Optimal Voronoi Partitions for K-Adaptable Robust Optimization

Paula Weller

May 9, 2023

Advanced Equilibrium Modeling Course Project

1 Overview of Problem

Adjustable robust optimization originated in the idea of time stages: Some decisions must be made here-and-now, before an event with uncertain characteristics occurs, followed by another time stage of wait-and-see decisions. This order of events translates directly into levels within the model, creating a complex hierarchy of decision-making. K-adaptable robust optimization seeks to simplify that hierarchy by limiting the choices for the wait-and-see decisions. More specifically, the first stage specifies not only the here-and-now decisions, but also a selection of K actions for the second stage. After the uncertainty has been revealed, the decision maker can choose the best among these K options.

This is equivalent to partitioning the uncertainty set, i.e. the set of considered uncertainty scenarios, into K cells, each of which contains the scenarios associated with one of the K options. Although there are solution methods which do not require the explicit representation of this partitioning in the model [1] [3], it may be advantageous from a practical standpoint. Fram an explicit partitioning, the unfolding scenario can directly be attributed to a cell in the partitioning and therefore its optimal second-stage decision. This may even allow to overcome the incompleteness of data, as long as the corresponding cell can be determined. An implicit partitioning, in constrast, only provides the K recourse plans without connecting them to scenarios, thus making it necessary to determine the best among the K plans before acting.

Due to the exponential possibilities for this partitioning, solution algorithms in the literature usually rely on pre-fixed partitions, which are iteratively improved [2] [4] or adapted [5] in order to approximate the optimal partition. This leads to limitations concerning the flexibility of the cell architecture. The aim of this project is to explore an approach which doesn't rely on pre-fixed partitions but instead integrates the partition parameters as problem variables to be optimized. This approach, while computationally challenging, allows the decision-maker to draw upon a wider variety of optimal solutions, increasing solution quality in dimensions other than worst-case performance, such as observable average performance.

2 Mathematical Formulation

We consider as an example the K-adaptable emergency supply pre-allocation problem with linear constraints and a polyhedral uncertainty set. In essence, we have an amount b of emergency supplies which can be stored at n different supply points i with locations l_i. After the emergency has occurred, we know the demand levels at the m demand points j and can then allocate the supplies from the supply points to the demand points for a cost of c_{ij}. The complete formulation is as follows.

$$
\begin{aligned}
\min_{\mathcal{U}_1,\dots,\mathcal{U}_K} \min_{x,y,s} \; & z && && (1)\\
\text{s.t. } & z_k && \le z && k=1,\dots,K && (2)\\
& p\sum_{j=1}^{m} s_j^k + \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} y_{ij}^k && \le z_k && k=1,\dots,K && (3)\\
& \sum_{j=1}^{m} y_{ij}^k && \le x_i && i=1,\dots,n,\ k=1,\dots,K && (4)\\
& \sum_{i=1}^{n} x_i && \le b && && (5)\\
& \sum_{i=1}^{n} y_{ij}^k + s_j^k && \ge \xi_j^k && j=1,\dots,m,\ k=1,\dots,K && (6)\\
& \xi_j^k \in S_j(\mathcal{U}_k) && && && (7)\\
& \mathcal{U} = \mathcal{U}_1 \cup \dots \cup \mathcal{U}_K && && && (8)\\
& x \in \mathbb{N}^n,\ y^k \in \mathbb{N}^{n\times m},\ s^k \in \mathbb{N}^m. && && && (9)
\end{aligned}
$$

The formulation aims to minimize

$$
z = \max_{k=1,\dots,K} z_k,
$$

where z_k is the worst-case objective for the k-th cell in the partition. The variables of the problem are the numbers of supplies x_i stored at each supply point i, the allocation of the supplies from the supply points to the demand points according to plan k, y_{ij}^k, slack variables s_j^k to account for unsatisfied demand, and, most importantly, the cells of the partition $\mathcal{U}_1,\dots,\mathcal{U}_K$. Constraint (3) sets z_k, the worst-case objective of cell k, to be the sum of transport costs $c_{ij}y_{ij}^k$ as well as a penalty p for every unit of unsatisfied demand. Constraint (4) ensures that a supply point only provides as many supplies as are stored there according to the first-stage decision x. Constraint (5) bounds the aggregated amount of supplies, and (6) ensures the satisfaction of ξ_j^k, the worst-case demand at demand point j in cell k. This worst-case demand is obtained from the subproblem

$$
\begin{aligned}
S_j(\mathcal{U}_k) = \max\nolimits_{\xi} \quad & \xi_j \\
\text{s.t.} \quad & \xi \in \mathcal{U}_k
\end{aligned}
\tag{10}
$$

and ensures that the solution is feasible for all scenarios in the uncertainty set.

The complexity of problem (1)-(9) lies within constraint (8), which ensures that the cells $\mathcal{U}_1,\dots,\mathcal{U}_K$ form a cover of the uncertainty set $\mathcal{U}$. The partitioning condition $\mathcal{U}_k \cap \mathcal{U}_l = \emptyset$ can be relaxed, since having several possible recourse actions for one scenario in $\mathcal{U}$ does not impact feasibility.

To integrate the sets $\mathcal{U}_k$ as problem variables, we must find a suitable representation. Inspired by Bertsimas and Dunning [2], we use Voronoi Diagrams to represent partitions. Finitely representable Voronoi diagrams can not form arbitrary shapes, but they are able to represent any polyhedral shape, and add the advantage of comparability to the method of [2].The diagrams are represented in the form of Voronoi points $v_1, \dots, v_K$. Each point is the center of the corresponding cell, such that we have

$$\mathcal{U}_k = \{\xi \in \mathcal{U} \mid (v_l - v_k) \cdot \xi \le (v_l - v_k) \cdot \frac{1}{2}(v_l + v_k) \quad \forall l = 1, \dots, K,\ l \ne k\}. \tag{11}$$

By assuming a Voronoi partition, we can add the Voronoi-type constraints in the lower level, where $\mathcal{U}_k$ occurs, and omit constraint 8. Then, by including the Voronoi points as upper-level decision variables, we enable the model to optimize the partition. The lower level thus becomes

$$\begin{aligned} S_j^k(v) = \max_{\xi} \quad & \xi_j \\ \text{s.t.} \quad & \xi \in \mathcal{U} \\ & (v_l - v_k) \cdot \xi \le (v_l - v_k) \cdot \tfrac{1}{2}(v_l + v_k) \quad \forall l = 1, \dots, K,\ l \ne k \end{aligned} \tag{12}$$

The lower level is linear in ξ and therefore convex, making it possible to use the KKT conditions for reformulation to a single-level problem. However, there is an even simpler solution: Since the lower-level variable ξ does not appear in the objective and is thus only relevant to the feasibility of the upper-level solution, we can use a duality scheme for reformulation and the relax lower-level optimality as in the following.

First, we specify the uncertainty set $\mathcal{U}$.

$$\mathcal{U} = \{\xi \in \mathbb{R}^m : \ |\xi_j - \xi_{j'}| \ \le \ ||\, l_j - l_{j'} \,||_\infty, \qquad 1 \le j, j' \le m. \tag{13}$$

$$\sum_{j=1}^{m} \xi_j \ \le \ b\}. \tag{14}$$

Constraint (13) ensures that closely located demand points share similar demands, to introduce the notion of locality to the emergency, and constraint (14) bounds the aggregated demand to match the number of available supplies. This excludes unrealistic scenarios wherein all demand points are highly affected. Linearizing (13) and then dualizing the problem, strong duality dictates that

$$S_j^k(v) = \min_{\alpha,\beta,\gamma} f_j^k(\alpha,\beta,\gamma) \tag{15}$$

$$\text{s.t.} \quad \sum_{\substack{j_2=1 \\ j_2 \ne j_1}}^{m} (\alpha_{j,k}^{j_1,j_2} - \alpha_{j,k}^{j_2,j_1}) + \beta_{j,k} + \sum_{\substack{l=1 \\ l \ne k}}^{K} (v_{l,j_1} - v_{k,j_1})\gamma_{j,k}^{l} \ge \begin{cases} 1 & \text{if } j_1 = j \\ 0 & \text{otherwise} \end{cases} \qquad j_1 = 1, \dots, m \tag{16}$$

$$\alpha, \beta, \gamma \ge 0 \tag{17}$$

where

$$f_j^k(\alpha,\beta,\gamma) = \sum_{j_1=1}^{m} \sum_{\substack{j_2=1 \\ j_1 \ne j_2}}^{m} ||\, l_{j_1} - l_{j_2} \,||_\infty \alpha_{j,k}^{j_1,j_2} + \beta_{j,k} \cdot b + \sum_{\substack{l=1 \\ l \ne k}}^{K} \frac{1}{2}(v_l - v_k)(v_l + v_k)\gamma_{j,k}^{l}. \tag{18}$$

We therefore have for any feasible (i.e., satisfying (16)) $\alpha, \beta, \gamma \geq 0$, and any $\xi_j^k \in S_j^k(v)$, that

$$\xi_j^k \leq f_j^k(\alpha, \beta, \gamma). \tag{19}$$

Since ξ_j^k does not appear in the objective of problem (1)-(9), we can therein replace ξ_j^k by $f_j^k(\alpha, \beta, \gamma)$ in constraint (6), add constraint (16) to the problem, and thus guarantee feasibility of the upper-level decisions without solving a lower-level problem. The final problem is

$$
\begin{aligned}
\min_{x,y,s,v,\alpha,\beta,\gamma} \quad & z && && (20)\\
\text{s.t.} \quad & z_k && \leq z && k = 1, \ldots, K && (21)\\
& p \sum_{j=1}^{m} s_j^k + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} y_{ij}^k && \leq z_k && k = 1, \ldots, K && (22)\\
& \sum_{j=1}^{m} y_{ij}^k && \leq x_i && i = 1, \ldots, n,\ k = 1, \ldots, K && (23)\\
& \sum_{i=1}^{n} x_i && \leq b && && (24)\\
& \sum_{i=1}^{n} y_{ij}^k + s_j^k && \geq f_j^k(\alpha, \beta, \gamma) && j = 1, \ldots, m,\ k = 1, \ldots, K && (25)\\
& \sum_{\substack{j_2=1\\ j_2 \neq j_1}}^{m} (\alpha_{j,k}^{j_1,j_2} - \alpha_{j,k}^{j_2,j_1}) + \beta_{j,k} + \sum_{\substack{l=1\\ l \neq k}}^{K} (v_{l,j_1} - v_{k,j_1}) \gamma_{j,k}^l && \geq \begin{cases} 1 & \text{if } j_1 = j \\ 0 & \text{otherwise} \end{cases} && j_1 = 1, \ldots, m && (26)\\
& \sum_{j=1}^{m} (v_{k,j} - v_{l,j})^2 && \geq 0.1 && 1 \leq k < l \leq K && (27)\\
& x \in \mathbb{N}^n,\ y^k \in \mathbb{N}^{n \times m},\ s^k \in \mathbb{N}^m,\ \alpha, \beta, \gamma \geq 0. && && && (28)
\end{aligned}
$$

We add constraint (27) to ensure that the Voronoi points are pairwise distinct. Although now single-level and continuous, this model is not easily solvable. In $f_j^k(\alpha, \beta, \gamma)$, we have triples of variables being multiplied with each other: $(v_l - v_k)(v_l + v_k)\gamma_{j,k}^l$. This makes the problem nonconvex.

3 Data Sources

A problem instance consists of the locations l of service and demand points, and the number of available supplies b. The allocation cost we calculate from the locations l to be the distance between the corresponding locations, measured by $||\cdot||_\infty$, e.g. as an estimate of response time.

Due to the complexity of the model and for the purpose of illustration, we consider very small instances with $n = 1, m = 2$. This way, the two-dimensional uncertainty set can be easily depicted. Locations are randomly generated on a grid of $0, \ldots, 5 \times 0, \ldots, 5$ such that no two locations coincide. We set the number of available supplies to be 5 and the penalty for unsatisfied demand to be $p = 100$ to motivate the model to prioritize the satisfaction of demand over saving allocation costs.

(a) Instance I_1 (b) Instance I_2

Figure 1: Example Instances

4 Analysis

We first inspect instance I_1, where the service point is located at $[1, 3]$ and the demand points at (2 2) and (4 3), respectively, as illustrated in Figure 1. The model is solved using Gurobi and Julia, the corresponding code can be viewed in the GitHub Repository [6]. The solving process does not terminate and is therefore automatically stopped after 120s, with the incumbent remaining unchanged after the first 30s. The resulting objective is $z_{I_1} = 110.2$, where the main contribution to the objective is given by the penalty for unsatisfied demand. For I_1, the solution is

k	y	s	ξ	v
1	(2.5, 2.5)	(0, 1)	(2.5, 3.5)	(22, 38)
2	(3.5, 1.5)	(0, 1)	(3.5, 2.5)	(38, 22)

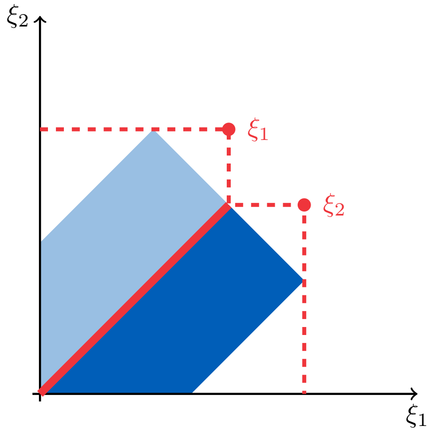
This is illustrated in Figure 2a. The uncertainty set is divided symmetrically into two parts, which have mirrored worst-cases ξ_1 and ξ_2. Preparing for these worst cases guarantees preparation for the whole uncertainty set, as indicated by the dashed lines. Thus, the allocations y_1 and y_2 with the respective slacks cover the demand for their respective worst case.

Even though the model was not solved to verifiable optimality, it is easy to see that this is an optimal partition. The objective is mainly influenced by the expensive slack variables, which must pick up all unsatisfied demand. The unsatisfied demand, in turn, is obtained via the worst-case scenarios

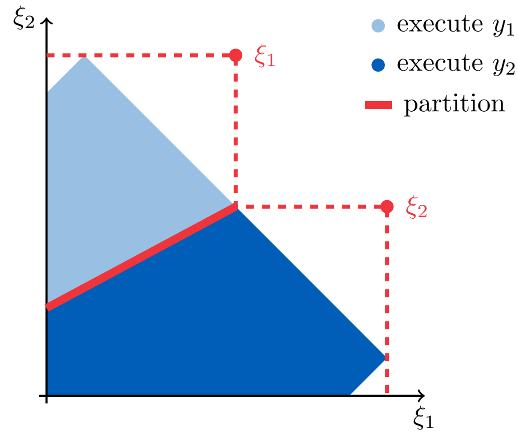
$$\xi^k = \{(\xi_1^k, \xi_2^k) \mid \xi_j^k = \max_{\xi \in \mathcal{U}_k} \xi_j,\ j = 1, 2\} \tag{29}$$

which outline a box around their respective cells $\mathcal{U}_k$. The only way to reduce a slack variable is to reduce that box, but in this case, reducing one box means increasing the size of the other, which then increases the corresponding slack variable and thus the worst-case objective. Therefore, the optimal partitioning assigns equally large boxes to both cells. This partitioning scheme coincides with the one [2] produces, where the Voronoi points are $v_1 = (3.5, 1.5)$ and $v_2 = (1.5, 3.5)$.

Note, that the slack variables are only non-negative for the demand node at the location $(4, 3)$, never for the other demand node. The reason for this is the allocation cost: $(4, 3)$ is further away from the supply node, thus by using the slack in this node, the model can at least save allocation costs.



(a) I_1



(b) I_2

Figure 2: Optimal Partitionings of the Uncertainty Set

A more interesting result is produced by instance I_2 with supply location [3 1] and demand locations $(1\ 1), (4\ 5)$. As illustrated in Figure 1, the main difference between the two instances is the distance between the demand nodes. For I_2, we obtain $z_{I_2} = 213.2$ as well as the following results:

k	y	s	ξ	v
1	(2.5, 2.5)	(0, 2)	(2.5, 4.5)	(19.4,29.5)
2	(4.5,0.5)	(0, 2)	(4.5, 2.5)	(34.3,0)

Here, we observe that the expensive slack variables have values up to 2, which is due to the larger uncertainty set: Since the demand points are located further away from each other, their demands are allowed to differ more, i.e., the locality constraint (13) is less restrictive. The more noteworthy difference, however, lies with the voronoi points. While the worst-case demand scenarios ξ still mirror each other, the voronoi points do not. As a consequence, the partitioning is not symmetric, contrary to the one from [2], which would resemble the one in I_1.

This does not impact feasibility since the affected scenarios are within both boxes and can therefore be attributed to either in terms of feasibility. Neither does it affect the overall objective value, which is only influenced by the worst-case box scenarios ξ_1 and ξ_2. The assignment of these scenarios may well, however, influence the performance of the solution in the case of these non-worst-case scenarios, observable through simulations or the average observable performance.

The algorithm in [2], even though in this case it would be able to match the objective of our algorithm, would not enable the model to accommodate these non-worst case scenarios. With respect to the worst case, both partitions are optimal. However, consider, for instance, the scenario $\hat{\xi} = (0, 0.5)$. Then, we have for transportation cost $c = (2, 4)$:

$$
\begin{aligned}
z(\hat{\xi}, y_1) &= 100 \cdot \sum_{j=1}^{2} \max\{0, \hat{\xi}_j - y_{2,j}\} \ + \ 2.5c_1 + 2.5c_2 && = 15 \\
z(\hat{\xi}, y_2) &= 100 \cdot \sum_{j=1}^{2} \max\{0, \hat{\xi}_j - y_{1,j}\} \ + \ 4.5c_1 + 0.5c_2 && = 11,
\end{aligned}
$$

making y_2 the better plan for this scenario. A fixed partition, designed to minimize the worst-case objective, does not grant the flexibility to accommodate this insight into the solution. It thus could easily attribute many scenarios to a suboptimal second-stage plan, thereby neglecting the usefulness of an explicit partitioning.

However, granting this flexibility to the model does not automatically translate to the model making use of it. Consider $\overline{\xi} = (0, 1)$. Here, we have

$$
\begin{aligned}
z(\overline{\xi}, y_1) &= 15 \\
z(\overline{\xi}, y_2) &= 61,
\end{aligned}
$$

since $y_{2,2} = 0.5$ does not satisfy the demand $\overline{\xi}_2 = 1$. But, as visible from Figure 2b, our solution maps $\overline{\xi}$ to decision y_2, which is clearly suboptimal. Thus, even though the model has the power to seek this better solution, in this case, it does not. In order to motivate the model, one needs to include a corresponding incentive, which opens an avenue for future research.

5 Conclusion

Due to the exponential amount of partitioning possibilities, previous solution approaches in the literature resort to iteratively constructing or improving partitions which optimize the worst-case behavior. This usually involves fixing partitions first, and then optimizing the remaining variables. This research has given to the model the power of managing itself the partitioning and thereby the architecture of its lower levels.

Unfortunately, this leads to a nonconvex formulation even for continuous variables, which is difficult to solve. Due to this computational issue, the model could not be solved to optimality. The solution was able to match previously proposed algorithms with respect to solution quality, but with a much longer computation time. In addition, the proposed conversion of the bilevel formulation to a single-level problem prevents the use of integer recourse variables.

The currently evident value of this approach is more theoretical: It offers new perspectives on the partitioning schemes used by previously proposed algorithms, and can be used for comparisons and critical examinations. Its solutions have, as this research has revealed, the potential to be tuned to average performance as well as worst-case ones, offering a wider range of optimal robust solutions.

To fulfill this potential, next steps include further refinement and testing of the approach: How can the model be incentivized to pursure additional criteria? How does it perform in combination with larger instances, different model parameters and uncertainty sets, and can it be adapted for other applications? In addition, an attempt to resolve or mitigate the computational issues seems promising, for example by investigating possibilities to make the solving process faster and more efficient. Overall, the project is not concluded with a new, practical solution method, but with a valuable insight into the possibilities and limitations of K-adaptability, and, as always, a new abundance of open questions.

References

[1] D. Bertsimas and C. Caramanis. Finite adaptability in multistage linear optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, 2010.

[2] D. Bertsimas and I. Dunning. Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research*, 64(4):980–998, 2016.

[3] G. A. Hanasusanto, D. Kuhn, and W. Wiesemann. K-adaptability in two-stage robust binary programming. *Operations Research*, 63(4):877–891, 2015.

[4] K. Postek and D. den Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28:553–574, 07 2016.

[5] A. Subramanyam, C. E. Gounaris, and W. Wiesemann. K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation*, 12(2):193–224, Jun 2020.

[6] P. Weller. Optimal partitions project. `https://github.com/paulaweller/OptimalPartitionsProject`, 2022.

[7] İhsan Yanıkoğlu, B. L. Gorissen, and D. den Hertog. A survey of adjustable robust optimization. *European Journal of Operational Research*, 277(3):799–813, 2019.