European Journal of Operational Research 324 (2025) 925–940

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Decision support

Streamlining emergency response: A K-adaptable model and a column-and-constraint-generation algorithm ☆

Paula Weller, Fabricio Oliveira *

Department of Mathematics and Systems Analysis, Aalto University, Otakaari 1, Espoo, 02150, Finland

ARTICLE INFO

Keywords:
Robust optimization
K-adaptability
Emergency response
Bilevel optimization

ABSTRACT

Emergency response refers to the systematic response to an unexpected, disruptive occurrence such as a natural disaster. The response aims to mitigate the consequences of the occurrence by providing the affected region with the necessary supplies. A critical factor for a successful response is its timely execution, but the unpredictable nature of disasters often prevents quick reactionary measures. Preallocating the supplies before the disaster takes place allows for a faster response, but requires more overall resources because the time and place of the disaster are not yet known. This gives rise to a trade-off between how quickly a response plan is executed and how precisely it targets the affected areas. Aiming to capture the dynamics of this trade-off, we develop a K-adjustable robust model, which allows a maximum of K second-stage decisions, i.e., response plans. This mitigates tractability issues and allows the decision-maker to seamlessly navigate the gap between the readiness of a proactive yet rigid response and the accuracy of a reactive yet highly adjustable one. The approaches we consider to solve the K-adaptable model are twofold: Via a branch-and-bound method as well as a static robust reformulation in combination with a column-and-constraint generation algorithm. In a computational study, we compare and contrast the different solution approaches and assess their potential.

1. Introduction

In 2019, a year marked by temperature records and severe, large-scale wildfires across the globe, disasters related to natural hazards affected a total of 95 million human lives worldwide, injuring them and depriving them of their homes and livelihoods (CRED, 2020). In 2021, this total increased to 102 million people (CRED, 2022). In 2022, the number of affected people reached a staggering 185 million (CRED, 2023). Amid the growing momentum of climate change, it is anticipated to increase even further (CRED & UNISDR, 2020). According to the 2021 IPCC report, (Masson-Delmotte, Zhai, Pirani, Connors, Péan, Berger, Caud, Chen, Goldfarb, Gomis, Huang, Leitzell, Lonnoy, Matthews, Maycock, Waterfield, Yelekçi, Yu, , & Zhou, 2021), the number of 10-year (that is, once every ten years) heavy precipitation events is set to at least double, the number of 10-year hot temperature extremes over land is predicted to more than quadruple, and events which would have occurred once in fifty years without human influence are likely to occur up to 40 times in the event of a 4-degree temperature change. On average, 90 percent of the natural hazards which occurred during the last twenty years were sudden onset events, such as wildfires, storms or floods (CRED, 2020, 2022, 2023). Due to their highly irregular nature, these events are usually met unprepared, and the response operations are characterized by a challenging combination of uncertainty, complexity, and urgency. In such situations, a short response time is not a matter of money, but rather of life and death. Still, a swift response is often obstructed due to unreliable data, inefficient logistics, and, most importantly, a lack of preparation (Van Wassenhove, 2006). Preparation allows for shifting the most time-consuming processes, i.e., decision-making and organizing response actions, to the time before the contingency, thus allowing for an efficient and streamlined response (Alem, Bonilla-Londono, Barbosa-Povoa, Relvas, Ferreira, & Moreno, 2021). The difficulty lies in balancing the preparation efforts to accommodate the uncertainty: The more rigid the response, the more efficiently it can be prepared and executed, but the less accurate it will be with respect to the disaster scenario, which is not known during the preparations.

Traditional models for optimization under uncertainty are not able to support this balancing act, because they either fail to model the risk averseness required in connection to emergency response or correspond to either one of two extremes. The first extreme is a so-called *static model* wherein all decisions are made here and now. This results in

☆ We gratefully acknowledge the funding provided by the Wihuri Foundation as well as the computational resources provided by the Aalto Science-IT project.
* Corresponding author.
E-mail address: fabricio.oliveira@aalto.fi (F. Oliveira).
URL: https://www.aalto.fi/en/people/fabricio-oliveira (F. Oliveira).

https://doi.org/10.1016/j.ejor.2025.02.016
Received 13 March 2024; Accepted 14 February 2025
Available online 24 February 2025

the same decision behavior regardless of the sequentially discovered information and corresponds to a response which is preparationally efficient, but too rigid for practical use. In addition, having to satisfy all possible futures simultaneously, static solutions are by definition not able to take advantage of the strong geographical correlations (demand being concentrated around the area of impact, as opposed to evenly distributed across a whole region or country) which are typically present in the uncertainty connected to natural disasters (Alexander, 1993). The second extreme is a fully adaptable model, which assigns to every considered possible future a different decision. This results in a behavior equivalent to making decisions later, when all information is available. Thus, a corresponding response would be highly accurate and, due to its flexibility, able to exploit correlations between uncertain parameters. However, preparing individual plans for possibly infinite amounts of futures is not financially and organizationally sensible.

By using a K-adaptable model, our proposed approach can limit the amount of required preparations while at the same time offering flexibility and taking advantage of correlations in the uncertainty. A robust K-adaptable model partitions the set of possible scenarios into at most K subsets, for each of which it subsequently determines the optimal recourse decision in a robust manner, offering a finite amount of response plans. It thus offers the possibility of navigating between the two extremes, allowing the decision-makers to exploit the balance between efficiency and accuracy according to their own preferences.

The concept of finite adaptability has enjoyed attention in the literature in both its robust (i.e., optimizing the worst-case) and its stochastic (i.e., optimizing the expected value) form, mainly due to two reasons. Firstly, the notion of selecting a small number of decision alternatives to accommodate future uncertainties is more faithful to the human decision-making process (Buchheim & Pruente, 2019). Secondly, K-adaptable formulations mitigate tractability issues for problems which are small in some dimensions, such as the dimension of the uncertainty set (in the robust case) or the number of constraints in which the uncertainty is present. Nevertheless, in general, the K-adaptable problem is NP-hard even for $K = 2$ (Bertsimas & Caramanis, 2010; Buchheim & Kurtz, 2017; Malaguti, Monaci, & Pruente, 2022). Aiming to circumvent this complexity issue, the literature offers several algorithmic approaches, especially for the robust version, which due to its risk-averse nature is the one we consider here. Most existing solution algorithms are subject to restrictive assumptions such as binary decision variables (Buchheim & Kurtz, 2018; Ghahtarani, Saif, Ghasemi, & Delage, 2023; Hanasusanto, Kuhn, & Wiesemann, 2015, 2016; Kurtz, 2023), requiring "not too many good solutions" (Arslan, Poss, & Silva, 2022), or a restriction concerning the number of recourse decisions K (Chassein, Goerigk, Kurtz, & Poss, 2019). Bertsimas and Dunning (2016) as well as Postek and Hertog (2016) developed algorithms which iteratively split the uncertainty set, gradually increasing K in each iteration. While computationally tractable due to the procedures' approximative nature, as a downside, the exact value of K cannot be chosen beforehand. The number of algorithms in the literature allowing the decision-maker to fix K beforehand without any additional assumptions is remarkably small. Subramanyam, Gounaris, and Wiesemann (2020) proposed a branch-and-bound scheme which solves the K-adaptable problem to optimality for a given K. In Section 3.2, we describe how their method can be applied to the pre-allocation problem considered here. Han, Bandi, and Nohadani (2022) developed a partitioning scheme based on translated orthants, which permits a reformulation of the K-adaptable problem to an equivalent static model with an exponential number of constraints. This reformulation is based on the same principle as the reformulation we make use of in Section 3.3. However, their solution approach requires an additional bounding assumption to overcome the exponential number of constraints, whereas our proposed approach is akin to a column-and-constraint generation method.

Despite these developments, to the best of our knowledge, finite adaptability has never been applied to the subject of emergency response. Sabbaghtorkan, Batta, and He (2020) offer a comprehensive overview of the literature with a focus on the pre-disposition of emergency supplies. They also identify several gaps within the current research, most notably the need for more research on considering pre-positioning as a risk mitigation strategy. In addition, most papers neglect to limit the number of supplies that are available for storage, thus perhaps not sufficiently motivating the storage network to be flexible.

As a methodology, the K-adaptable approach to emergency response can address several previously faced issues. Firstly, having a limited number of plans offers mitigation of the tractability issues surrounding fully adaptable models. K-adaptable problem formulations, while theoretically challenging, offer in practice many tractable ways of obtaining near-optimal solutions.

Secondly, a small set of K optional response actions facilitates the preparation and organization of the response while retaining the flexibility necessary to prepare for an uncertain event. Since conditions within a disaster situation are usually highly dependent (for example, demand for emergency supplies correlates with proximity to the disaster site), only a small amount of flexibility is sufficient for an efficient response.

Furthermore, being able to choose the number of plans K allows the decision-maker to control the trade-off between efficiency and effectiveness of the response, as the larger the number of plans, the more targeted and effective, but also complex and resource-intensive the response becomes.

Lastly, the K-adaptable approach shifts the time-intensive decision-making process of the second stage into the first stage, offering a guaranteed-to-be-feasible distribution plan for the relief items *ahead of time*. All of the previously mentioned robust approaches for emergency response use robustness as a warding measure. They optimize with respect to the worst case, without actually assuming this case to be realized, but simply ensuring that a distribution plan can be created after the actual data are known. None of them explores preemptively offering this plan, allowing for a rehearsal of the response, for a clear plan of action ahead of time, and, most importantly, for an improved response time. Therefore, we posit that K-adaptability, as a methodology for emergency response, provides a new modeling perspective better aligned with disaster response decision-making.

In light of these qualifications, the contributions of this paper are as follows.

1. Formulate a K-adaptable model for the pre-allocation of emergency supplies.
2. Propose a new reformulation and column-and-constraint generation algorithm for the K-adaptable model and compare it with a state-of-the-art algorithm from the literature.
3. Analyze the two algorithms with respect to computational effort and solution quality.
4. Investigate the potential of K-adaptability for the application of emergency response using both synthetic and real-world-based instances.

The remainder of the paper is structured as follows: In Section 3.1, we formulate the K-adaptable pre-allocation problem. Section 3.2 relays a version of the branch-and-bound method proposed by Subramanyam et al. (2020) adapted to the pre-allocation model. Section 3.3 introduces an equivalent reformulation of the pre-allocation model and a newly proposed column-and-constraint generation (CCG) solution algorithm based on this reformulation, the Box CCG algorithm. In Section 4, we study the application of both algorithms to the pre-allocation model and analyze their performance for both a synthetic data set as well as a case study relating to floods in the state of Rio de Janeiro, Brazil. Finally, Section 5 concludes the paper with some final observations and outlooks.

2. Background

The most challenging aspect of disaster response is the uncertainty accompanying natural hazards. Due to their irregular nature, the time, place and severity of the disaster are extremely difficult to predict (Tippong, Petrovic, & Akbari, 2022). In addition, even after the disaster has occurred, available information is frequently unreliable, incomplete, or non-existent. For a successful response management, it is therefore crucial to consider carefully the integration of this element of uncertainty.

In general, to account for uncertainty in mathematical programming models, the literature offers two popular approaches: Stochastic programming (Birge & Louveaux, 2011) and robust optimization (Ben-Tal, El Ghaoui, & Nemirovski, 2009).

The majority of literature on the subject of emergency response employs stochastic programming (SP) (Caunhye, Nie, & Pokharel, 2012; Sabbaghtorkan et al., 2020), which uses probability distributions to represent uncertain parameters in the model. However, probabilistic information is reliably accurate only for large amounts of data. When modeling disaster situations, unreliable, incomplete, and non-existent data may thus compromise the relevance of the model. In addition, the model-predicted quality of solutions optimized for stochastic performance is only achieved over a similarly large amount of implementations. As singular occurrences, natural disasters do not meet this profile. Probability-based solutions are therefore very likely to fail to achieve the predicted quality in reality. Duran, Gutierrez, and Keskinocak (2011), for example, develop a mixed-integer programming (MIP) inventory-location model for emergency response supplies, such as water or medicine. The network is configured to minimize the average response time over a set of historical data, optimizing the likelihood of a good performance. However, this solution does not offer any protection against risk since the dispersion does not need to comply with any binding constraints such as a certain, short response time. This illustrates how SP solutions are unable to provide a decisive course of action in the form of a disaster response prior to the observation of the uncertain event.

2.1. Robust optimization

A more suitable approach for the purpose of risk aversion is robust optimization (RO), which measures the quality of a solution by its performance in the worst of all considered outcomes. In general, RO does not rely on a probability distribution, but rather on a set of possible outcomes, the so-called uncertainty set $\mathcal{U} \subseteq \mathbb{R}^m$, within which lie all the realizations that the solution must be robust towards. In its classical form, this framework places all decisions $x \in \mathbb{R}^n$ before the uncertain event and provides a solution which is feasible for all scenarios $\xi \in \mathcal{U}$. For the sake of simplicity, we only consider problems with right-hand side uncertainty, but the generalization to constraint and objective uncertainty is straightforward (Ben-Tal et al., 2009). For $c \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m\times n}$, the so-called static robust problem with right-hand side uncertainty considered here is

$$\min_x \; c^T x \tag{1a}$$

$$\text{s.t.} \;\; Ax \geq \xi \qquad \forall \xi \in \mathcal{U}. \tag{1b}$$

Static robust problems retain the complexity of their deterministic version and provide a single solution which is robust towards all elements in the uncertainty set $\mathcal{U}$. However, because these solutions are independent of the uncertainty realization, they automatically neglect any information related to correlations among the uncertain parameters, which, in contrast, we assume to be present here. Consider, for example, a simple budgeting problem

$$\min_x \; x_1 + x_2 \tag{2a}$$

$$\text{s.t.} \;\; x_1 \geq \xi_1 \qquad \forall \xi \in \mathcal{U} \tag{2b}$$

$$x_2 \geq \xi_2 \qquad \forall \xi \in \mathcal{U}, \tag{2c}$$

where $\mathcal{U} = \{(\xi_1, \xi_2) \in \mathbb{R}^2_+ \mid \xi_1 + \xi_2 \leq 1\}$, that is, the uncertain parameters ξ_1 and ξ_2 are correlated. The optimal static robust solution must accommodate all possibilities for ξ simultaneously and is, therefore, $x = (1, 1)$, even though the corresponding scenario $\xi = (1, 1)$ is not an element of the uncertainty set. Thus, when using static robust models, the solution is often overly conservative, as it is in fact robust towards a set much larger than $\mathcal{U}$ (see Fig. 1(a)). The static model accurately reflects the uncertainty set only if the uncertainties are uncorrelated.

To take advantage of the correlations between the components within the uncertainty set, the model must include a post-disaster decision stage, wherein it does not have to ward against all possibilities simultaneously. To this end, Ben-Tal, Goryashko, Guslitzer, and Nemirovski (2004) have extended the classical robust framework to accommodate recourse decisions $y(\xi)$, $\xi \in \mathcal{U}$, which are decisions that are made after the uncertainty has been revealed. They depend on the realization of the uncertainty $\xi \in \mathcal{U}$, therefore making them mappings $y : \mathcal{U} \to \mathbb{R}^n$. This results in a two-stage robust model, also referred to as adjustable robust optimization (ARO). Formally, the incorporation of recourse decisions into the robust framework gives rise to the adjustable robust counterpart (ARC), or completely adaptable problem, where for the variables $x, y(\xi) \in \mathbb{R}^n$ for all $\xi \in \mathcal{U}$, and $z \in \mathbb{R}$, as well as the parameters $c, d \in \mathbb{R}^n$; $A, B \in \mathbb{R}^{m\times n}$ we have

$$\min_{x,y,z} \; z \tag{3a}$$

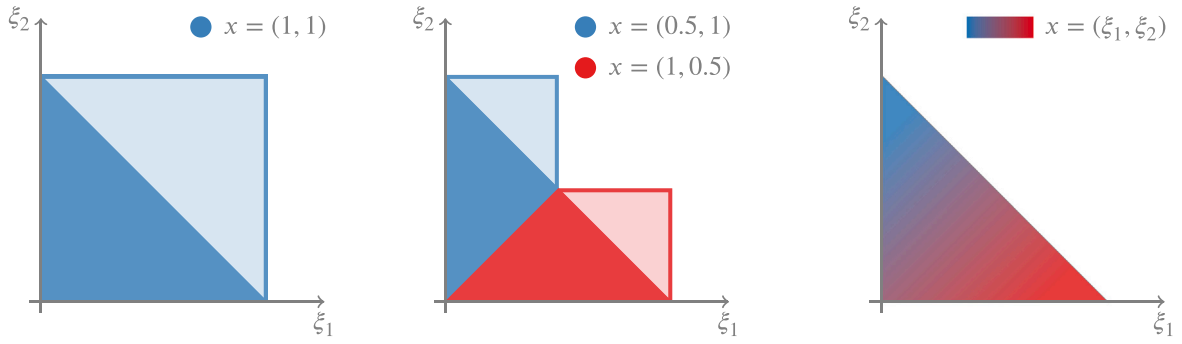
$$\text{s.t.} \;\; c^T x + d^T y(\xi) \leq z \qquad \forall \xi \in \mathcal{U} \tag{3b}$$

$$Ax + By(\xi) \geq \xi \qquad \forall \xi \in \mathcal{U}. \tag{3c}$$

The variable $z \in \mathbb{R}$ measures the maximum (worst-case) value of the objective function with respect to the uncertainty. By choosing the functions y to be constant, one can establish the classical RO formulation without recourse as a special case of the adjustable one. As such, the adjustable problem always performs at least as well as the static problem with respect to the objective value. In the general case, however, the adjustable solution cannot be guaranteed to perform better than the static one.

ARO has already seen successful applications in emergency response. Safaei, Farsad, and Paydar (2017) developed a robust two-stage model to optimize the flow of relief commodities in the relief chain, focusing on uncertainty with respect to candidate suppliers. Ke (2022) designed a robust two-stage model for reliable emergency logistics for hazardous materials. Sun, Wang, and Xue (2021) proposed a robust bi-objective integrated model for facility location, resource allocation, and casualty transportation planning in order to organize a coordinated treatment of injured people. Aliakbari, Komijan, Tavakkoli-Moghaddam, and Najafi (2022) used a scenario-based approach to model relief logistics planning under uncertainty of demand and travel times. Li and Liu (2023) designed an emergency relief network considering international cooperation, formulated as a two-stage distributionally robust optimization model and solved with a decomposition-based algorithm. Wang, Yang, Yang, and Dong (2023) considered two-stage distributionally robust optimization for disaster relief logistics under option contract and demand ambiguity. Avishan, Elyasi, Yanıkoğlu, Ekici, and Özener (2023) provided an ARO model for humanitarian relief distribution taking into account uncertain travel times to optimize routes and service times for visited locations.

Nevertheless, the use of ARO in emergency response faces several challenges, one of which is computational tractability. While the ARO approach usually leads to significant improvements in the quality of the solutions, the sheer number of possible courses of action leads to tractability issues which often render its employment inoperable in practice. Caunhye, Zhang, Li, and Nie (2016), for instance, proposed a two-stage capacitated location-routing model for relief distribution with transshipment considerations, converting it to a single-stage counterpart that can be solved with off-the-shelf solvers. However, the



(a) Static (b) 2-Adaptable (c) Completely Adaptable

Fig. 1. Uncertainty set enlargement effect for the uncertainty set $\mathcal{U}$ in example (2). The solution of the static problem is robust towards the smallest box set containing $\mathcal{U}$. Increasing the adaptability of the solution mitigates this effect.

problem is NP-hard and remains so even in the equivalent single-stage formulation. The corresponding experimental studies are therefore limited to a small illustrative example, which demonstrates the benefits of planning routing in coordination with other operations.

Thus, ARO does not share one of the most appealing features of its static counterpart, namely the ability to preserve the computational complexity of its deterministic counterpart. As a result, the related literature has focused on resolving this tractability issue. One popular remedy to the tractability issue consists of limiting the second-stage variables, which effectively are functions of the uncertainty, to the space of affine functions. This is referred to as affinely adjustable robust optimization (AARO) (Ben-Tal et al., 2004) and has been shown to work well within certain classes of problems (Yanıkoğlu, Gorissen, & den Hertog, 2019). Ben-Tal, Chung, Mandala, and Yao (2011) make use of affine decision rules for outbound logistics from the disaster site, solving a dynamic multi-period emergency response and evacuation traffic assignment problem with uncertain demand at the source nodes.

However, there are several issues which prevent AARO from serving as a suitable framework for the preparation of an emergency response. Firstly, AARO does not accommodate discrete recourse decisions. Since heavy machinery and the supplies considered for pre-positioning and transport are usually only available in predefined units, their representation as integer numbers appears logical. More significantly, however, an affine decision policy does not allow control over the adjustability of the solution. It always provides a different second-stage decision for every single possible realization of the uncertainty, which is naturally misaligned with human decision-making practice. Preparing the execution of such a large number of response plans is simply not viable, especially when we know that no more than one of the plans will be indeed put into motion.

2.2. K-Adaptability

A worthwhile response must make efficient use of the available resources, namely, it must not require a disproportionate amount of supplies, time, or organizational effort. This can only be achieved if the number of prepared response plans is limited, resulting in a finitely adaptable, or K-adaptable, robust model (Bertsimas & Caramanis, 2010).

From a mathematical standpoint, solving a K-adaptable robust optimization problem corresponds to finding an optimal piecewise constant recourse function. The function maps elements from the uncertainty set to the K available plans, creating one subset within the uncertainty set $\mathcal{U}$ for each of the plans. It hereby defines a partition of $\mathcal{U}$ into multiple cells, such that all scenarios in one cell share the same second-stage decision. Now, instead of considering one static problem with the whole uncertainty set as in Fig. 1(a), the problem represents a collection of K static subproblems, thus mitigating conservativeness Fig. 1(b). By increasing K, we are able to approximate the completely adaptable problem, where every scenario is contained in its own singleton subset with its own second-stage decision Fig. 1(c), and where the uncertainty enlargement effect does not hold. As a result, the K-adaptable model is able to mitigate the conservativeness of the static robust model while at the same time offering a more straightforward response plan than the completely adaptable model.

From the completely adaptable formulation (3) we derive the K-adaptable formulation by first replacing the recourse function $y : \mathcal{U} \to \mathbb{R}$ with $y : \mathcal{U} \to \{y_1, \dots, y_K\}$. From this, we obtain the partition of $\mathcal{U}$ via the reverse images of the plans y_k, i.e.,

$$\mathcal{U} = \mathcal{U}_1 \cup \dots \cup \mathcal{U}_K, \text{ where } y(\mathcal{U}_k) = y_k.$$

Thus, we obtain the K-adaptable formulation with variables $\mathcal{U}_1, \dots, \mathcal{U}_K$, as well as $x, y_1, \dots, y_K, z$:

$$\min_{x,y,z,\mathcal{U}_1,\dots,\mathcal{U}_K} \quad z \tag{4a}$$

$$\text{s.t. } c^T x + d^T y_k \le z \qquad k = 1, \dots, K \tag{4b}$$

$$Ax + By_k \ge \xi_k \qquad k = 1, \dots, K,\ \forall \xi_k \in \mathcal{U}_k \tag{4c}$$

$$\mathcal{U} = \mathcal{U}_1 \cup \dots \cup \mathcal{U}_K. \tag{4d}$$

In terms of bounds, the static (one-stage) problem (1) corresponds to the K-adaptable problem (4) with the additional constraint $y_1 = \dots = y_K$. For the corresponding optimal objective values z_{Static} and $z_{K\text{-adapt}}$, we therefore have $z_{\text{Static}} \ge z_{K\text{-adapt}}$. In the computational study in Section 4, when estimating the quality of a K-adaptable solution, we will accordingly use the static solution as a reference point. Similarly, we may consider (4) to be a special case of complete adaptability (3), thus obtaining $z_{K\text{-adapt}} \le z_{\text{CompAdapt}}$.

3. Methodology

3.1. The adjustable pre-allocation model

The setting considered here is as follows. A given amount of supplies is to be stored at a given set of locations such that in the event of a disaster, they can be transported quickly and efficiently to the affected regions, whose demands are not known during the preparation stage. We model the supply network via a complete bipartite graph with service nodes $i = 1, \dots, n$ and demand nodes $j = 1, \dots, m$. The demand ξ_j arises at demand node j and ought to be satisfied. Unsatisfied demand is penalized in the objective. In addition, the objective function measures the cost c_{ij} of allocating a supply kit from service node i to demand node j. This cost can represent various matters, such as the estimated transportation cost or response time from i to j, depending on the decision maker's priorities. All in all, we have the following given data.

- $i = 1, \dots, n$ service nodes

- $j = 1, \dots, m$ demand nodes
- c_{ij} - cost of allocating one supply unit from i to j
- p - penalty per unit of unsatisfied demand
- r - total available units of supply
- $\mathcal{U}$ - uncertainty set containing different demand scenarios ξ

The decision variables concern the amount of supplies $x \in \mathbb{N}^n$ stored at each location and the allocation of those supplies to the demand points, denoted by the variable $y \in \mathbb{N}^{n\times m}$. The formulation aims to minimize $z = \max z_k$, where z_k is the worst-case objective for the kth cell in the partition. In addition to x and y, the formulation includes slack variables s_j^k to account for unsatisfied demand, and, most importantly, the cells of the partition $\mathcal{U}_1, \dots, \mathcal{U}_K \subset \mathbb{R}$, which are variables as well. The model is as follows.

$$\min_{\mathcal{U}_1,\dots,\mathcal{U}_K} \min_{x,y,s,z} z \tag{5a}$$

$$\text{s.t. } z_k \le z \quad k = 1, \dots, K \tag{5b}$$

$$p\sum_{j=1}^{m} s_j^k + \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} y_{ij}^k \le z_k \quad k = 1, \dots, K \tag{5c}$$

$$\sum_{j=1}^{m} y_{ij}^k \le x_i \quad i = 1, \dots, n,\ k = 1, \dots, K \tag{5d}$$

$$\sum_{i=1}^{n} x_i \le r \tag{5e}$$

$$\sum_{i=1}^{n} y_{ij}^k + s_j^k \ge \xi_j^k \quad \forall \xi_j^k \in \mathcal{U}_k; j = 1, \dots, m,\ k = 1, \dots, K \tag{5f}$$

$$\mathcal{U} = \mathcal{U}_1 \cup \dots \cup \mathcal{U}_K \tag{5g}$$

$$x \in \mathbb{N}^n,\ y^k \in \mathbb{N}^{n\times m},\ z \in \mathbb{R};\ s^k \in \mathbb{N}^m,\ z_k \in \mathbb{R}\ \forall k = 1, \dots, K. \tag{5h}$$

Constraint (5c) 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 (5d) ensures that a supply point only provides as many supplies as are stored there according to the first-stage decision x. Constraint (5e) bounds the aggregated amount of supplies, and (5f) ensures the satisfaction of ξ_j^k, the worst-case demand at demand point j in cell k. Analogously to Han et al. (2022), we considered adding a constraint $z_1 \le z_2 \le \dots \le z_K$ to eliminate the symmetry arising from the permutation independence of the subsets, but preliminary experiments did not conclusively show a computational benefit in our experimental setting.

For the uncertainty set, we assume for each demand node $j = 1, \dots, m$ location coordinates $l_j \in \mathbb{R}^2$ as well as a node-specific demand bound $D_j \in \mathbb{N}$, which represents the quantity at risk at the corresponding demand point j, such as the maximum number of people susceptible to the emergency in the location represented by j. Additionally, $b \in \mathbb{N}$ represents a bound for the aggregated demand over all nodes. Thus, for some $\alpha \in \mathbb{R}$ the definition is as follows.

$$\mathcal{U} = \{\xi \in \mathbb{R}^m :\ |\xi_j - \xi_{j'}| \le \alpha\, ||l_j - l_{j'}||_\infty, \quad 1 \le j, j' \le m, \tag{6a}$$

$$0 \le \xi_j \le D_j \tag{6b}$$

$$\sum_{j=1}^{m} \xi_j \le b\}. \tag{6c}$$

Constraint (6a) ensures that closely located demand points share similar demands, which introduces the notion of locality to the emergency. The parameter α hereby scales location distance to demand distance. Constraint (6b) ensures that demands are nonnegative and bounded above by the quantity at risk $D_j, j = 1, \dots, m$. Finally, constraint (6c) bounds the aggregated demand to the given value b. This excludes scenarios wherein all demand points are maximally affected. The complexity of problem (5) lies within constraint (5g), which ensures that the cells $\mathcal{U}_1, \dots, \mathcal{U}_K$ form a cover of the uncertainty set $\mathcal{U}$. This constraint prevents the direct use of an off-the-shelf optimization solver, motivating the development of tailored solution methods.

3.2. Branch-and-bound method

The algorithm was originally proposed by Subramanyam et al. (2020). The main idea is to circumvent the complexity stemming from constraint (5g) by solving the model for a predefined partition τ of the uncertainty set $\mathcal{U}$. This partition is iteratively constructed, starting with K empty sets, which are to form the partition. After solving the K-adaptable problem (5a)–(5f),(5h) for this empty partition, that is, without uncertain constraints (5f), the algorithm obtains the corresponding solution $(\overline{z}, \overline{x}, \overline{y}, \overline{s})$. With this solution, the separation problem (7) calculates the maximal violation in the set of uncertain constraints. To this end, the separation problem contains a binary variable $v_j^k \in \{0, 1\}$ indicating for each plan $k = 1, \dots, K$ in which constraint $j \in 1, \dots, m$ the maximum violation occurs. The continuous variable $\zeta \in \mathbb{R}$ then assumes the smallest of these K violations, because only one of the K plans must be feasible. For $M \in \mathbb{R}$ sufficiently large, the separation problem is

$$\max_{\zeta,\xi,v} \zeta \tag{7a}$$

$$\text{s.t. } \xi \in \mathcal{U} \tag{7b}$$

$$\sum_{j=1}^{m} v_j^k = 1 \quad \forall k = 1, \dots, K \tag{7c}$$

$$\zeta + \sum_{i=1}^{n} \overline{y}_{ij}^k + \overline{s}_j^k + M v_j^k \le M + \xi_j \quad \forall j = 1, \dots, m,\ \forall k = 1, \dots, K \tag{7d}$$

$$v_j^k \in \{0, 1\} \quad \forall j = 1, \dots, m,\ \forall k = 1, \dots, K. \tag{7e}$$

If $\zeta > 0$, a violation has been identified, and the algorithm proceeds to branch. The scenario ξ leading to the violation, obtained from the separation problem (7), must be added to one of the sets in τ, giving rise to K branches (one for each subset in the partition).

Algorithm 1 Branch-and-Bound Method (Subramanyam et al., 2020).

1. Initialize node set $\mathcal{N} = \{\tau^0\}$, where $\tau^0 = (\mathcal{U}_1^0, \dots, \mathcal{U}_K^0)$ with $\mathcal{U}_k^0 = \emptyset,\ \forall k = 1, \dots, K$. Set as incumbent solution $(\overline{\theta}, \overline{x}, \overline{y}) = (\infty, \emptyset, \emptyset)$.
2. Check convergence: If $\mathcal{N} \neq \emptyset$, continue to step 3. Otherwise, if $\overline{\theta} = \infty$, declare infeasibility, and if $\overline{\theta} < \infty$, then $(\overline{x}, \overline{y})$ is an optimal solution.
3. Select a node $\tau = (\mathcal{U}_1, \dots, \mathcal{U}_K) \in \mathcal{N}$ and remove it from $\mathcal{N}$.
4. For partition τ, solve the K-adaptable problem (5a)–(5f), (5h) and let (z, x, y) be an optimal solution. If $z > \overline{\theta}$, go to step 2.
5. Solve the separation problem (7), obtaining (ζ, ξ). If $\zeta \le 0$, set $(\overline{\theta}, \overline{x}, \overline{y}) = (z, x, y)$ and go to step 2, otherwise go to step 6.
6. Add the following K nodes to $\mathcal{N}$: $\tau_k = \{\mathcal{U}_1, \dots, \mathcal{U}_k \cup \{\xi\}, \dots, \mathcal{U}_K\}$ for each $k = 1, \dots, K$. Go to step 3.

Subramanyam et al. (2020) provide proof that the algorithm, in general, converges asymptotically. Furthermore, since the number of subsets K is decided beforehand and maintained throughout the algorithm, it is entirely in the hands of the decision-maker. However, because K also determines the number of branches, the tree grows very quickly for large K, which may cause computational problems in practice. This problem is mitigated by the fact that in the context we consider here, small K values lead to simpler responses and are therefore preferable.

3.3. Box CCG algorithm

Most existing solution algorithms, such as Bertsimas and Dunning (2016), Postek and Hertog (2016) and Subramanyam et al. (2020) rely

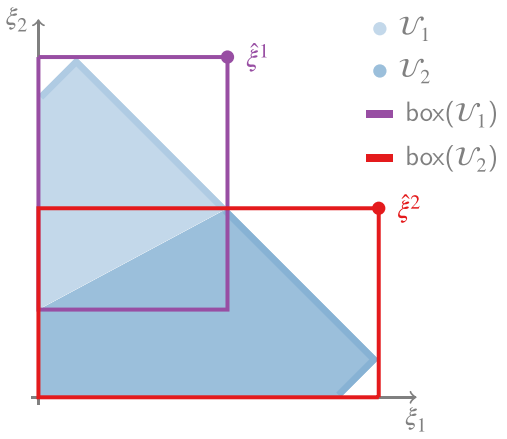


Fig. 2. Partitioning and parameterization of box sets. For every subset $\mathcal{U}_k, k = 1, \dots, K$ of the partition, box($\mathcal{U}_k$) denotes the smallest box set containing $\mathcal{U}_k$. The subset $\mathcal{U}_k$ is then parameterized by the worst-case scenario $\hat{\xi}^k$ found in box($\mathcal{U}_k$).

on pre-fixed partitions, which are iteratively improved or adapted in order to approximate the optimal partition. Consequently, the model is unable to shape the partition to accommodate better solutions. This motivates us to explore an approach which does not rely on pre-fixed partitions but instead integrates the partition parameters as problem variables to be optimized. To this end, we must find a class of partitions which is parameterizable and able to represent any optimal partitioning.

We depart from the results in Bertsimas and Caramanis (2010), which show that solving a static robust problem with uncertainty set $\mathcal{U}$ is equivalent to solving the problem with the smallest box set containing $\mathcal{U}$ as uncertainty set. This extends to the K-adaptable case: Every second-stage decision is robust with respect to the smallest box set containing the corresponding subset in the partition (see Fig. 2). Consequently, every solution to the problem has a corresponding box-partitioning. We may thus partition, without loss of generality, directly using box-shaped subsets.

The parameterization of the box partitioning retains only the information relevant to the solution of the problem: Its worst-case scenarios $\hat{\xi}^k$. If these scenarios satisfy demand constraints, then all other scenarios in the corresponding box (which are dominated by the worst case) are also satisfied, as illustrated in Fig. 2. Note that the partitions derived in this way are not true partitions, as they may overlap. This, however, does not pose a problem, as having several feasible policies instead of one for every uncertain scenario does not violate feasibility. Contrarily, the difficulty in this approach is to make sure that the whole uncertainty set is covered by the ensuing boxes.

To obtain a formulation, we replace the partition variables $\mathcal{U}_k$ in problem (5) by their worst-case scenario as follows. We introduce for every $k = 1, \dots, K$, a variable $\hat{\xi}^k \in \mathbb{R}^m_+$ representing the worst-case scenario associated with $\mathcal{U}_k$ (note that previously, ξ was merely a parameter). To make sure that the implicit boxes ensuing from these scenarios cover the whole uncertainty set, we add binary variables $v^k_u \in \{0,1\}$ for all $u \in \mathcal{U}, k = 1, \dots, K$, indicating whether scenario u is covered by the box of $\hat{\xi}^k$. Thus, the *box reformulation* is

$$\min_{\hat{\xi}^k, k=1,\dots,K} \ \min_{x,y,s,z} \ z \tag{8a}$$

$$\text{s.t. } z_k \le z \quad k = 1, \dots, K \tag{8b}$$

$$p \sum_{j=1}^{m} s^k_j + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} y^k_{ij} \le z_k \quad k = 1, \dots, K \tag{8c}$$

$$\sum_{j=1}^{m} y^k_{ij} \le x_i \quad i = 1, \dots, n, \ k = 1, \dots, K \tag{8d}$$

$$\sum_{i=1}^{n} x_i \le r \tag{8e}$$

$$\sum_{i=1}^{n} y^k_{ij} + s^k_j \ge \hat{\xi}^k_j \quad \forall j = 1, \dots, m, \ k = 1, \dots, K \tag{8f}$$

$$u_j v^k_u \le \hat{\xi}^k_j \quad \forall j = 1, \dots, m, \ k = 1, \dots, K, u \in \mathcal{U} \tag{8g}$$

$$\sum_{k=1}^{K} v^k_u \ge 1 \quad \forall u \in \mathcal{U} \tag{8h}$$

$$x \in \mathbb{N}^n, \ y^k \in \mathbb{N}^{n \times m}, \ s^k \in \mathbb{N}^m, \ \hat{\xi}^k_j \in \mathbb{R}_+,$$
$$v^k_u \in \{0,1\}, \ z, z_k \in \mathbb{R} \ \forall k = 1, \dots, K. \tag{8i}$$

The feasibility of (8) requires that every u must be covered by at least one scenario, which is ensured by constraints (8g)–(8h). Consequently, we obtain an equivalent reformulation of the K-adaptable pre-allocation problem.

Theorem 1. *The K-adaptable problem (5) and the box reformulation (8) are equivalent.*

Proof. Because constraints (5b)–(5e) and (8b)–(8e) as well as the objective functions are the same, we only need to show that (5f)–(5g), in the following referred to as (A), are equivalent to (8f)–(8h), referred to as (B). For this purpose, let $(x, y, s, \mathcal{U}_1, \dots, \mathcal{U}_K)$ be a feasible solution for (A). For all $k = 1, \dots, K$ and $u \in \mathcal{U}$, we set $v^k_u = 1$ if $u \in \mathcal{U}_k$ and $v^k_u = 0$, otherwise. Now (5f) implies $\sum_{i=1}^{n} y^k_{ij} + s^k_j \ge u_j v^k_u$ for all $j = 1, \dots, m$ and $k = 1, \dots, K$ and therefore (8f)–(8g) hold. Furthermore, (5g) implies (8h). Thus, (x, y, s) is feasible for (B).

Conversely, let (x, y, s, v) be feasible for (B). We define $\mathcal{U}_k = \{u \in \mathcal{U} \mid v^k_u = 1\}$ for all $k = 1, \dots, K$. Then (8f)–(8g) implies (5f), and (8h) implies (5g). Therefore, (x, y, s) is feasible for (A). □

The reformulation of the K-adaptable problem presented here is tailored to the pre-allocation problem with right-hand-side uncertainty presented in Section 3.1. Han et al. (2022) offer a set-based reformulation originating from the same principle but catered towards distributionally robust programs, referring to the box-shaped subsets as translated orthants. Demonstrating that their reformulation is hard to solve due to an exponential number of constraints needed to make sure the orthants cover the whole uncertainty set, they devise several approximative algorithms referred to as orthant-based policies.

Alternatively, to solve (8), we devise an exact approach using column-and-constraint generation (CCG). To handle the semi-infinite constraint (8g), we define as main problem the following relaxation of (8), where $\mathcal{U}^t$ is a finite subset of $\mathcal{U}$ with $|\mathcal{U}^t| \le t \in \mathbb{N}$:

$$\min_{\hat{\xi}^k, k=1,\dots,K} \ \min_{x,y,s,z} \ z \tag{9a}$$

$$\text{s.t. } z_k \le z \quad k = 1, \dots, K \tag{9b}$$

$$p \sum_{j=1}^{m} s^k_j + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} y^k_{ij} \le z_k \quad k = 1, \dots, K \tag{9c}$$

$$\sum_{j=1}^{m} y^k_{ij} \le x_i \quad i = 1, \dots, n, \ k = 1, \dots, K \tag{9d}$$

$$\sum_{i=1}^{n} x_i \le r \tag{9e}$$

$$\sum_{i=1}^{n} y^k_{ij} + s^k_j \ge \hat{\xi}^k_j \quad \forall j = 1, \dots, m, \ k = 1, \dots, K \tag{9f}$$

$$u_j v^k_u \le \hat{\xi}^k_j \quad \forall j = 1, \dots, m,$$

$$k = 1, \dots, K, u \in \mathcal{U}^t \quad (9g)$$

$$\sum_{k=1}^{K} v_u^k \geq 1 \quad \forall u \in \mathcal{U}^t \quad (9h)$$

$$x \in \mathbb{N}^n,\ y^k \in \mathbb{N}^{n\times m},\ s^k \in \mathbb{N}^m,\ \hat{\xi}_j^k \in \mathbb{R}_+,$$

$$v_u^k \in \{0,1\},\ z, z_k \in \mathbb{R}\ \forall k = 1, \dots, K. \quad (9i)$$

Problem (9) is a mixed-integer linear program with a finite number of variables and constraints. Starting with $t = 0$ and $\mathcal{U}^0 = \emptyset$, we obtain an optimal solution $(x^*, y^*, s^*, \hat{\xi}^*)$ for the main problem. To evaluate the feasibility of this solution for the complete problem (8), we use the separation problem (7) from the branch-and-bound scheme in Section 3.2. Replacing $\sum_{i=1}^{n} \overline{y}_{ij}^k + \overline{s}_j^k$ by the worst-case scenarios $(\hat{\xi}^*)_j^k$ in (7), we obtain the following separation problem:

$$\max_{\zeta,u,w} \ \zeta \quad (10a)$$

$$\text{s.t. } u \in \mathcal{U} \quad (10b)$$

$$\sum_{j=1}^{m} w_j^k = 1 \quad \forall k = 1, \dots, K \quad (10c)$$

$$\zeta + M w_j^k \leq M + u_j - \hat{\xi}_j^k \quad \forall j = 1, \dots, m, \forall k = 1, \dots, K \quad (10d)$$

$$w_j^k \in \{0,1\} \quad \forall j = 1, \dots, m, \forall k = 1, \dots, K. \quad (10e)$$

If $\zeta \leq 0$, then no constraint violation is occurring. Otherwise, we obtain a scenario $u^t \in \mathcal{U}$ which is not in any box set of the partition represented by $(\hat{\xi}^*)^k, k = 1, \dots, K$. Setting $\mathcal{U}^{t+1} = \mathcal{U}^t \cup \{u^t\}$, we resolve the main problem and repeat the procedure until a stopping criterion has been reached, giving rise to the Box CCG algorithm (see algorithm 2).

Algorithm 2 Box CCG Algorithm.

1. Initialize $t = 0$ and $\mathcal{U}^0 = \emptyset$.
2. Solve the main problem (9) for $\mathcal{U}^t$, obtaining a collection of worst-case scenarios $\hat{\xi}^1, \dots, \hat{\xi}^k$.
3. Solve the separation problem (10) for the obtained worst-case scenarios, obtaining the maximum violation ζ and corresponding scenario $u^t \in \mathcal{U}$.
4. If a violation occurred, i.e., $\zeta > 0$, set $t = t + 1$ as well as $\mathcal{U}^{t+1} = \mathcal{U}^t \cup \{u^t\}$ and go to step 2. Otherwise, return the current solution.

This formulation accommodates the right-hand-side uncertainty of the supply pre-allocation problem, however, the extension to the general case is straightforward. The Box CCG algorithm is comparable to the branch-and-bound method (algorithm 1) by relating the binary variables v_u^k to the branching decisions: If $v_u^k = 1$, then scenario u is added to subset $\mathcal{U}_k$. Instead of fixing the v_u^k to different combinations and solving the problem for each combination as the branch-and-bound method does, the Box CCG algorithm integrates the combination decision, that is, the decision which scenarios belong to which subset of the partition, into the MIP formulation. It thereby allows for an implicit representation of all possible combinations of v_u^k within the MIP formulation to avoid explicitly and exhaustively exploring them all. Theorem 2 states that, similarly to the branch-and-bound method (Subramanyam et al., 2020, Theorem 2), algorithm 2 enjoys asymptotic convergence in general.

Theorem 2. *The Box CCG algorithm (algorithm 2) converges asymptotically towards a feasible, optimal solution to the K-adaptable supply pre-allocation problem.*

Proof. Let $\mathcal{U}^\ell, \ell = 0, 1, \dots$, be the sequence of subsets of $\mathcal{U}$ generated by the algorithm. We denote by ξ^ℓ the sequence of corresponding optimal box-corners from the scenario-based K-adaptability problem in step 2, and by u^ℓ the sequence of optimal uncovered scenarios from the separation problem in step 3, respectively. Since the feasible sets for ξ and u are compact (bounded below by 0, and due to the minimization, a sufficiently large upper bound can be introduced without loss of generality), the Bolzano–Weierstrass theorem (Bartle & Sherbert, 1994) implies that ξ^ℓ and u^ℓ each have at least one accumulation point $\hat{\xi}$ and $\hat{u}$, respectively.

We first show that the solution $(\hat{x}, \hat{y}, \hat{s})$ to the K-adaptable problem obtained together with $\hat{\xi}$ is optimal. As $\hat{\mathcal{U}}$, the subset corresponding to this solution, satisfies $\hat{\mathcal{U}} \subseteq \mathcal{U}$, the main problem is a relaxation of the complete problem (8) and any solution optimal for the relaxation is therefore by definition optimal to the complete problem.

Now, it remains to be shown that every accumulation point $\hat{\xi}$ of the sequence ξ^ℓ corresponds to a feasible partition of the uncertainty set, that is, for all $\xi \in \mathcal{U}$ exists $k \in \{1, \dots, K\}$ such that $\xi \leq \hat{\xi}^k$. By possibly going over to subsequences, we can without loss of generality assume that the two sequences ξ^ℓ and u^ℓ converge themselves to $\hat{\xi}$ and $\hat{u}$, respectively. By continuity of the separation problem (10), and denoting with $S(\bar{\xi}, \bar{u})$ the value of the separation problem for $\xi = \bar{\xi}$ and $u = \bar{u}$, we have

$$S(\hat{\xi}, \hat{u}) = S(\lim_{\ell\to\infty} \xi^{\ell+1}, \lim_{\ell\to\infty} u^\ell) = \lim_{\ell\to\infty} S(\xi^{\ell+1}, u^\ell). \quad (11)$$

Now, because $u^\ell \in \mathcal{U}^{\ell+1}$ according to step 4, we have $S(\xi^{\ell+1}, u^\ell) < 0$, and therefore

$$S(\hat{\xi}, \hat{u}) = \lim_{\ell\to\infty} S(\xi^{\ell+1}, u^\ell) < 0. \quad (12)$$

Consequentially, $\hat{\xi}$ corresponds to a feasible partition of $\mathcal{U}$, which concludes the proof. □

3.4. Observable objectives

Being robust optimization methods, all of the previously discussed modeling and solution approaches return as the objective value that of the scenario $\hat{\xi}^k$, which dominates all worst-case scenarios together as the corner of the smallest box set within which $\mathcal{U}$ is contained. This scenario is, as illustrated in example (2), not necessarily within the uncertainty set and therefore does not reflect the performance of the solution in reality. Given second-stage decisions $\overline{y}^k \in \mathbb{N}^{n\times m}, k = 1, \dots, K$, we can instead calculate the true worst-stage cost, that is, the upper bound for the actually realized cost, by solving the robust bilevel programming problem (13) (for more information about bilevel optimization under uncertainty, see e.g., Beck, Ljubić, & Schmidt, 2023). Within the problem, the upper level maximizes over all possible demand scenarios $\xi \in \mathcal{U}$ the true response cost $\theta \in \mathbb{R}$, which is determined by the lower level. For each plan $k = 1, \dots, K$ and the scenario ξ provided by the upper level, the lower level calculates the (nonnegative) unsatisfied demand $s^k \in \mathbb{R}_+^m$ as well as the cost $z^k \in \mathbb{R}$ of plan k for this scenario, which is to be minimized. The lower-level variable θ is then maximized to assume the smallest of the z^k, as in reality, the most advantageous plan will be implemented, thus determining the truly observed cost in scenario ξ. The small weight $\varepsilon > 0$ in the lower-level objective ensures that the model prioritizes the minimization of z^k over the maximization of θ. The problem is posed as follows.

$$\max_{\xi} \ \theta \quad (13a)$$

$$\text{s.t. } \xi \in \mathcal{U} \quad (13b)$$

$$(\theta, z, s) \in Z(\xi), \quad (13c)$$

where

$$Z(\xi) = \operatorname*{argmin}_{\theta,z,s} \sum_{k=1}^{K} z^k - \varepsilon\theta \quad (13d)$$

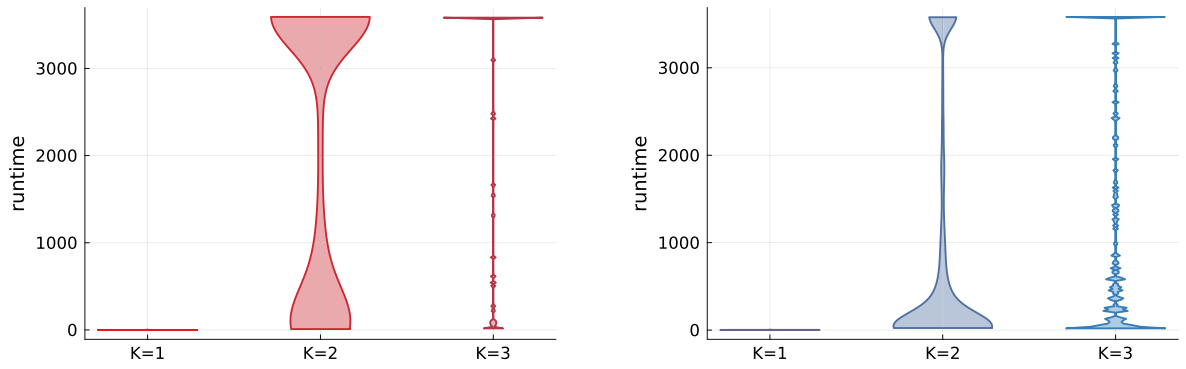


Fig. 3. Distribution of runtimes in seconds over all instances for $K = 1, 2, 3$.

$$\text{s.t. } z^k = p\sum_{j=1}^{m} s_j^k + \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}\bar{y}_{ijk} \qquad \forall k = 1, \dots, K \tag{13e}$$

$$\theta \le z^k \qquad \forall k = 1, \dots, K \tag{13f}$$

$$s_j^k \ge \xi_j - \sum_{i=1}^{n} \bar{y}_{ij}^k \qquad \forall j = 1, \dots, m, k = 1, \dots, K \tag{13g}$$

$$s \in \mathbb{R}_+^{m\times K}, z \in \mathbb{R}^K, \theta \in \mathbb{R}. \tag{13h}$$

4. Computational study

In this section, we analyze the algorithms presented in Section 3 concerning computational performance and practical applicability. Initially, we implemented three algorithms for the solution process: The branch-and-bound algorithm (in the following referred to as BB) from Section 3.2, the Box CCG algorithm (hereafter referred to as Box) from 2, as well as the partition-and-bound-method from (Bertsimas & Dunning, 2016) (see Appendix A). However, the lack of control over K in this method prevented a meaningful comparison to the remaining two methods, leading us to omit the results for conciseness. We shall therefore in the following compare BB and Box, using the static robust solution as a benchmark. To this end, the algorithms are implemented in Julia (Bezanson, Edelman, Karpinski, & Shah, 2017, v1.9.3) using JuMP (Lubin, Dowson, Dias Garcia, Huchette, Legat, & Vielma, 2023, v1.16) in combination with Gurobi (Gurobi Optimization, LLC, 2023 v10.0.0), and observable objectives are calculated using BilevelJuMP (Dias Garcia, Bodin, & Street, 2022, v0.6.2). All instances were solved on the Triton computing cluster, which is maintained by the Aalto Science-IT project. The Julia code as well as the problem data are available at Weller (2024).

4.1. Synthetic dataset

In this experiment, the K-adaptable framework was applied to a collection of randomly generated instances of the pre-allocation problem. We generated 50 instances each for $n = 4, 6, 8, m = 10, 15, 20$ and $pc = 0.1, 0.3$ (resulting in $18 \times 50 = 900$ instances in total), where pc represents the percentage of affliction, which is used to calculate the aggregated supply bound $r = pc \cdot \sum_{j=1}^{m} D_j$ as well as the aggregated demand bound, set to the same value $b = pc \cdot \sum_{j=1}^{m} D_j$. This means we assume to have enough supplies to, in theory, cover all demand. The demand bounds $D_j, j = 1, \dots, m$ are randomly generated integers from the interval [1,100] and the transportation costs c correspond to the Euclidian distances between the randomly generated locations of the service and demand points. The penalty for unsatisfied demand was set to $p = 1000c_{\max}$, where $c_{\max} = max\{c_{ij} \mid i \in \{1, \dots, n\}, j \in \{1, \dots, m\}\}$. Furthermore, we have $\alpha = 1$. The 900 instances were solved for $K = 1, 2, 3, 4, 5$ each, which results in 4 500 input combinations overall.

Table 1
Number of instances solved to optimality for different methods and values of K.

method	$K = 1$	$K = 2$	$K = 3, 4, 5$
both	900	401	75
Box only	0	353	119
BB only	0	0	1
neither	0	146	2 505
	900	900	2 700

Computational runtime

A time limit of 3 600 s was set as a termination criterion for both BB and Box. Table 1 shows the number of instances that were solved to optimality by either both BB and Box, by only one of the methods, respectively, and by neither of them for different values of K. For all K, Box appears to be superior to BB in this aspect, as it can solve to optimality a significant amount of instances more than BB. However, for $K \ge 3$, the majority of instances remain unsolved by both algorithms.

Fig. 3 shows the distribution of computational runtime for different values of K, demonstrating the significant impact of K on the runtime. While for $K = 2$ at least half of instances are solved by both algorithms, this reduces to a very small percentage for $K = 3$. The significant runtime difference is rooted in the larger number of branching decisions for BB, and the increase of binary variables in Box. Interestingly, the majority of runtimes veer close to the extremes, being solved either very quickly or close to the cutoff time. This suggests that the runtime is highly instance-specific. Supporting this notion, the results also show that the instances with $pc = 0.1$ had significantly longer runtimes on average than those with $pc = 0.3$ (see appendix, Fig. 13).

As shown in Fig. 4, the effect of the instance size n, m on runtime is less dramatic than the effect of K but is noticeable nonetheless. While the influence of n on runtime is negligible, the number of demand points m seems to impact it more significantly. This can be explained by the number of second-stage decisions as well as the dimensions of the uncertainty set, which both depend on m but not on n.

In conclusion, the runtime is mostly affected in descending order by the number of response plans and the number of demand points. The scalability with regards to K may be considered of less importance as, in this setting, a small number of response plans is desirable. In addition, this result is in line with the literature, as the only state-of-the-art algorithms able to handle large K are partitioning schemes such as the partition-and-bound method of Bertsimas and Dunning

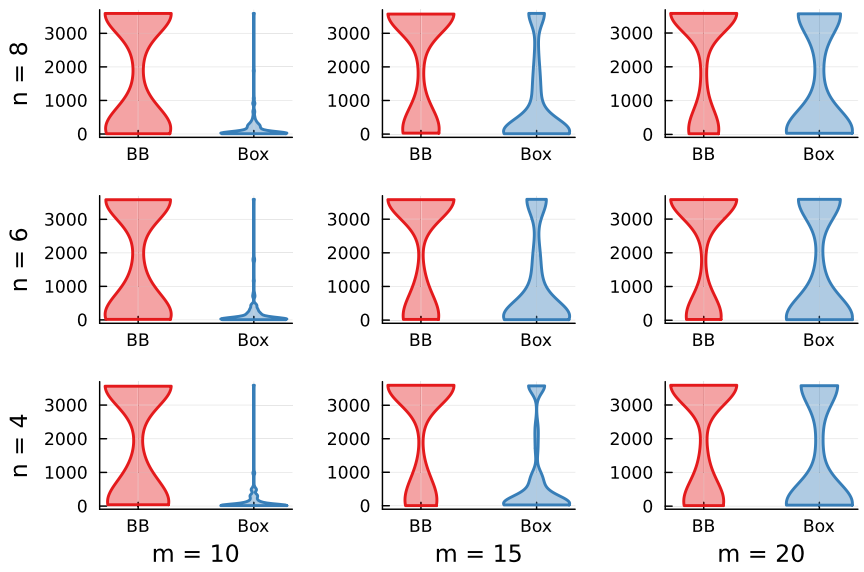
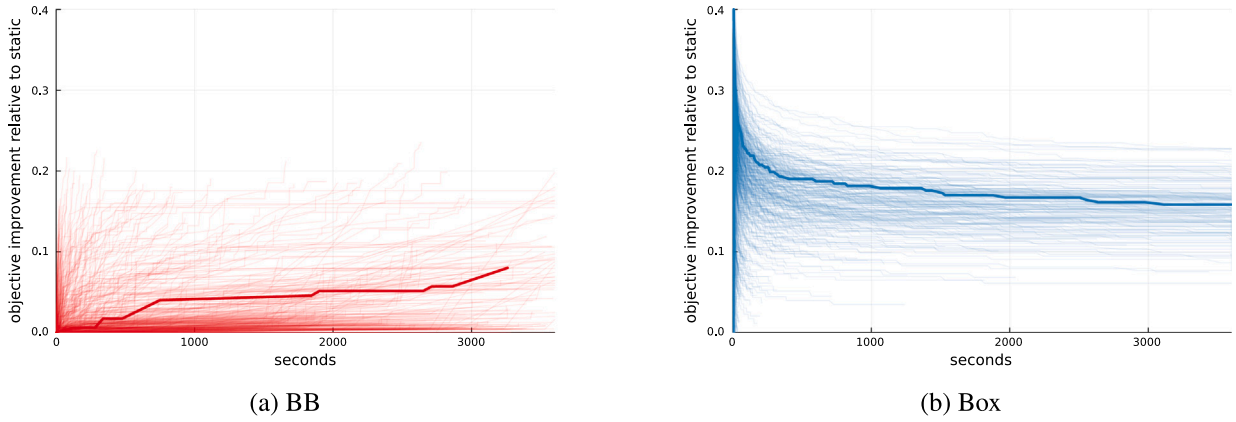


Fig. 4. Distribution of runtimes in seconds for $K = 2$ for different instance sizes.



(a) BB (b) Box

Fig. 5. Objective progression (relative to static solution) during algorithm execution for $K = 2$, one arbitrarily chosen representative curve is highlighted in each to showcase the bound value increase rate of BB versus the bound value decrease rate of Box.

(2016), wherein the partitions are fixed beforehand and iteratively improved, such that the decision-maker can neither decide the value for K nor obtain truly optimal partitions. With respect to instance size, its impact on runtime is more important, as in this experiment, the size is much smaller than expected for a real-world setting. This raises questions about the scalability of the approaches and necessitates further experimentation with larger, perhaps more realistic instances.

Convergence

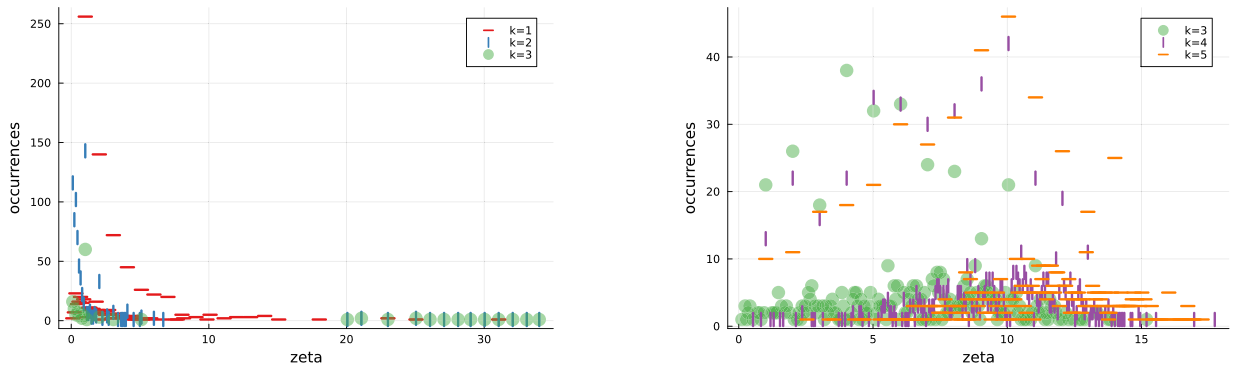
The majority of instances were only solved for $K \leq 2$, which inspires some investigation into the convergence behavior of BB and Box. For Box, the evolution appears to be exponentially decaying, meaning that it is very close to the final solution already in an early stage. When computation time is limited, this behavior is potentially valuable, provided that the objective plays an important role. If, however, feasibility is a major concern, the development of BB is more advantageous, as the incumbent solution is always feasible (see Fig. 5).

For Box, the feasibility is measured in terms of the subproblem objective ζ. Fig. 6 shows the occurrences of values for the last ζ before convergence (a) or termination, i.e., interruption due to the time limit (b). In case of convergence, unsurprisingly, most values are very close to zero. For the terminated runs, the bulk of ζ has not yet approached zero; however, the smaller K, the closer it appears to be to zero, indicating that at time $t = 3600$ the runs with smaller K had progressed further and implying a distinctive pattern of convergence even for the terminated instances. We conclude that in practice, with a limited timeframe, both algorithms are potentially useful depending on model characteristics and the decision-maker's priorities.

Solution quality

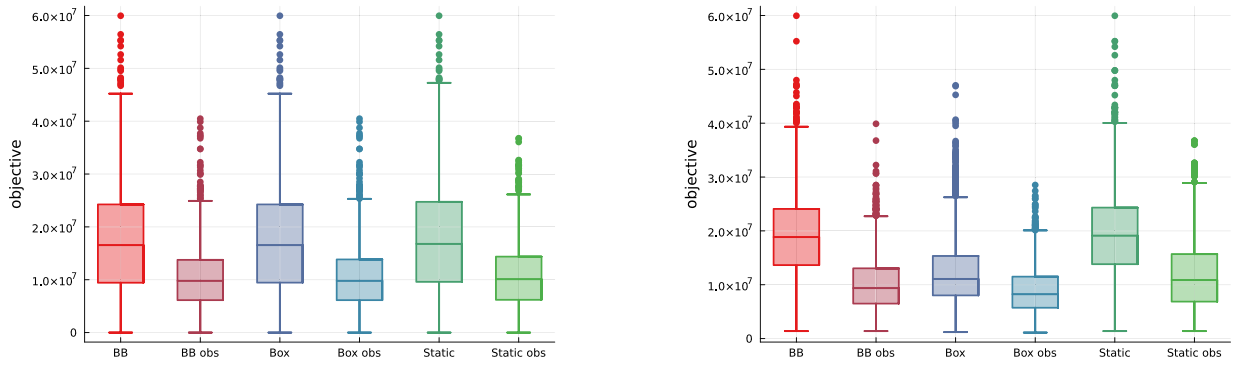
As mentioned in Section 3.4, the objective function value of robust models may not be a good indicator of observable performance. In addition, in our case, the model is designed in a way that feasibility is not a concern due to the slack variables measuring (and enabling) unsatisfied demand. Here, an infeasible solution simply means a solution in which not all unsatisfied demand was accounted for in the objective function. Making use of the observable objective, we can therefore analyze performance for suboptimal or infeasible solutions.

Fig. 7 shows the distribution of objectives across all 4500 input combinations for which both or neither algorithm converged, respectively (see Table 1 for exact numbers). It demonstrates that the model objective is not a good approximation for real-world performance if the objective is mainly centered around unsatisfied demand. The only case where the distribution of model objectives comes close to the actual performance is for the terminated cases of Box, because for those, the model objective is a lower bound due to solution infeasibility. In addition, a difference in performance between the two algorithms is visible mainly for the terminated cases, and thus for higher K.



(a) Converged runs for $K = 1, 2, 3$ (b) Terminated runs for $K = 3, 4, 5$

Fig. 6. Distribution of the last values of the infeasibility indicator ζ (zeta).



(a) Both converge (b) Neither converges

Fig. 7. Optimal objective values and observable objective values (obs) of the different algorithms.

Fig. 8 shows the observable performance of BB and Box relative to the static robust solution. Unsurprisingly, for $K = 1$, there is no improvement at all, as all three return a static solution. Nevertheless, it is noteworthy that these static solutions do not only share the model objective but also the observable objective, indicating that the optimal solution is either unique or that the three models are tantamount to this case. For $K = 2$, BB returns the best solutions on average. For $K = 3$, the average is similar, with a slightly higher variance for BB. For $K > 3$, the solution quality of BB stagnates while Box solutions improve further. An explanation for this is the higher variety in explored solutions — BB only returns feasible incumbents, and must therefore explore a branch in full depth before a solution is found. Our implementation of BB searches the tree width-first, which makes it difficult for BB to discover a wide range of incumbents at the beginning of the algorithm. Box, on the other hand, returns all infeasible incumbents, therefore having access to a wider variety of incumbent solutions. The graphs also indicate that for some instances, improvement is negative, which means that the static solution performs better in the observable case. There are two possible explanations for this. Firstly, these cases only occur for $K \geq 2$, which means it is possible that the corresponding algorithm was not able to converge within the time limit, thus not guaranteeing an optimal solution. Secondly, the model optimizes the model objective and not the observable objective, so that it is theoretically possible for the model to prefer a solution which is observably worse. For more detailed results on observable performance, see Fig. 14 (Appendix).

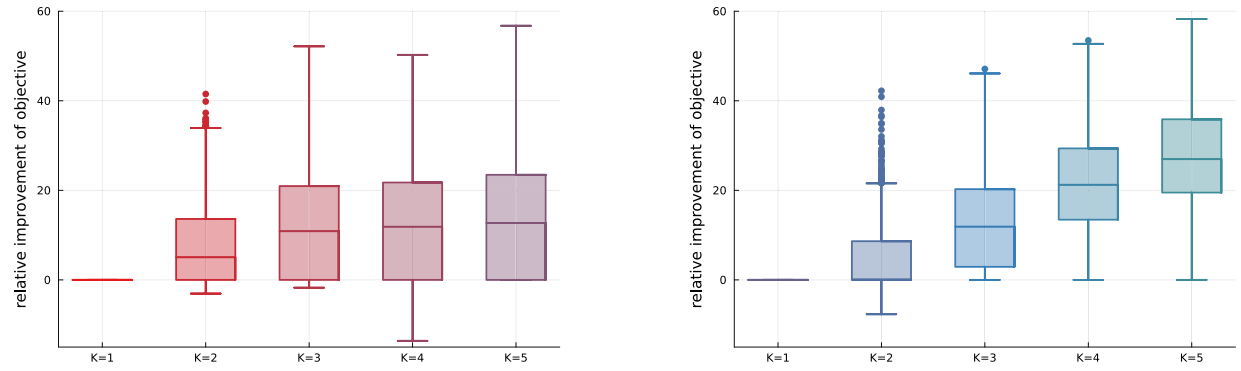
In conclusion, despite the poor runtime performance of BB and Box, both algorithms were able to significantly improve solution quality with respect to the static case. This is possible because of modeling decisions that accommodate the evaluation of theoretically infeasible solutions. As, in emergency response, the foremost concern is demand satisfaction, these modeling decisions are reasonable.

4.2. Case study

This case study is based on data from the flooding and landslide disaster in the Serrana region of Rio de Janeiro state, Brazil, in January 2011. Based on (Alem, Clark, & Moreno, 2016), four municipalities ($n = 4$) host depots for emergency supplies and there were nine affected regions ($m = 9$). The transportation costs c are set to the travel times between these nodes.

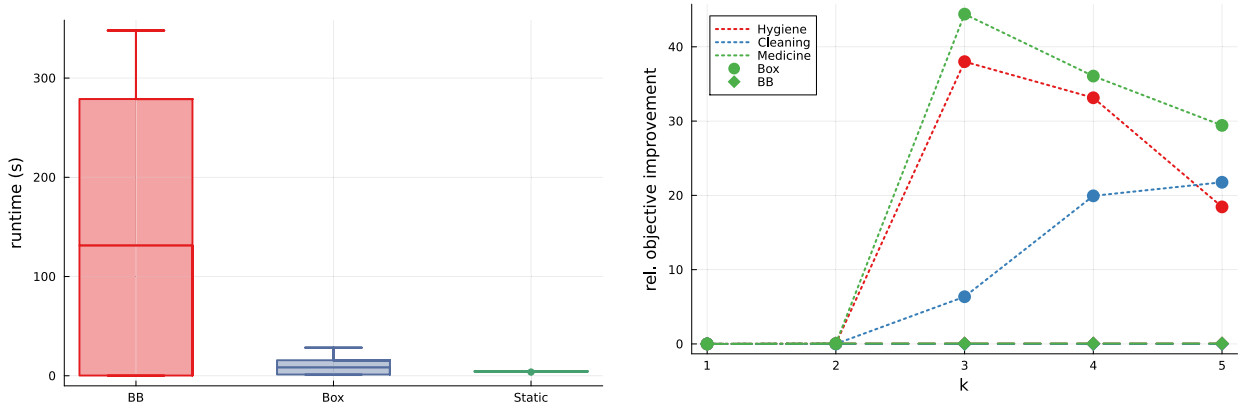
The uncertainty set is constructed considering the population of the respective areas as follows: The total population within the nine areas is 827 825, out of which a total of 304 562, or 36.79% was affected (that is, displaced, homeless, or otherwise directly or indirectly affected) by the disaster. We, therefore, chose this percentage of the corresponding node's population to define the node-specific demand bounds $D_j, j = 1, \ldots, m$. For instance, the region of Teresópolis with a population of 163 746 has a total demand bound $D_{TRS} = 163\,746 \cdot 0.3679$. However, not everyone who was affected will need supplies. Thus, for the aggregated demand bound b, we rely on the same estimate of the number of displaced, homeless and fatal victims as (Alem et al., 2016), which is 33 370 victims out of 304 562 total affected, or 10.96%. Therefore, $b = 33\,370$. The remaining constraint of the uncertainty set in Section 3.1, constraint (6a), is omitted for this case study since the historic demand values are not strictly concentric due to the mountainous geography of the region in combination with the nature of the disaster.

As emergency supply items, we considered hygiene kits, cleaning products, and medical products, each forming an individual instance. According to the available data, a kit of personal hygiene products serves one person, a unit of cleaning products serves a five-person family, and a kit of medical products serves 90 people. By multiplying the number of total available kits by the number of people the



(a) BB (b) Box

Fig. 8. Observable improvement of the objective value with respect to the static solution ($\frac{\theta_{static}-\theta}{\theta_{static}}$) for $K = 1, 2, 3, 4, 5$.



(a) Runtime in seconds for $K = 1,2$ (b) Objective improvement relative to the static robust solution

Fig. 9. Results from the case study. BB shows substantially larger computation times compared to Box, while not yielding any improvement for any K. Box yields no improvement for $K = 2$, but substantial improvement for higher K.

Table 2
Number of supplies r and observable objectives for the different supply items (instances). BB and Box objectives are for $K = 3$.

Supply	Quantity	Static	BB	Box
Hygiene	44 394	7.6802e9	7.6772e9	4.7615e9
Cleaning	40 000	7.6776e9	7.6772e9	7.1898e9
Medicine	64 260	7.6810e9	7.6772e9	4.2705e9

corresponding item serves, we obtain the supply bounds r as stated in Table 2.

For each of the three supply items, we executed the model for $K = 1, 2, 3, 4, 5$ with a time limit of 7 200 s, resulting in $3 \cdot 5 = 15$ instances in total. For $K = 1, 2$, all models were able to converge within the time limit, with the runtime statistics shown in Fig. 9(a). For $K \geq 3$ however, neither of the algorithms was able to converge for any of the instances. These results are consistent with the findings in Section 4.1, where the vast majority of instances did not converge within the time limit for $K \geq 3$.

For the converged instances $K = 1,2$, the observable objective value yields no improvement compared to the static model for either algorithm. The reason for this behavior lies in the number of available supplies: The presence of oversupply in this case study allows the model to partially compensate for a lack of flexibility in the case $K = 1$, forestalling any improvements for the case $K = 2$. For the non-converged instances, that is, $K \geq 3$, BB continues to show no improvements at all, while Box improvements range from 5 to 45% (see Fig. 9(b)). The improvement is not strictly increasing, with Medicine and Hygiene showing a decrease in improvement for $K = 4$ and $K = 5$, because the algorithm did not converge for these instances, therefore yielding suboptimal solutions.

As a measure of convergence, we again observe the subproblem objective value ζ, which measures infeasibility for the Box algorithm, as shown in Fig. 10. Once more, the behavior seems largely consistent with the observations from Section 4.1: The smaller K, the closer ζ is to zero, indicating that at time $t = 7\,200$ the runs with smaller K had progressed further, implying a distinctive pattern of convergence. In this case, the values of ζ are several orders of magnitude higher than the same values for the synthetic dataset, which can be attributed to the generally higher objectives, indicating a higher amount of unsatisfied demand in general.

Overall, we conclude that this case study seems more computationally challenging than the synthetic data, but nevertheless leads to significant improvements in the observable objective, and is aligned with most of the conclusions from Section 4.1.

5. Conclusion

This research aimed to shed light on the application of K-adaptability to the subject of emergency response. To this end, we developed a K-adaptable model for the preallocation of emergency supplies as well as a new reformulation of the same in terms of the worst-case scenarios. In a computational study, we assessed the potential of this approach by comparing two solution algorithms, one of them being our proposed

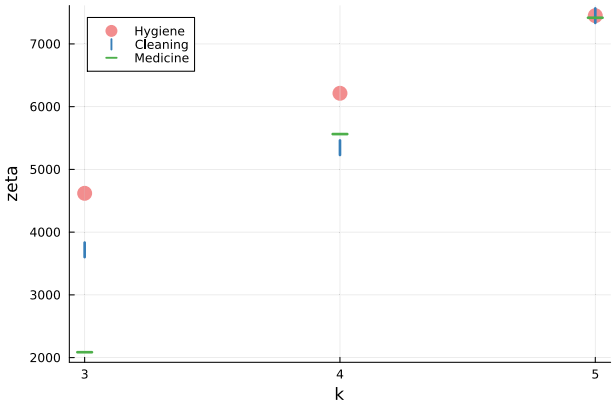


Fig. 10. Last zeta values of terminated runs.

novel Box CCG algorithm. From the numerical results, we conclude that K-adaptability possesses great potential with respect to this application. The experiments suggest that it is sufficient to consider very small values for K to obtain significant improvements in preparedness, a fitting result for the context of emergency response, where a small number of responses is desirable in many aspects, such as swiftness and cost efficiency.

Despite, or perhaps because of its potential, this avenue of research still faces many challenges, the overcoming of which shall be the subject of future work. Computation time could potentially be reduced for example, by introducing a symmetry-breaking constraint to all models, which in their current form are symmetric with respect to which subset is assigned to which $k \in \{1, \dots, K\}$. In addition, the branch-and-bound algorithm might benefit from a more careful node selection. The Box CCG algorithm could be improved by refining the interaction of the main and subproblems, for example by introducing regularization measures. The pre-allocation model could be enhanced by adding features such as the routing of transport or the unreliability of network components such as the used infrastructure. Furthermore, the simplified model adopted here exposes the relevant features for our developments but lacks the additional features relevant to specific cases, which will give further indication towards the applicability of K-adaptability. Likewise, it would be interesting to see how the model and algorithms perform in other real-world instances.

Based on these conclusions, this preliminary study successfully opens the door for further considerations of K-adaptability as a well-fitted framework for emergency response planning.

CRediT authorship contribution statement

Paula Weller: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Fabricio Oliveira:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Conceptualization.

Appendix A. Partition-and-bound method

Similar to the branch-and-bound method in Section 3.2, the method of Bertsimas and Dunning (2016) relies on pre-fixed partitions to solve the K-adaptable problem. However, instead of the partitioning being iteratively constructed, it is iteratively refined until a stopping criterion has been reached (see algorithm 3). To refine the partition, the number of subsets K is steadily increased by the algorithm, and can therefore not be chosen beforehand. The partitioning process is represented by a tree structure $\mathcal{T}$, wherein every node represents a subset of the uncertainty set. The tree $\mathcal{T}$ is initialized with a single node, which represents the whole uncertainty set $\mathcal{U}$. By solving the subproblems

$$Sub_j(\mathcal{U}) := \max_{\xi \in \mathcal{U}} \xi_j \qquad \text{for } j = 1, \dots, m,$$

we obtain, for each $j = 1, \dots, m$, an optimal $\overline{\xi}^j \in \mathcal{U}$, the so-called *active values*, which are the worst-case scenarios contained in $\mathcal{U}$. If constraint (5f) is fulfilled for the maximum component $\overline{\xi}^j_j$ for all $j = 1, \dots, m, k = 1, \dots, K$, then it must also hold for any other $\xi \in \mathcal{U}$. Thus, by solving the deterministic problem (5a)–(5e),(5h) with deterministic demand constraints

$$\sum_{i=1}^{n} y_{ij} + s_j \geq \overline{\xi}^j_j \quad \text{for all } j = 1, \dots, m \tag{15}$$

the algorithm obtains an optimal solution to the 1-adaptable pre-allocation model with the uncertainty set $\mathcal{U}$. Now, for each of the active values $\overline{\xi}^j, j = 1, \dots, m$, a child node is added to the tree $\mathcal{T}$. From now on, the following steps are repeated until the stopping criteria are met.

Algorithm 3 Partition-and-Bound Method (Bertsimas & Dunning, 2016).

1. Initialize: partitioning tree $\mathcal{T}$ with one node containing an arbitrary scenario $\xi \in \mathcal{U}$, and $K = 1$.
2. Compute Partition: Given the leaves of $\mathcal{T}$, compute a Voronoi diagram of $\mathcal{U}$ and partition accordingly into subsets $\overline{\mathcal{U}}_1, \dots, \overline{\mathcal{U}}_K$.
3. Solve K-adaptable problem (5a)–(5f), (5h) optimally for this partition, i.e., $\mathcal{U}_k = \overline{\mathcal{U}}_k$. Obtain active values $\overline{\xi}^{jk} = \operatorname{argmax}_{\xi \in \mathcal{U}_k} \xi_j$, $j = 1, \dots, m$, $k = 1, \dots, K$.
4. Grow Tree: For $k \in \{1, \dots, K\}$, add m children to node k in $\mathcal{T}$, one for each $\overline{\xi}^{jk}$. Update $K \leftarrow K \cdot m$.
5. If a stopping criterion is met, stop. Otherwise go to step 2.

Let K be the number of leaf nodes in $\mathcal{T}$ and $\xi^k \in \mathcal{U}, k = 1, \dots, K$ the scenarios contained within them. From these scenarios, the next partition is computed via a Voronoi diagram. Given the set of Voronoi points $\xi^1, \dots, \xi^K \in \mathcal{U}$, the Voronoi diagram separates the uncertainty set by Euclidian distance to these points: Each element in the uncertainty set is assigned to the point ξ^k it is closest to, thus defining K new subsets

$$\mathcal{U}_k = \{\xi \in \mathcal{U} \mid ||\xi^k - \xi|| \leq ||\xi^l - \xi|| \ \forall l = 1, \dots, K, l \neq k\} \tag{16a}$$

$$= \{\xi \in \mathcal{U} \mid (\xi^l - \xi^k) \cdot \xi \leq \frac{1}{2}(\xi^l - \xi^k) \cdot (\xi^l + \xi^k) \ \forall l = 1, \dots, K, l \neq k\} \tag{16b}$$

for $k = 1, \dots, K$ in accordance with (Bertsimas & Dunning, 2016). Fig. 11 illustrates this principle. Consequently, every leaf node of $\mathcal{T}$ represents one subset $\mathcal{U}_k \subseteq \mathcal{U}$, and together they form a partition of $\mathcal{U}$. Given the partition $\mathcal{U} = \mathcal{U}_1 \cup \dots \cup \mathcal{U}_K$, the algorithm solves the subproblems $Sub^k_j = Sub_j(\mathcal{U}_k)$ for all $j = 1, \dots, m$ and $k = 1, \dots, K$. The resulting active parameters $\overline{\xi}^{jk}$ are then used to construct a deterministic set of demand constraints

$$\sum_{i=1}^{n} y^k_{ij} + s^k_j \geq \overline{\xi}^{jk}_j \quad \text{for all } j = 1, \dots, m, \ k = 1, \dots, K, \tag{17}$$

which, together with (5a)–(5e),(5h) leads to an optimal solution to the K-adaptable pre-allocation model with the fixed partition $\mathcal{U} = \mathcal{U}_1, \dots, \mathcal{U}_K$. Subsequently, the partitioning tree $\mathcal{T}$ is grown by adding, for every $j = 1, \dots, m$, the child node $\overline{\xi}^{jk}$ to ξ^k. Thus, the number of leaf nodes K increases by m in every iteration. In the next partitioning step, the values are separated by the Voronoi diagram. This leads to an improvement of the current partition: by separating $\overline{\xi}^{jk}$, $j = 1, \dots, m$, via partitioning, the model is freed from the need to accommodate several "extreme" scenarios with the same plan y_k. By always generating a new subset for every active value each, which is to say m new subsets, we

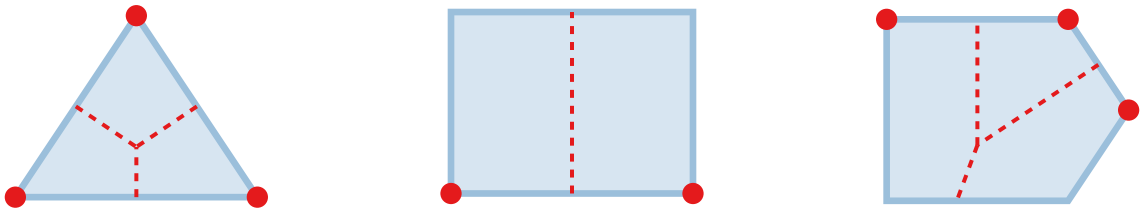


Fig. 11. Voronoi diagrams. The Voronoi diagram splits the set according to proximity to the Voronoi points.

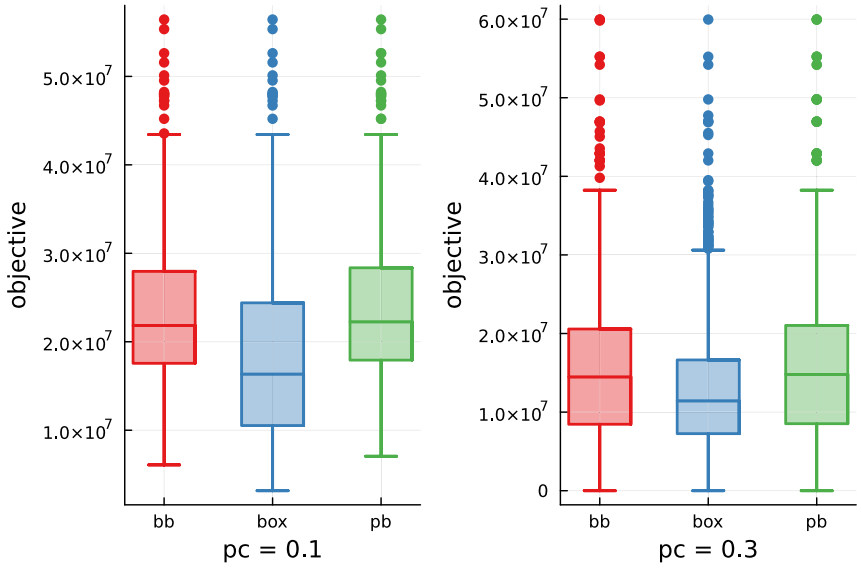


Fig. 12. The Effect of pc on the optimal objective value (pb corresponds to the static solution). For pc = 0.3, the optimal objective values are consistently smaller.

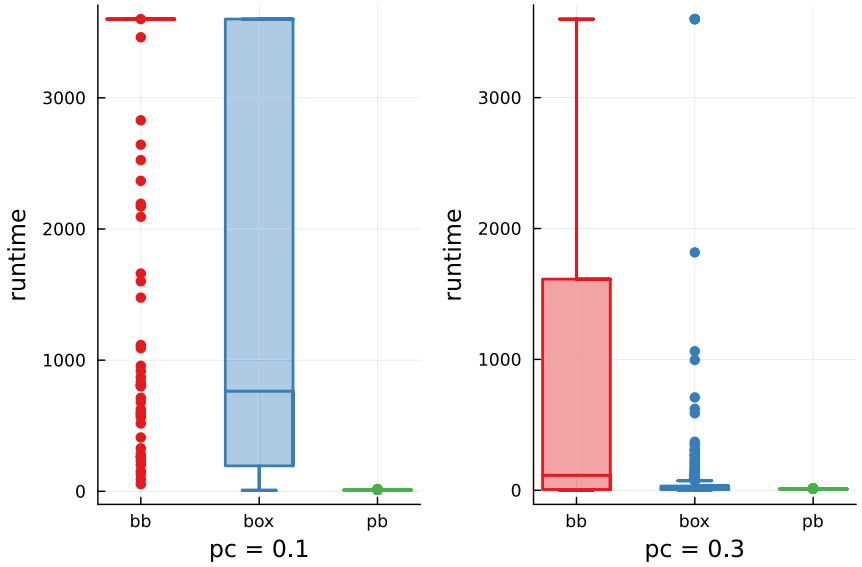


Fig. 13. The Effect of pc for $K = 2$ on the runtime (pb corresponds to the static solution). For pc = 0.3, the objective value is drastically reduced.

ensure that the active values are pairwise separated and therefore the partitioning will lead to an improvement (Bertsimas & Dunning, 2016). Therefore, this process is repeated until some stopping criterion has been met, such as a maximum amount of subsets or iterations.

Since the sequence of partitions is nested, and different cells are not prohibited from using the same second-stage solutions, every previous solution is also feasible for the subsequent partitions. It follows that with this partitioning scheme, the optimal solution can never deteriorate, only improve across iterations. The algorithm will thus, in every iteration, provide an increasingly adaptable solution, such that the decision-maker may stop the algorithm at any time according to their assessment.

When partitioning as described in algorithm 3, every iteration yields m new cells in each of the former cells, one for each active parameter, so that in iteration t we obtain m^t cells. To decelerate this fast growth in cell numbers, one simple modification, suggested in Bertsimas and Dunning (2016), suffices: We place a condition on the re-partitioning of a cell k. The cell is only partitioned further if it is active, i.e., its objective

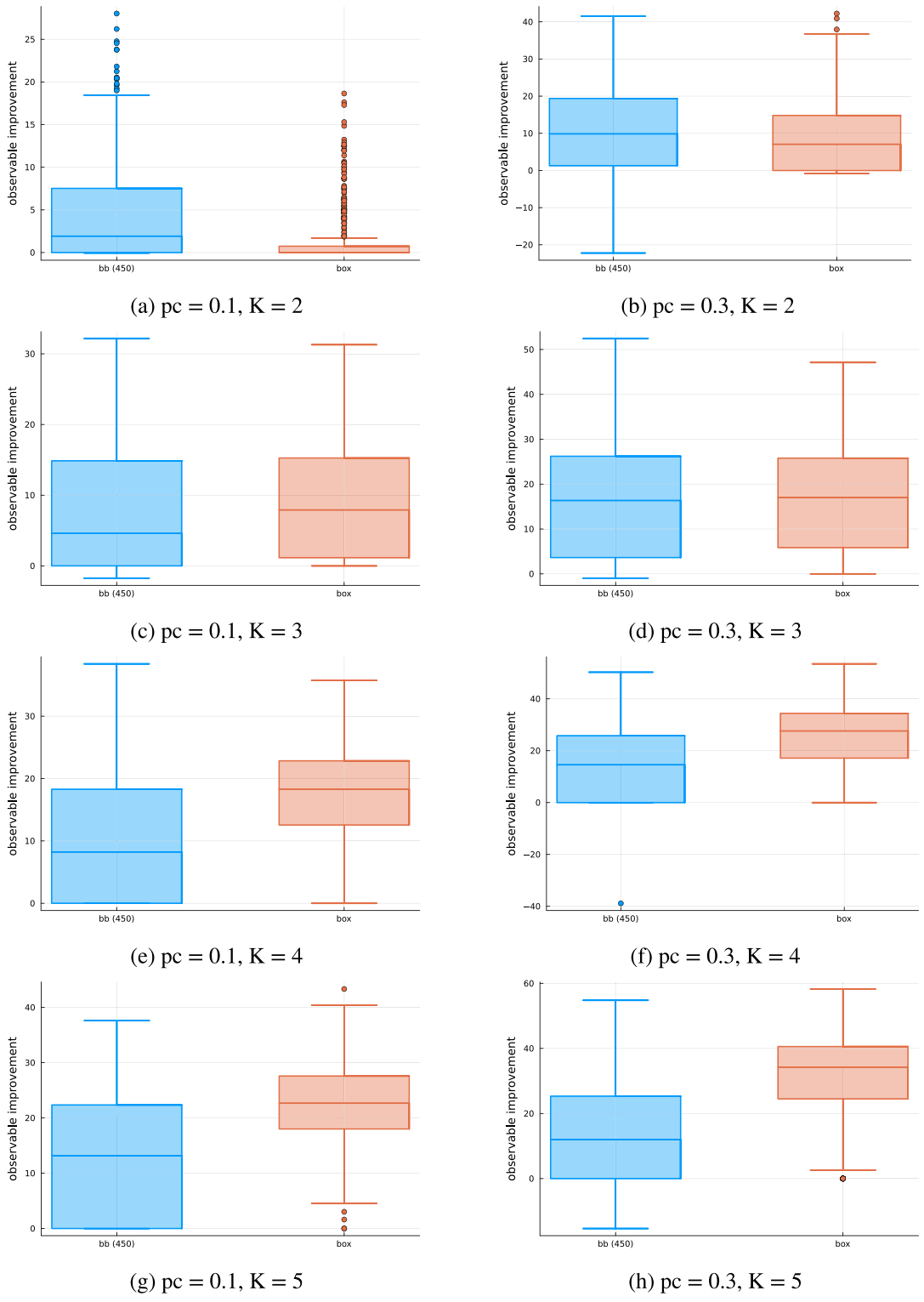


Fig. 14. Observable objective value relative to the static solution for different combinations of pc and K.

z_k equals the optimal objective of (5a), that is if $k = \text{argmax}_{k=1,\dots,K} z_k$. With this condition, the increase in the number of cells is roughly linear, as empirically observed in Section 4, and the iterative improvements of the objective values remain almost unchanged. Since now the objectives of all cells z_k, $k = 1, \dots, K$, gain importance, as opposed to only the worst one (i.e, cell $k = \text{argmax}_{k=1,\dots,K} z_k$), we motivate the model to minimize all z_k (and not only the worst one) by adding a term containing all z_k to the objective constraint (5b):

$$z \geq z_k + \varepsilon \sum_{k=1}^{K} z_k, \quad k = 1, \dots, K.$$

The weight ε hereby secures robustness in the traditional sense, that is, the optimization of the worst case, by discounting the sum, thus making the model prioritize the worst-case objective. To this end, for

$l = \text{argmin}_{k=1,\dots,K} z_k$, we choose $\epsilon > 0$ such that $\epsilon \sum_{k=1}^{K} z_k < z_l$, or

$$\epsilon < z_l \cdot \left(\sum_{k=1}^{K} z_k\right)^{-1} < z_l \cdot \left(\sum_{k=1}^{K} z_l\right)^{-1} = \frac{z_l}{K \cdot z_l} = \frac{1}{K} \tag{18}$$

throughout the algorithm. Thus, for any upper bound K' of K, we may set $\epsilon = \frac{1}{K'}$.

For a continuous uncertainty set, this algorithm is not guaranteed to converge in a finite number of iterations, even if the completely adaptable solution is better than the static one and attainable with a finite partition (see Bertsimas and Dunning (2016), Proposition 3 for a counterexample). The benefits of this algorithm lie mainly in its ability to approximate the optimal completely adaptable solution by means of the hierarchical adaptability it offers. Too many cells equal the completely adaptable solution, which we want to avoid, but too few cells may not significantly improve the objective with respect to the static formulation. Navigating between these two extremes permits one to optimize the trade-off between efficiency and effectiveness according to the decision-maker's priorities. As a downside, the exact value of K cannot be chosen beforehand but is instead determined by the algorithm. In addition, it can only be increased in steps of m, which, in our context, is usually a large number.

Appendix B. Additional results

This appendix contains some additional graphical representations of the results from the computational study in Section 4 (see Fig. 12).

References

Alem, D., Bonilla-Londono, H. F., Barbosa-Povoa, A. P., Relvas, S., Ferreira, D., & Moreno, A. (2021). Building disaster preparedness and response capacity in humanitarian supply chains using the social vulnerability index. *European Journal of Operational Research*, *292*(1), 250–275. http://dx.doi.org/10.1016/j.ejor.2020.10.016, URL https://www.sciencedirect.com/science/article/pii/S0377221720308924.

Alem, D., Clark, A., & Moreno, A. (2016). Stochastic network models for logistics planning in disaster relief. *European Journal of Operational Research*, *255*(1), 187–206. http://dx.doi.org/10.1016/j.ejor.2016.04.041, URL https://www.sciencedirect.com/science/article/pii/S0377221716302788.

Alexander, D. (1993). *Natural disasters*. Dordrecht: Springer.

Aliakbari, A., Komijan, A., Tavakkoli-Moghaddam, R., & Najafi, E. (2022). A new robust optimization model for relief logistics planning under uncertainty: a real-case study. *Soft Computing*, *26*, http://dx.doi.org/10.1007/s00500-022-06823-4.

Arslan, A. N., Poss, M., & Silva, M. (2022). Min-Sup-Min Robust Combinatorial Optimization with Few Recourse Solutions. *INFORMS Journal on Computing*, *34*(4), 2212–2228. http://dx.doi.org/10.1287/ijoc.2021.1156, URL https://ideas.repec.org/a/inm/orijoc/v34y2022i4p2212-2228.html.

Avishan, F., Elyasi, M., Yanıkoğlu, İ., Ekici, A., & Özener, O. Ö. (2023). Humanitarian relief distribution problem: An adjustable robust optimization approach. *Transportation Science*, *57*(4), 1096–1114. http://dx.doi.org/10.1287/trsc.2023.1204, arXiv:https://doi.org/10.1287/trsc.2023.1204.

Bartle, R., & Sherbert, D. (1994). Introduction to real analysis. *Math. Gazette*, *78*, http://dx.doi.org/10.2307/3619462.

Beck, Y., Ljubić, I., & Schmidt, M. (2023). A survey on bilevel optimization under uncertainty. *European Journal of Operational Research*, *311*(2), 401–426. http://dx.doi.org/10.1016/j.ejor.2023.01.008, URL https://www.sciencedirect.com/science/article/pii/S0377221723000073.

Ben-Tal, A., Chung, B. D., Mandala, S. R., & Yao, T. (2011). Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. *Transportation Research Part B: Methodological*, *45*(8), 1177–1189. http://dx.doi.org/10.1016/j.trb.2010.09.002, Supply chain disruption and risk management.

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). Robust optimization. In *Princeton series in applied mathematics*, Princeton University Press.

Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, *99*, 351–376. http://dx.doi.org/10.1007/s10107-003-0454-y.

Bertsimas, D., & Caramanis, C. (2010). Finite adaptability in multistage linear optimization. *IEEE Transactions on Automatic Control*, *55*(12), 2751–2766. http://dx.doi.org/10.1109/TAC.2010.2049764.

Bertsimas, D., & Dunning, I. (2016). Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research*, *64*(4), 980–998. http://dx.doi.org/10.1287/opre.2016.1515, arXiv:https://doi.org/10.1287/opre.2016.1515.

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98.

Birge, J. R., & Louveaux, F. (2011). *Introduction to stochastic programming*. New York, NY, USA: Springer-Verlag.

Buchheim, C., & Kurtz, J. (2017). Min—max—min robust combinatorial optimization. *Mathematical Programming*, *163*(1–2), 1–23. http://dx.doi.org/10.1007/s10107-016-1053-z.

Buchheim, C., & Kurtz, J. (2018). Complexity of min–max–min robustness for combinatorial optimization under discrete uncertainty. *Discrete Optimization*, *28*, 1–15. http://dx.doi.org/10.1016/j.disopt.2017.08.006, URL https://www.sciencedirect.com/science/article/pii/S1572528617302268.

Buchheim, C., & Pruente, J. (2019). K-adaptability in stochastic combinatorial optimization under objective uncertainty. *European Journal of Operational Research*, *277*(3), 953–963. http://dx.doi.org/10.1016/j.ejor.2019.03.045, URL https://www.sciencedirect.com/science/article/pii/S0377221719303030.

Caunhye, A. M., Nie, X., & Pokharel, S. (2012). Optimization models in emergency logistics: A literature review. *Socio-Economic Planning Sciences*, *46*(1), 4–13. http://dx.doi.org/10.1016/j.seps.2011.04.004, Special Issue: Disaster Planning and Logistics: Part 1.

Caunhye, A. M., Zhang, Y., Li, M., & Nie, X. (2016). A location-routing model for prepositioning and distributing emergency supplies. *Transportation Research Part E: Logistics and Transportation Review*, *90*, 161–176. http://dx.doi.org/10.1016/j.tre.2015.10.011, Risk Management of Logistics Systems.

Chassein, A., Goerigk, M., Kurtz, J., & Poss, M. (2019). Faster algorithms for min-max-min robustness for combinatorial problems with budgeted uncertainty. *European Journal of Operational Research*, *279*(2), 308–319. http://dx.doi.org/10.1016/j.ejor.2019.05.045, URL https://www.sciencedirect.com/science/article/pii/S0377221719304758.

CRED (2020). *Natural disasters 2019: Now is the time to not give up*. CRED.

CRED (2022). *2021 Disasters in numbers*. CRED.

CRED (2023). *2022 Disasters in numbers*. CRED.

CRED, & UNISDR (2020). *The human cost of disasters 2000–2019*. UN.

Dias Garcia, J., Bodin, G., & Street, A. (2022). BilevelJuMP.jl: Modeling and solving bilevel optimization in julia. arXiv preprint arXiv:2205.02307.

Duran, S., Gutierrez, M. A., & Keskinocak, P. (2011). Pre-positioning of emergency items for CARE international. *INFORMS Journal on Applied Analytics*, *41*(3), 223–237. http://dx.doi.org/10.1287/inte.1100.0526, arXiv:https://pubsonline.informs.org/doi/pdf/10.1287/inte.1100.0526.

Ghahtarani, A., Saif, A., Ghasemi, A., & Delage, E. (2023). A double-oracle, logic-based benders decomposition approach to solve the K-adaptability problem. *Computers & Operations Research*, *155*, Article 106243. http://dx.doi.org/10.1016/j.cor.2023.106243, URL https://www.sciencedirect.com/science/article/pii/S0305054823001077.

Gurobi Optimization, LLC (2023). Gurobi Optimizer Reference Manual. URL https://www.gurobi.com.

Han, E., Bandi, C., & Nohadani, O. (2022). On finite adaptability in two-stage distributionally robust optimization. *Operations Research*, http://dx.doi.org/10.1287/opre.2022.2273.

Hanasusanto, G. A., Kuhn, D., & Wiesemann, W. (2015). K-adaptability in two-stage robust binary programming. *Operations Research*, *63*(4), 877–891. http://dx.doi.org/10.1287/opre.2015.1392, arXiv:https://doi.org/10.1287/opre.2015.1392.

Hanasusanto, G. A., Kuhn, D., & Wiesemann, W. (2016). K-adaptability in two-stage distributionally robust binary programming. *Operations Research Letters*, *44*(1), 6–11. http://dx.doi.org/10.1016/j.orl.2015.10.006, URL https://www.sciencedirect.com/science/article/pii/S0167637715001376.

Ke, G. Y. (2022). Managing reliable emergency logistics for hazardous materials: A two-stage robust optimization approach. *Computers & Operations Research*, *138*(C), http://dx.doi.org/10.1016/j.cor.2021.105557.

Kurtz, J. (2023). Approximation algorithms for min-max-min robust optimization and K-adaptability under objective uncertainty. URL https://arxiv.org/abs/2106.03107, arXiv:2106.03107.

Li, Y., & Liu, Y. (2023). Distributionally robust optimization for collaborative emergency response network design. *Transportation Research Part E: Logistics and Transportation Review*, *176*, Article 103221. http://dx.doi.org/10.1016/j.tre.2023.103221.

Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B., & Vielma, J. P. (2023). JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, http://dx.doi.org/10.1007/s12532-023-00239-3.

Malaguti, E., Monaci, M., & Pruente, J. (2022). K-adaptability in stochastic optimization. *Mathematical Programming*, *196*(1), 567–595. http://dx.doi.org/10.1007/s10107-021-01767-3.

Masson-Delmotte, V., Zhai, P., Pirani, A., Connors, S., Péan, C., Berger, S., et al. (2021). Summary for policymakers. In *Climate change 2021: The physical science basis. Contribution of working group I to the sixth assessment report of the intergovernmental panel on climate change*. IPCC.

Postek, K., & Hertog, D. d. (2016). Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, *28*(3), 553–574. http://dx.doi.org/10.1287/ijoc.2016.0696, arXiv:https://doi.org/10.1287/ijoc.2016.0696.

Sabbaghtorkan, M., Batta, R., & He, Q. (2020). Prepositioning of assets and supplies in disaster operations management: Review and research gap identification. *European Journal of Operational Research*, *284*(1), 1–19. http://dx.doi.org/10.1016/j.ejor.2019.06.029.

Safaei, A. S., Farsad, S., & Paydar, M. M. (2017). Robust bi-level optimization of relief logistics operations. *Applied Mathematical Modelling*, *56*, 359–380. http://dx.doi.org/10.1016/j.apm.2017.12.003.

Subramanyam, A., Gounaris, C. E., & Wiesemann, W. (2020). K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation*, *12*(2), 193–224. http://dx.doi.org/10.1007/s12532-019-00174-2.

Sun, H., Wang, Y., & Xue, Y. (2021). A bi-objective robust optimization model for disaster response planning under uncertainties. *Computers & Industrial Engineering*, *155*, Article 107213. http://dx.doi.org/10.1016/j.cie.2021.107213.

Tippong, D., Petrovic, S., & Akbari, V. (2022). A review of applications of operational research in healthcare coordination in disaster management. *European Journal of Operational Research*, *301*(1), 1–17. http://dx.doi.org/10.1016/j.ejor.2021.10.048, URL https://www.sciencedirect.com/science/article/pii/S0377221721008973.

Van Wassenhove, L. (2006). Humanitarian aid logistics: Supply chain management in high gear. *Journal of the Operational Research Society - J OPER RES SOC*, *57*, 475–489. http://dx.doi.org/10.1057/palgrave.jors.2602125.

Wang, D., Yang, K., Yang, L., & Dong, J. (2023). Two-stage distributionally robust optimization for disaster relief logistics under option contract and demand ambiguity. *Transportation Research Part E: Logistics and Transportation Review*, *170*, Article 103025. http://dx.doi.org/10.1016/j.tre.2023.103025.

Weller, P. (2024). K-adaptable robust optimization for the pre-allocation of emergency supplies. https://github.com/paulaweller/k-adaptable-er,

Yanıkoğlu, İ., Gorissen, B. L., & den Hertog, D. (2019). A survey of adjustable robust optimization. *European Journal of Operational Research*, *277*(3), 799–813. http://dx.doi.org/10.1016/j.ejor.2018.08.031.