets and overcome these tractability issues. We evaluate these approaches on a production planning problem inspired from an industrial case study. The results obtained are compared with those of the SVC-based uncertainty set and the well known budget-based uncertainty set. We find that the approximated uncertainty set based formulation can be solved much faster than the SVC-based formulation. Still, the obtained solutions are comparable to the SVC-based solutions in term of performance.

Keywords: Data-driven, Robust Optimization, Production planning

1. INTRODUCTION

Robust optimization (RO) has become a popular approach to deal with uncertainties in optimization problems in the last decades and have been applied to a wide range of application fields (Söüzür and Thiele, 2016). The core concept of RO is to restrict the possible realizations of uncertain parameters to a given uncertainty set \( \mathcal{U} \), then to optimize against the worst case within this set to obtain solutions that are immunized against all scenarios included in \( \mathcal{U} \). With the growing complexity of supply chain and production systems, RO appears to be a promising approach to improve the performances and the reliability of industrial systems by reducing the impact of uncertain or unpredictable events. Since the pioneering work of Soyster (1973), a lot of efforts have been dedicated to propose and characterize different types of uncertainty sets in order to obtain tractable robust models (El Ghaoui et al., 1998; Ben-Tal and Nemirovski, 1998; Bertsimas and Sim, 2004).

Among the numerous applications of RO, Bertsimas and Thiele (2006) were the first to apply the model of Bertsimas and Sim (2004) to multi-period inventory management problems where demands are uncertain. On similar topics, José Alem and Morabito (2012) considered a production planning problem with uncertain costs and demands, while Wei et al. (2011) considered uncertain returns and demands. In Aouam and Brahimi (2013), the authors proposed a robust model for an integrated production planning problem. The RO approach have been extended to different source of uncertainty Varas et al. (2014), Thorsen and Yao (2017).

Thanks to the growing amount of data collected in industrial processes, the paradigm of Data-driven robust optimization (DDRO) have recently emerged with the objective to find new ways to use historical data as a support to build more sophisticated uncertainty sets. Several mathematical tools have been applied to incorporate a large fraction of the support of random parameters in the uncertainty set (Bertsimas and Brown, 2009; Bertsimas et al., 2018; Zhang et al., 2018). Chassein et al. (2019) have proposed a construction procedure of different class of uncertainty sets from real observations on a robust shortest path problem and compared their respective performance on a real-case application. Other approaches have addressed the exploitation of available data as an unsupervised learning problem, leading to the application of Machine Learning methods (Ning and You, 2019) to construct uncertainty sets. Among the different approaches proposed in the literature, Ning and You (2018) apply principal component analysis and kernel density estimation to construct uncertainty sets for different optimization problem encountered in the chemical industry. Multiple kernel support vector machine is applied in Han et al. (2021) for a multistage inventory management problem.

Among kernel learning approaches, Shang et al. (2017) proposed a support vector clustering (SVC) uncertainty set that efficiently captures correlations and asymmetries in the distribution of uncertain parameters. This technique produces polyhedral uncertainty sets, which offers a relative computational efficiency. This has been demonstrated by some recent studies such as a multi-product inventory management problem (Qiu et al., 2019), an energy system optimization (Shen et al., 2020) or a resource allocation in a cellular network (Wu et al., 2021). As another perspective on this technique, the recent work of Goerigk and Kurtz (2021) compare deep learning methods with the SVC-based method of Shang et al. (2017). As we shall see later, one drawback of this type of uncertainty
set comes from the number of variables and constraints introduced in the robust model. Indeed, these are linearly increasing in both the number of uncertain parameters and the size of the data set. In many industrial applications, this results in a large mathematical formulation that quickly become intractable.

In this paper we address this scalability issue by introducing two simple methods to approximate the SVC-based uncertainty set described in Shang et al. (2017). Both approaches rely on a parameter that directly controls the number of additional variables and constraints in the robust model. This allows us to compute solutions that are identical or comparable to those obtained with the SVC-based uncertainty set with a significant reduction in the computational burden.

The remaining of this paper is organized as follows: Section 2 describes the uncertainty set of Shang et al. (2017) and exhibits the reasons of the scalability issue. Section 3 describes in details the approximation methods. Section 4 presents the application of our approximations method on a robust production planning problem, before Section 5 concludes this study.

2. SUPPORT VECTOR CLUSTERING BASED UNCERTAINTY SET

Throughout the theoretical construction of the approximate SVC-based uncertainty sets, we consider that our goal is to solve a robust combinatorial optimization problems of the form

\[
\begin{align*}
\min & \quad c^\top x \\
\text{s.t.} & \quad a_j^\top x \leq b_j \quad \forall a_j \in \mathcal{U} \\
& \quad x \in \mathbb{N}^m
\end{align*}
\]

where uncertain parameters \(a_j\) belongs to a given uncertainty set \(\mathcal{U}\). Constraint (2) is equivalent to

\[
\max \quad a_j^\top x \leq b_j
\]

, where \(\mathcal{U}\) is constructed from a sample of data that consists of a set \(\mathcal{D} = \{u(1), \ldots, u(N)\}\) of \(N\) points in a \(m\)-dimensional space. Ideally, \(\mathcal{U}\) should provide a good representation of the support of the multi-dimensional random variable that generated \(\mathcal{D}\). Various Machine Learning techniques have been developed to achieve this goal. SVC (Ben-Hur et al., 2001) is one of them that relies on a technique that generates a representation of the support of the multi-dimensional random variable that generated \(\mathcal{D}\). On of the major strengths of SVC is its ability to naturally capture information such as the covariance of random variables or asymmetries in their distributions. In addition when the SVC algorithm is well designed, it does not need any sophisticated tuning of hyper-parameters and only requires to solve a quadratic program (QP).

Recently, Shang et al. (2017) have built data-driven uncertainty sets for robust optimization by applying the SVC approach with a custom piecewise linear kernel function (Weighted General Intersection Kernel, WGIK). The solution of their QP defines a subset \(\mathcal{B}\) of outliers called support vectors (SV), among which a subset \(\mathcal{B}_\nu\subseteq \mathcal{S}\), called boundary support vectors (BSV), are exactly on the boundary of the cluster. To summarize, those two sets are such that \(\mathcal{B} \subseteq \mathcal{S} \subseteq \mathcal{D}\) and \(|\mathcal{S} \setminus \mathcal{B}| \leq N\nu \leq |\mathcal{S}|\). We refer the reader to Shang et al. (2017) for more details on their method and the theoretical aspects of SVC. Figure 1 represents these sets for 1000 samples following a bivariate gaussian distribution when \(\nu = 0.15\).

To keep the remainder of this paper concise, we sometimes refer to a data point \(u(i)\) by its index \(i\) when it is clear from the context. For each point \(i \in \mathcal{D}\) the QP computes the vector of weights \(\alpha = [\alpha_1, \ldots, \alpha_N]\) where \(\alpha_i = 0\) iff \(i \in \mathcal{D} \setminus \mathcal{S}\) and \(\alpha_i = 1/\nu N\) iff \(i \in \mathcal{S} \setminus \mathcal{B}\). Shang et al. (2017) define the SVC based uncertainty set as

\[
\mathcal{U}_v(\alpha, \mathcal{D}) = \left\{ u \mid \sum_{i \in \mathcal{S}} \alpha_i ||Q(u - u(i))||_1 \leq \theta \right\}
\]

where ||·||_1 stands for the \(\ell_1\)-norm and

\[
\theta = \min_{\nu \in [0,1]} \left( \sum_{i \in \mathcal{S}} \alpha_i ||Q(u(i') - u(i))||_1 \right)
\]

Using auxiliary variables \(v = [v_1, \ldots, v_N]\), one can then reformulate \(\mathcal{U}_v(\alpha, \mathcal{D})\) as a polyhedron:

\[
\mathcal{U}_v(\alpha, \mathcal{D}) = \left\{ u \mid \exists v_i, \quad i \in \mathcal{S} \quad \text{s.t.} \quad \sum_{i \in \mathcal{S}} \alpha_i v_i^T 1 \leq \theta \quad -v_i \leq Q(u - u(i)) \leq v_i, \quad i \in \mathcal{S} \right\}
\]

which is bounded and nonempty for \(0 < \nu < 1\) (Shang et al., 2017). Based on formulation (7), the left-hand side of the robust linear constraint

\[
\max_{\alpha \in \mathcal{U}_v(\alpha, \mathcal{D})} a^T x \leq b
\]

is thus equivalent to the following LP:

\[
\begin{align*}
\max & \quad a^T x \\
\text{s.t.} & \quad \sum_{i \in \mathcal{S}} \alpha_i v_i^T v_i \leq \theta \\
& \quad -v_i \leq Q(a - u(i)) \leq v_i, \quad \forall i \in \mathcal{S}
\end{align*}
\]

Fig. 1. Representation of \(\mathcal{S}\) and \(\mathcal{B}\) for a bivariate gaussian distribution when \(\nu = 0.15\).
\[
\begin{align*}
\min & \sum_{i \in S} (\mu_i - \lambda_i)^T Q u^{(i)} + \eta \theta \\
\text{s.t.} & \sum_{i \in S} Q(\lambda_i - \mu_i) + x = 0 \\
\lambda_i + \mu_i &= \eta \alpha_i, \forall i \in S \\
\lambda_i, \mu_i &\in \mathbb{R}^n, \forall i \in S \\
\eta &\geq 0
\end{align*}
\] (12)

where \(\mu_i\) and \(\lambda_i\) are the modified weights that directly define the maximum number of strictly positive modified weight. The vector \(\hat{\alpha}\) is then computed using a linear program that maximizes the quality of the approximation with respect to \(\hat{S}\).

### 3.1 K-medoid based subset selection

From the construction of the SVC, we can assume that relatively close samples that lie strictly outside \(\mathcal{U}_b(\alpha, D)\) have similar contributions to cluster boundaries. Our first approach selects a subset \(\hat{S}\) of \(K\) data points, each representing the support vectors that are “close” to them. Our procedure relies on the so-called K-medoid clustering algorithm Kaufman and Rousseeuw (1990) to select the subset of points \(\hat{S}\) that aggregates the information contained in \(S\). We compute the distances between two points \(u, v \in \mathbb{R}^m\) using the \(\ell_2\)-norm \(||u - v||_2\). We define the set \(\hat{S}\) of selected support vectors as the union of the \(K\) cluster centers collected as the output of the algorithm with the set \(B\), as represented in Figure 2.

![Fig. 2. Subset \(\hat{S}\) of support vectors selected with the K-medoid approach](image)

### 3.2 Crown based subset selection

The idea motivating the “crown” method is to only keep the \(K\) support vectors that are closest to the cluster boundaries. We can achieve this selection easily by expressing \(K\) as a fraction \(\delta = K/N\) of the total number of points and compute a second SVC with parameter \((\nu - \delta)\). Let \(S'\) be the resulting set of support vectors: We simply define \(\hat{S} = S \setminus S'\) as the set of selected support vectors for the approximation. The set \(\hat{S}\) contains at most \(K\) of the original support vectors. Figure 3 illustrates the subset obtained for the example introduced above.

### 3.3 Updating coefficients \(\hat{\alpha}\)

The classification of point \((i')\) as being inside or outside of the SVC-based uncertainty set \(\mathcal{U}_b(\alpha, D)\) is based on the comparison of \(\theta(i')\) with a given threshold \(\theta\). A reasonable approach to approximate the \(\theta(i')\) thus consists in defining modified weights \(\hat{\alpha}\) in such a way that \(\hat{\theta}(i')\) is comparable to \(\theta(i')\) for any sample point \(i'\). This leads...
to the definition of a distance between $U_v(\alpha, D)$ and its approximation $\hat{U}_v(\hat{\alpha}, D)$ as:

$$
\Delta = \sum_{i' \in D} |\hat{\theta}(i') - \theta(i')|
$$

(20)

Obtaining the best approximation then comes down to computing weights $\hat{\alpha}$ that minimize $\Delta$, which can be formulated as the following LP:

$$
\min \Delta \\
\text{s.t. } \sum_{i' \in D} (\alpha_i - \hat{\alpha}_i) ||Q(u^{(i')} - u^{(i)})||_1 \leq \Delta \\
\sum_{i' \in D} (\hat{\alpha}_i - \alpha_i) ||Q(u^{(i')} - u^{(i)})||_1 \leq \Delta \\
\hat{\alpha}_i \geq 0 \\
\Delta \geq 0
$$

(21)  (22)  (23)  (24)  (25)

This LP enables us to derive the optimal weights $\hat{\alpha}$ and the approximate uncertainty set $U_v(\hat{\alpha}, D)$ is defined as:

$$
\hat{U}_v(\hat{\alpha}, D) = \left\{ u | \sum_{i' \in S} \hat{\alpha}_i ||Q(u - u^{(i')})||_1 \leq \theta \right\}
$$

(26)

where $\theta$ is the initial bound of (5).

4. APPLICATION TO AN ASSEMBLY PROBLEM

We consider the production planning problem introduced in Loger et al. (2021) where an assembly line combines a finite planning horizon $\mathcal{T}$. The set of final products and components are respectively denoted by $I$ and $J$. $d_{it}$ denotes the demand for product $i \in I$ faced by the system in each period $t \in \mathcal{T}$. For each product $i \in I$ and component $j \in J$, let $r_{ij}$ be the number of components $j$ required to produce one unit of product $i$. We assume that all components are available in a warehouse managed by a third party logistic provider (TPL). The TPL is in charge of delivering components to the assembly line according to its orders as represented on Figure 4. For each picking operation, an operator collects a given quantity of a single type of component, bounded by a maximum batch size of $m_j$ units. Thus several picking operations of the same component $j \in J$ may be scheduled in the same period due to this limitation. We presume that the picking time of a particular batch of component $j$ of size $x_j$ is composed of (1) a fixed time $p_j$, that corresponds to the travel time between the shipping point and the zone where components $j$ are stored and (2) a picking time per unit denoted $\tau_j$. The overall picking time of a given period $t \in \mathcal{T}$ is bounded by the maximum work capacity of the TPL $C_t$. Any demand for product $i$ that is not satisfied immediately is backlogged and incurs a backlogging penalty cost $b_i$ in each period until the corresponding product is assembled in a subsequent period.

Whenever a component $j$ is available on the assembly line but is not immediately used to manufacture an end product, it disturbs the production process by interfering with people and other goods moving nearby. We model this situation with a per-unit, per-period obstruction cost $\alpha_j$. The problem consists in planning the quantity of each component delivered to the assembly line by the TPL in each period such that the sum of the obstruction and backlogging costs is minimized.

To model this problem, we define variables $x_{jt}$ which represent the quantity of components $j \in J$ that should be delivered to the assembly line during period $t$. These variables are closely related to variables $v_{jt}$, representing the number of distinct picking operations of components $j \in J$ performed during period $t$. Variables $s_{jt}$ represent the number of components $j \in J$ held on the border of the assembly line at the end of period $t$. Variables $u_{jt}$ denote the number of final products $i \in I$ produced during period $t$ and finally, variables $P_t$ and $O_t$ denote the total backlogging penalty and obstruction costs incurred in period $t$, respectively.

We formulate the problem with the following MIP:

$$
\min \sum_{t=1}^{T} O_t + P_t
$$

(27)

s.t. $s_{jt} = s_{jt+1} + \sum_{k=1}^{t-1} x_{jk} - \sum_{i \in I} r_{ij} u_{ik} \forall j \in J, \forall t \in \mathcal{T} \setminus 1$

(28)

$$
\sum_{i \in I} u_{it} r_{ij} \leq s_{jt} + x_{jt} \forall j \in J, \forall t \in \mathcal{T}
$$

(29)

$$
\sum_{i \in I} u_{ik} \leq d_{it} \forall i \in I, \forall t \in \mathcal{T}
$$

(30)

$$
v_{jt} \geq x_{jt}/m_j \forall j \in J, \forall t \in \mathcal{T}
$$

(31)

$$
\sum_{j \in J} v_{jt} p_j + \tau_j x_{jt} \leq C_t \forall t \in \mathcal{T}
$$

(32)
The objective (27) aims at minimizing the total cost incurred over the planning horizon. The inventory balance constraints (28) update the stock levels in each period. Constraints (29) ensure that the quantity of component \( j \in J \) available in period \( t \) is sufficient to perform the planned assembly operations. Constraints (30) impose that the system never manufactures more units of product \( i \in I \) than the expressed demand. The minimum number of picking batches corresponding to the quantity of components \( j \) ordered in each period \( t \) is defined in constraints (31). Constraints (32) ensure that the total picking time does not exceed the picking capacity of the TPL provider in any period \( t \). Constraints (33) and (34) define the components holding costs and final product backlogging costs incurred in each period \( t \), respectively. Finally, constraints (35) and (36) define the domain of the decision variables.

4.1 Picking time uncertainties

We consider a case where the manufacturer has an incomplete or imprecise knowledge on the picking times. As a consequence, some combinations of his orders may exceed the picking capacity of the TPL provider, forcing the latter to postpone some operations to subsequent periods. We assume that we are given a set of historical setup times \( D = \{p^{(1)}, \ldots, p^{(N)}\} \) and present a comparison of the performance of three different robust models, each based on a particular implementation of \( \mathcal{U}(D) \). Namely, we compare the performances of the approximate SVC uncertainty set we propose built with the K-medoid method and the Crown method with both the original SVC of Shang et al. (2017) and a budget based uncertainty set similar to the Crown method with both the original SVC of Shang et al. (2017) and a budget based uncertainty set similar to the Crown method with both the original SVC of Shang et al. (2017) and a budget based uncertainty set.

4.2 Experimental results

Figure 5 and 6 presents the results obtained for instances with 5 final products over 10 periods and for five different number of components in \{(5, 10, 15, 20, 25)\}. The different uncertainty sets are built with a data-set \( D \) of \( N = 400 \) data-points and the solutions obtained are evaluated on a larger test data-set of \( N' = 10000 \) data points following the same distribution. For all uncertainty sets, we consider different values of parameters \( \Gamma \) and \( \nu \). The budget based solutions presented in Figure 5 and 6 always correspond to the best overall solutions obtained with this type of uncertainty set. The solutions presented for the SVC-based uncertainty set and the two approximations are always obtained with the same values of \( \nu \) and the additional parameter is set to \( K = 2 + |J| \).

Figure 5 clearly shows that SVC based model can help to reduce costs when in the case of asymmetric distributions. We observe that both the Crown and the K-medoid based approximations efficiently captured asymmetries in the distribution of picking time and lead to better solutions than the classic budget based uncertainty set. For a given value of \( \nu \), the solution obtained with the K-medoid based approximation tends to be closer to the solution obtained with \( \mathcal{U}_{(\alpha, D)} \) than those obtained with the Crown based approximation.

Figure 6 shows the solving time of the four different robust counterparts obtained for each instances. We observe for both approximation methods that restricting the number of support vector efficiently reduces the time needed to
solve the robust counterpart. On the instance with 25 components, the gap achieved with the SVC-based model exceeds 99% at the end of the maximum allocated time of 20 minutes. On the same instance, the K-medoid and the crown based approximations are respectively solved to optimality in 68 and 97 seconds. In our test set, the budget-based models always required less computational efforts than the SVC-based model and the two approximations, with a time difference that increases with the number of components.

5. CONCLUSION AND PERSPECTIVES

We propose two approximations of the SVC-based uncertainty set that efficiently reduce the computational effort while maintaining good quality solutions. Our main contribution is to make it possible to apply this type of data-driven approach to larger industrial problems in a reasonable amount time. Both methods lead to comparable computational time but the K-medoid based method produces a better approximation than the Crown based one. Future works could extend the K-medoid based approach by defining a specific metric space to compute the distance between two points in the K-medoid algorithm in order to improve the quality of the approximation. Using an optimization approach to select the best support vectors to define \( U_\ell(\hat{\alpha}, D) \) could also be an interesting perspective for future research work.

REFERENCES


