

K-Adaptability in Stochastic Integer Optimization with Recourse

Arnav Aakash, Jim Luedtke, Carla Michini

Department of Industrial and Systems Engineering, University of Wisconsin–Madison
aaakash@wisc.edu, jim.luedtke@wisc.edu, michini@wisc.edu

October 2, 2026

Abstract

Many optimization problems solved repeatedly in practice, such as in power systems or logistics, share a fixed structure but with different parameter values. Re-optimizing each instance from scratch can be computationally prohibitive, while adopting a single solution for all realizations is overly conservative. The K -adaptability paradigm offers a flexible compromise by precomputing a small set of candidate solutions and selecting the best once uncertainty is revealed. When K is small, this approach enables real-time decision making and provides human decision-makers with a manageable set of implementable solutions. However, it is often impossible to identify a small number of representative solutions that remain feasible across all parameter realizations. We introduce K -adaptability with recourse, in which the goal is to precompute K candidate assignments for the *discrete variables* while allowing the *continuous* recourse decisions to be optimized online after uncertainty is observed. This structure preserves online computational tractability and implementability while substantially improving adaptability. We present formulations of this model and develop heuristic and exact methods to solve these formulations. An extensive computational analysis compares the performance of the proposed algorithms on single and multi-dimensional knapsack and capacitated facility-location problem classes.

Keywords: Stochastic integer optimization, K-Adaptability, Parametric optimization

1 Introduction

We study a general class of parametric optimization problems of the form

$$f^*(\omega) = \min c(\omega)^\top x + d(\omega)^\top y \quad (1a)$$

$$\text{s.t. } T(\omega)x + W(\omega)y \geq h(\omega), \quad (1b)$$

$$x \in X, y \in \mathbb{R}^n, \quad (1c)$$

where ω denotes a realization of a random variable ω defined on Ω , $c(\omega) \in \mathbb{R}^p$, $d(\omega) \in \mathbb{R}^n$, $T(\omega) \in \mathbb{R}^{m \times p}$, $W(\omega) \in \mathbb{R}^{m \times n}$, $h(\omega) \in \mathbb{R}^m$, and $X = \{x \in \{0, 1\}^p : Ax \geq b\}$ is a binary set.

Many optimization problems that are solved repeatedly over time can be cast as problem (1). These problems typically display a fixed underlying structure, but with varying parameters based on the current system state. Ideally, one would solve (1) to obtain an optimal solution for each realization of ω , yielding the “wait-and-see” benchmark: $WS := \mathbb{E}_\omega[f^*(\omega)]$. However, in many applications it is not feasible to fully adapt the solution in response to ω , or even if it is, solving (1) may be too time-consuming. Examples include dynamic facility location, with binary facility openings and continuous resource assignments (Laporte et al. 1994); optimal transmission switching, with line-status decisions adapted to changing system conditions (Misra et al. 2022); and dynamic vehicle routing, with routes updated as new information becomes available

(Pillac et al. 2013). A common alternative is a *static* model that selects a single solution (x, y) minimizing expected cost while remaining feasible for every realization $\omega \in \Omega$. A more flexible approach is provided by the *K-adaptability* paradigm, in which one precomputes offline a *candidate set* of at most K solutions and then, after the realization of the uncertain parameters, selects the best feasible solution in the candidate set. This framework was first introduced for adjustable robust optimization by Bertsimas and Caramanis (2010) and was further developed in that setting by Hanasusanto et al. (2015) and Subramanyam et al. (2020). Buchheim and Prunte (2019) extended the approach to stochastic optimization with random objective, and Malaguti et al. (2022) further extended it to stochastic optimization with random constraints, introducing the *K-adaptable static* problem:

$$\min_{\mathcal{F} \subseteq X \times \mathbb{R}^n: |\mathcal{F}| \leq K} \mathbb{E}_\omega \left[\min \{ c(\omega)^\top x + d(\omega)^\top y : (x, y) \in \mathcal{F} \cap F(\omega) \} \right], \quad (K\text{-ST})$$

where $F(\omega)$ represents the set of (x, y) that satisfy (1b) and $\mathcal{F}$ represents the candidate set. This model offers more flexibility than the static approach, since after observing a realization ω , one can select the best solution from the candidate set rather than being restricted to a single solution. It also preserves fast decision-making, as comparing objective values and checking feasibility across candidates is computationally cheap. However, this model still requires that, for every possible scenario, at least one solution in the candidate set is feasible, a condition that can be difficult to meet for small or even moderate K .

An alternative approach for modeling problems with limited recourse is the classic framework of stochastic optimization with recourse, where some variables model decisions that must be made “here-and-now”, as in the static model, and the other variables represent decisions that may be made after observing the realization of the parameters, as in the wait-and-see benchmark. Considering the binary variables x as the “here-and-now” variables, we define for $\omega \in \Omega$ and $x \in X$

$$f(\omega, x) = c(\omega)^\top x + \min_{y \in \mathbb{R}^n} d(\omega)^\top y \quad (2a)$$

$$\text{s.t. } W(\omega)y \geq h(\omega) - T(\omega)x, \quad (2b)$$

where we adopt the convention that $f(\omega, x) = +\infty$ if (2) is infeasible. The stochastic optimization model with recourse (SR) minimizes $\mathbb{E}_\omega[f(\omega, x)]$ over $x \in X$.

In this paper, we introduce the *K-adaptable stochastic optimization problem with recourse*:

$$\min_{\mathcal{X} \subseteq X: |\mathcal{X}| \leq K} \mathbb{E}_\omega \left[\min \{ f(\omega, x) : x \in \mathcal{X} \} \right] \quad (K\text{-SR})$$

as an extension of the *K-adaptable static* model proposed in Malaguti et al. (2022). This model integrates the advantages of *K-adaptability* and stochastic optimization with recourse: it allows full adaptability of the continuous variables y to the realization ω , while allowing a choice among the candidate set $\mathcal{X}$ for the binary solution. It thus shifts the trade-off between computational complexity and solution flexibility significantly toward flexibility, at a moderate increase in implementation complexity and online computational effort. A related but different stochastic two-stage extension of the *K-adaptable static* model is proposed by De Santis et al. (2025), where a single first-stage vector is selected together with a candidate set of second-stage solutions, and after a realization is observed the best feasible solution among the candidate set is implemented. In contrast, we select only a candidate set of K binary solutions, and after a realization is observed, we both select the best among these candidates and optimize the continuous recourse variables. Merging these models (e.g., adding to our model a single first-stage vector to be optimized along with the selection of the candidate set of binary solutions) would be an interesting direction for future work.

Related literature. Finite adaptability has also been studied in multistage robust optimization: Bertsimas and Georghiou (2015) design near-optimal restricted decision rules, whereas Bertsimas and Dunning (2016)

and Postek and den Hertog (2016) adaptively refine the uncertainty-set partition instead of fixing the number of solutions. Han et al. (2023) extend finite adaptability to two-stage distributionally robust optimization using orthant-based policies with performance guarantees. When there is no common first-stage decision, the problem takes a min–max–min form, formalized by Buchheim and Kurtz (2017); subsequent work studies its complexity under discrete uncertainty (Buchheim and Kurtz 2018), discrete budgeted uncertainty (Goerigk et al. 2020), and exact solution by row-and-column generation (Arslan et al. 2022). Related methods for general K -adaptability include double-oracle, logic-based Benders decomposition (Ghahtarani et al. 2023) and learning-based B&B node selection (Julien et al. 2025). Kurtz (2026) derives bounds on the size of the candidate set sufficient for optimality, while Kurtz (2024) provides guarantees under objective uncertainty. *Our contributions.* We introduce the new model (K -SR) and derive several methods to solve it. In Section 2 we introduce two compact formulations: one extends the model in Malaguti et al. (2022) for the K -adaptable static case to the case with recourse, and the other is new and more compact. We also define two exponential-sized formulations—the subset and the assignment formulation—that extend two formulations introduced in Malaguti et al. (2022) to the setting with recourse.

In Section 3.1, we derive a scenario-based heuristic in which we first generate a sample of realizations from the parameter distribution and solve problem (1) for each realization. This produces a solution pool, from which we select the candidate set by solving the assignment formulation. This approach is similar to the “refinement heuristic” recently proposed in Malaguti et al. (2026), with the key difference being our proposed method for generating the solution pool. By design, each of the selected K representatives is optimal for at least one parameter realization. We prove that restricting attention to solutions generated in this way is sufficient when only $c(\omega)$ is parametric and the set of attainable cost vectors $c(\Omega)$ is convex, but also show it is not sufficient in case the right-hand side of the constraints are parametric.

In Section 3.2, we extend the column generation approach for the subset formulation of Malaguti et al. (2022) to the setting with recourse and in Section 3.3 we introduce a column-and-constraint generation (CCG) method for solving the LP relaxation of the assignment formulation. This CCG method is new even for the K -adaptable static problem. A key challenge for the CCG algorithm is the subproblem that must be solved to identify if any solution $v \in X$ needs to be added to the LP relaxation of the assignment formulation. We propose a MIP formulation and a specialized branch-and-bound algorithm for solving this subproblem.

Finally, we conduct a thorough computational study (Section 4) of our proposed model and algorithms. To investigate the value of our recourse model, we compare the candidate sets that are obtained by a model that allows constraints to be violated (as a recourse action, but penalized) with those obtained by a model that does not allow recourse. We find that the recourse model can yield candidate sets that have substantially better objective values with minor constraint violations. We also find that in instances in which recourse variables are present that guarantee the model is feasible for any possible scenario, our proposed scenario-based heuristic consistently produces near-optimal solutions. Our new compact formulation substantially outperforms the natural extension of the compact formulation proposed in Malaguti et al. (2022) for the model without recourse. We also demonstrate that the column-and-constraint generation algorithm for solving the assignment formulation outperforms the column generation algorithm for solving the subset formulation.

Notation. We denote by $\mathbf{1}$ a vector of all ones. For a positive integer n , we use $[n]$ to denote the set $\{1, \dots, n\}$ and $2^{[n]}$ to denote the set of all subsets of $[n]$.

2 Formulations

We develop methods that solve problem (K -SR) over a fixed, finite equally likely set of scenarios. If the distribution of ω is continuous or the support is too large, this set of scenarios may be obtained by sample average approximation (SAA; see, e.g., Shapiro et al. (2014)), sampling N scenarios to define the finite support and replacing the expectation in (K -SR) by the empirical average over these scenarios. For notational

simplicity we define $(c^s, d^s, h^s, T^s, W^s) = (c(\omega^s), d(\omega^s), h(\omega^s), T(\omega^s), W(\omega^s))$ for $s \in [N]$. We make the following assumptions throughout this paper: (i) *scenario feasibility*: the problem (1) has a feasible solution for every $\omega \in \Omega$ and (ii) *subproblem dual feasibility*: the set $\{\eta \in \mathbb{R}^m : W(\omega)^\top \eta = d(\omega), \eta \geq 0\} \neq \emptyset$ for all $\omega \in \Omega$. Subproblem dual feasibility assures the minimization in (2) is never unbounded.

2.1 Compact formulations

We describe two explicit mixed-integer programming formulations for (K-SR). We refer to these as “compact” because the number of decision variables is polynomial in p, n, K , and N .

We first extend the compact formulation of Malaguti et al. (2022) to the setting with recourse. The first-stage variables in this formulation are x^i for $i \in [K]$, encoding the K solutions of the candidate set that are selected. Then, for each $s \in [N]$, indicator variables w_i^s for $i \in [K]$ are used to choose which solution in the candidate set is used in scenario ω^s and y^{is} denotes the recourse decision taken in scenario s if candidate solution x^i is selected. Using the definition of $f(\omega^s, x^i)$ for $i \in [K]$ and $s \in [N]$ yields the formulation :

$$\min_{x, w, y} \frac{1}{N} \sum_{i \in [K]} \sum_{s \in [N]} ((c^s)^\top x^i + (d^s)^\top y^{is}) w_i^s \quad (3a)$$

$$\text{s.t. } w_i^s = 1 \Rightarrow T^s x^i + W^s y^{is} \geq h^s, \quad \forall s \in [N], \forall i \in [K], \quad (3b)$$

$$\mathbf{1}^\top w^s = 1, w^s \in \{0, 1\}^K, \quad \forall s \in [N], \quad (3c)$$

$$x^i \in X, \quad \forall i \in [K]. \quad (3d)$$

Malaguti et al. (2022) linearize the objective by introducing auxiliary variables and Fortet inequalities for the product terms, and reformulate constraints (3b) using big- M inequalities. The resulting mixed-integer linear formulation has $O(NK(n + p))$ additional variables and constraints.

We next present a more compact formulation that avoids linearizing the product terms in the objective. It is inspired by the nonanticipative formulation of stochastic optimization with recourse, in which a copy q^s of the first-stage variables x is created for each $s \in [N]$ and all copies are forced to equal the common variables x . Here, however, we enforce $q^s = x^i$ only for the *selected* representative, i.e., the $i \in [K]$ with $w_i^s = 1$. This requires only a single set of recourse variables y^s per scenario, yielding the following formulation:

$$\min_{x, q, w, y} \frac{1}{N} \sum_{s \in [N]} ((c^s)^\top q^s + (d^s)^\top y^s) \quad (4a)$$

$$\text{s.t. } T^s q^s + W^s y^s \geq h^s, \quad \forall s \in [N], \quad (4b)$$

$$\mathbf{1}^\top w^s = 1, w^s \in \{0, 1\}^K, \quad \forall s \in [N], \quad (4c)$$

$$(w_i^s - 1)\mathbf{1} \leq x^i - q^s \leq (1 - w_i^s)\mathbf{1}, \quad \forall i \in [K], s \in [N], \quad (4d)$$

$$x^i \in X, \quad \forall i \in [K], \quad (4e)$$

$$q^s \in \mathbb{R}^p, y^s \in \mathbb{R}^n, \quad \forall s \in [N]. \quad (4f)$$

Constraints (4d) enforce that $q^s = x^i$ when $w_i^s = 1$, (4b) enforce that y^s is a feasible recourse action with respect to the selected first-stage vector, and therefore the objective correctly records the cost of each scenario s without the need to sum over $i \in [K]$ and multiply with w_i^s .

2.2 Subset Formulation

In this section we introduce a *subset formulation* (SF), extending the subset formulation of Malaguti et al. (2022) to the recourse setting. The idea is to partition the scenario set into at most K subsets and “cover” all

scenarios in a subset with a single solution x . The formulation then selects a partition that minimizes (4a). For each $S \in 2^{[N]}$ define

$$g(S) = \min_{x, \{y^s\}_{s \in S}} \sum_{s \in S} ((c^s)^\top x + (d^s)^\top y^s) \quad (5a)$$

$$T^s x + W^s y^s \geq h^s, \quad \forall s \in S, \quad (5b)$$

$$x \in X, \quad (5c)$$

where $g(S) = +\infty$ if the problem is infeasible. We also define a binary variable γ_S that is one if all the scenarios in S are covered by the same solution x (an optimal solution of (5)). We obtain

$$\min \frac{1}{N} \sum_{S \in 2^{[N]}} g(S) \gamma_S \quad (6a)$$

$$\sum_{S: s \in S} \gamma_S = 1, \quad \forall s \in [N], \quad (6b)$$

$$\sum_{S \in 2^{[N]}} \gamma_S \leq K, \quad (6c)$$

$$\gamma_S \in \{0, 1\}, \quad \forall S \in 2^{[N]}. \quad (6d)$$

Constraints (6b) ensure that every scenario belongs to exactly one selected subset, and (6c) ensures that at most K subsets are selected. This formulation is impractical to use directly due to the large number of decision variables and constraints, that grow exponentially with N . In Section 3.2 we present a column generation algorithm for solving the LP relaxation of this formulation.

2.3 Assignment Formulation

We next introduce an extension of the assignment formulation of Malaguti et al. (2022) to the setting with recourse. The formulation selects at most K vectors from X and assigns each scenario to one selected vector. For each $v \in X$, let σ_v indicate whether v is selected, and for each $s \in [N]$, let ρ_{sv} indicate whether scenario s uses v . The value $f(\omega^s, v)$ is the optimal recourse cost under v and scenario s ; accordingly, define $X_s = \{v \in X : f(\omega^s, v) < +\infty\}$ as the vectors admitting feasible recourse in scenario s . This yields:

$$\min \frac{1}{N} \sum_{s \in [N]} \sum_{v \in X_s} f(\omega^s, v) \rho_{sv} \quad (7a)$$

$$\sum_{v \in X_s} \rho_{sv} = 1, \quad \forall s \in [N], \quad (7b)$$

$$\rho_{sv} \leq \sigma_v, \quad \forall v \in X_s, \quad \forall s \in [N], \quad (7c)$$

$$\sum_{v \in X} \sigma_v \leq K, \quad (7d)$$

$$\sigma_v \in \{0, 1\}, \quad \forall v \in X, \quad (7e)$$

$$\rho_{sv} \in \{0, 1\}, \quad \forall v \in X_s, \quad s \in [N]. \quad (7f)$$

Constraint (7d) ensures that we select at most K first-stage solution vectors for the candidate set, constraints (7b) impose that we assign each scenario to exactly one solution in the candidate set, and constraints (7c) require that the solution selected for each scenario is chosen among the solutions in the candidate set. This formulation is impractical to use directly due to the number of decision variables and constraints depending on $|X|$. In Section 3.1 we propose a heuristic technique for choosing a subset of X to restrict attention to. In Section 3.3 we present a column and constraint generation algorithm for solving the LP relaxation of this formulation.

3 Solution methods

3.1 Scenario-based heuristic

The assignment formulation (7) motivates a natural heuristic: replace the set of solutions X used in the formulation with a tractable-size subset, say Q . For the static K -adaptable problem, Malaguti et al. (2026) propose to construct Q by using a heuristic column generation approach and then to use these columns within the assignment formulation in a step they refer to as the refinement step.

We propose a simple alternative for generating the set Q , motivated by the scenario decomposition algorithm of Ahmed (2013). The idea is to generate L samples $\omega^1, \dots, \omega^L$ of ω . For each $i \in [L]$ we solve problem (1) with $\omega = \omega^i$, let $(\hat{x}^i, \hat{y}^i)$ be an optimal solution, and set $Q = \{\hat{x}^i : i \in [L]\}$. That is, we restrict our attention to binary assignments that are optimal to some *individual* scenario. When SAA is used to define the finite scenario set that approximates the expected value in (K-SR), the L samples used to generate the set Q may be drawn from the original distribution of ω .

Define $X^* \subseteq X$ as the vectors in X that are optimal binary assignments to (1) for some realization $\omega \in \Omega$, so that $Q \subseteq X^*$ by construction and Q approximates X^* as $L \rightarrow \infty$. Even for $K = 1$, the optimal solution of (K-SR) may not be obtainable from X^* (Ahmed 2013). The following theorem identifies a case in which restricting attention to X^* is sufficient.

Theorem 1. *Suppose that in problem (1) only $c(\omega)$ is parametrized by realizations of ω , while $d(\omega) = d$, $h(\omega) = h$, $T(\omega) = T$ and $W(\omega) = W$ are fixed inputs. Let $\bar{X}$ be a minimum-cardinality optimal solution of (K-SR). If the set $c(\Omega) = \{c(\omega) : \omega \in \Omega\}$ is convex, then $\bar{X} \subseteq X^*$.*

Proof of Theorem 1. Let $\bar{X} = \{\bar{x}^1, \dots, \bar{x}^k\}$ with $k \leq K$ be a minimum-cardinality subset of X for which the minimum of (K-SR) is achieved, and for all $i \in [k]$ define

$$\Omega_i = \left\{ \omega \in \Omega : f(\omega, \bar{x}^i) \leq f(\omega, \bar{x}^j) \forall j \in [k], \text{ and } f(\omega, \bar{x}^i) < f(\omega, \bar{x}^j) \forall j < i \right\}.$$

The sets Ω_i , $i \in [k]$, form a partition of Ω . Moreover $\Pr(\omega \in \Omega_i) > 0$ for all $i \in [k]$. Indeed, suppose $\Pr(\omega \in \Omega_i) = 0$ for some i . For every $\omega \notin \Omega_i$ the minimum of $f(\omega, \cdot)$ over $\bar{X}$ is attained at some $\bar{x}^j$ with $j \neq i$, and therefore $\min\{f(\omega, x) : x \in \bar{X}\} = \min\{f(\omega, x) : x \in \bar{X} \setminus \{\bar{x}^i\}\}$. As the two sides differ only on the null set Ω_i , they have the same expectation, so $\bar{X} \setminus \{\bar{x}^i\}$ attains the minimum of (K-SR) as well, contradicting the minimum cardinality of $\bar{X}$.

We first argue that, for all $i \in [k]$, $\bar{x}^i$ minimizes $\mathbb{E}_\omega[f(\omega, x) | \omega \in \Omega_i]$ over $x \in X$. By contradiction, suppose there is $\hat{x} \in X$ such that

$$\mathbb{E}_\omega[f(\omega, \hat{x}) | \omega \in \Omega_\ell] < \mathbb{E}_\omega[f(\omega, \bar{x}^\ell) | \omega \in \Omega_\ell] \quad (8)$$

for some $\ell \in [k]$. Let $\hat{X} = \bar{X} \setminus \{\bar{x}^\ell\} \cup \{\hat{x}\}$. Conditioning on the sets Ω_i , $i \in [k]$, we obtain

$$\mathbb{E}_\omega \left[\min\{f(\omega, x) : x \in \hat{X}\} \right] = \sum_{i \in [k]} \mathbb{E}_\omega \left[\min\{f(\omega, x) : x \in \hat{X}\} | \omega \in \Omega_i \right] \Pr(\omega \in \Omega_i).$$

Let $i \in [k] \setminus \{\ell\}$. For all $\omega \in \Omega_i$, $\min\{f(\omega, x) : x \in \hat{X}\} \leq f(\omega, \bar{x}^i) = \min\{f(\omega, x) : x \in \bar{X}\}$, thus

$$\mathbb{E}_\omega \left[\min\{f(\omega, x) : x \in \hat{X}\} | \omega \in \Omega_i \right] \leq \mathbb{E}_\omega \left[\min\{f(\omega, x) : x \in \bar{X}\} | \omega \in \Omega_i \right].$$

When $i = \ell$ we obtain

$$\mathbb{E}_\omega \left[\min\{f(\omega, x) : x \in \hat{X}\} | \omega \in \Omega_\ell \right] \leq \mathbb{E}_\omega \left[f(\omega, \hat{x}) | \omega \in \Omega_\ell \right]$$

$$\begin{aligned}
&< \mathbb{E}_\omega[f(\omega, \bar{x}^\ell) | \omega \in \Omega_\ell] \\
&= \mathbb{E}_\omega[\min\{f(\omega, x) : x \in \bar{X}\} | \omega \in \Omega_\ell],
\end{aligned}$$

where the strict inequality follows from (8) and the equality follows from the definition of Ω_ℓ . Summing up, we obtain $\mathbb{E}_\omega[\min\{f(\omega, x) : x \in \hat{X}\}] < \mathbb{E}_\omega[\min\{f(\omega, x) : x \in \bar{X}\}]$, which contradicts the optimality of $\bar{X}$ for (K-SR).

We have proved that, for all $i \in [k]$, $\bar{x}^i$ is an optimal solution to

$$\min_{x \in X} \mathbb{E}_\omega[f(\omega, x) | \omega \in \Omega_i]. \quad (9)$$

Under our assumptions, $f(\omega, x) = c(\omega)^\top x + \theta(x)$, where $\theta(x) = \min\{d^\top y : Wy \geq h - Tx\}$ is a function of x that does not depend on ω ; it takes the value $+\infty$ precisely when the recourse problem is infeasible for x , in which case x is never selected. Thus, we can rewrite (9) as

$$\min_{x \in X} (\mathbb{E}_\omega[c(\omega)^\top x | \omega \in \Omega_i] + \theta(x)) = \min_{x \in X} ((\bar{c}^i)^\top x + \theta(x)),$$

where $\bar{c}^i = \mathbb{E}_\omega[c(\omega) | \omega \in \Omega_i]$ (we assume $\mathbb{E}[\|c(\omega)\|] < \infty$, as is implicit in the sample-average approximation approach). Since $c(\omega) \in c(\Omega)$ for every realization ω , the conditional distribution of $c(\omega)$ given $\omega \in \Omega_i$ is contained in $c(\Omega)$, which is convex by assumption; hence its mean $\bar{c}^i$ belongs to $c(\Omega)$. Thus $\bar{x}^i$ minimizes $(\bar{c}^i)^\top x + \theta(x)$ over $x \in X$. Since $\bar{c}^i = c(\hat{\omega})$ for some $\hat{\omega} \in \Omega$, $\bar{x}^i$ is an optimal solution to (1) corresponding to such realization, implying $\bar{x}^i \in X^*$. $\square$

The convexity requirement of Theorem 1 implies it is applicable only when $c(\omega)$ does not have finite support. For such a case, it motivates the heuristic by showing that the optimal solution of the true problem lies in X^* , which Q approximates as L increases.

Homem-de Mello et al. (2024) have demonstrated a similar property holds for problems in which d, W are fixed but h and T may be random. They showed for continuous stochastic optimization problems with recourse that there exists a single scenario such that optimizing a deterministic version of the problem with that single scenario yields an optimal solution to the stochastic optimization problem. While intuitively supporting the potential for our scenario-based heuristic, this theory does not quite fit our framework because it does not apply with integer restrictions on the variables as we have, and the scenario in their construction is not guaranteed to lie within the support of the random variables, and hence may not be obtainable by random sampling.

The following proposition, whose proof is constructive and given in Appendix A, shows that no analogue of Theorem 1 can hold when the uncertainty enters the right-hand side $h(\omega)$, even when the set of attainable right-hand sides $h(\Omega)$ is compact and convex.

Proposition 1. *There exists an instance of (1) with $p = 2$, $n = 1$, $m = 3$, $X = \{0, 1\}^2$, and ω uniformly distributed on a compact convex set $\Omega \subset \mathbb{R}^2$, in which c, d, T, W are fixed and only $h(\omega)$ is parametric, the set $h(\Omega) = \{h(\omega) : \omega \in \Omega\}$ is compact and convex, and the unique optimal solution $\bar{x}$ of (K-SR) with $K = 1$ satisfies*

$$f(\omega, \bar{x}) \geq \min_{x \in X} f(\omega, x) + 1/16 \quad \text{for every realization } \omega \in \Omega.$$

Thus, the unique optimal solution $\bar{x}$ is not in X^ .*

Proposition 1 complements Theorem 1: although restricting attention to X^* is exact under objective uncertainty and a convex set of attainable cost vectors, it can fail under right-hand side uncertainty even for $K = 1$ (i.e., for the classical stochastic program with recourse (SR)), $X = \{0, 1\}^2$, and by a margin that is uniform over the support. Moreover, integrality of X plays no role in this phenomenon: in Corollary 1

in Appendix A we show that the same construction yields an instance in which X is replaced by a closed convex cone and the same behavior occurs. This continuous variant is also consistent with the result of Homem-de Mello et al. (2024), which applies to continuous problems, and shows that the single scenario constructed in their theory must, in general, lie outside the support of the random parameters (see Remark 2 in Appendix A.4).

3.2 Column Generation For Subset Formulation

For the static K -adaptable problem, Malaguti et al. (2022) proposed a column generation algorithm for solving the LP relaxation of (6). In this section, we describe how this can be extended to K -adaptable problem with recourse. Relaxing the binary variables in (6) and dropping the constant $1/N$ objective scaling gives the following dual:

$$\max_{\lambda, \mu} \sum_{s \in [N]} \lambda_s + K\mu \quad (10a)$$

$$\sum_{s \in S} \lambda_s + \mu \leq g(S), \quad \forall S \in 2^{[N]}, \quad (10b)$$

$$\lambda \in \mathbb{R}^N, \mu \leq 0. \quad (10c)$$

Here duals λ and μ correspond to constraints (6b) and (6c), respectively. The restricted master problem (RMP) retains only a subset of the γ_S variables. After solving the RMP to optimality and obtaining an associated optimal dual solution (λ^*, μ^*) , we check whether there is a new variable γ_S that has negative reduced cost, i.e., whose associated dual constraint is violated. Using the definition of $g(S)$ in (5), we conclude that the dual constraint associated to γ_S is violated if $g(S) - \sum_{s \in S} \lambda_s^* - \mu^* < 0$. In order to determine if such a subset S exists or not, we replace the definition of $g(S)$ from (5) and obtain the following pricing subproblem:

$$\min_{x, y, \pi} \sum_{s \in [N]} ((c^s)^\top x + (d^s)^\top y^s - \lambda_s^*) \pi_s - \mu^* \quad (11a)$$

$$\pi_s = 1 \implies T^s x + W^s y^s \geq h^s, \quad \forall s \in [N], \quad (11b)$$

$$x \in X, \pi \in \{0, 1\}^N. \quad (11c)$$

Here, the binary variables π_s encode the subset $S = \{s \in [N] : \pi_s = 1\}$; scenarios with $\pi_s = 0$ are unconstrained and contribute zero to the objective. As described in Malaguti et al. (2022) the constraints (11b) can be enforced via big- M inequalities or indicator constraints, and if the y variables have finite bounds, the objective can be linearized by introducing new variables and Fortet inequalities to represent the product terms. This leads to a mixed-integer linear reformulation of (11) which introduces $(n + p)N$ additional variables and $O((n + p)N)$ additional constraints.

We derive an alternative formulation of (11) that introduces a copy q^s of the first-stage x variables for each $s \in [N]$, eliminating big- M constraints and products of variables:

$$\min_{x, y, \pi, q} \sum_{s \in [N]} ((c^s)^\top q^s + (d^s)^\top y^s - \lambda_s^* \pi_s) - \mu^* \quad (12a)$$

$$x \in X, \pi \in \{0, 1\}^N \quad (12b)$$

$$T^s q^s + W^s y^s \geq h^s \pi_s, \quad \forall s \in [N], \quad (12c)$$

$$A q^s \geq b \pi_s, \quad \forall s \in [N], \quad (12d)$$

$$0 \leq q^s \leq \mathbf{1} \pi_s, \quad \forall s \in [N], \quad (12e)$$

$$-\mathbf{1}(1 - \pi_s) \leq x - q^s \leq \mathbf{1}(1 - \pi_s), \quad \forall s \in [N]. \quad (12f)$$

The proof of the following Lemma relies on the subproblem dual feasibility assumption (i.e., $\{\eta \in \mathbb{R}^m : (W^s)^\top \eta = d^s, \eta \geq 0\} \neq \emptyset$ for all $s \in [N]$).

Lemma 1. *Formulation (12) solves the pricing problem (11).*

Proof. Let $(\tilde{x}, \tilde{y}, \tilde{\pi})$ be feasible for (11), and define $\bar{x} = \tilde{x}$, $\bar{\pi} = \tilde{\pi}$, $\bar{q}^s = \tilde{\pi}_s \tilde{x}$, and $\bar{y}^s = \tilde{\pi}_s \tilde{y}^s$ for every $s \in [N]$. Constraint (12b) follows directly from the feasibility of $(\tilde{x}, \tilde{y}, \tilde{\pi})$. If $\tilde{\pi}_s = 1$, then $\bar{q}^s = \tilde{x}$ and $\bar{y}^s = \tilde{y}^s$; hence (12c) follows from (11b), (12d) follows from $\tilde{x} \in X$, and (12e)–(12f) hold by construction. If $\tilde{\pi}_s = 0$, then $\bar{q}^s = \bar{y}^s = 0$, so (12c)–(12e) hold at zero; moreover, (12f) holds because $\tilde{x} \in \{0, 1\}^P$. Thus, $(\bar{x}, \bar{y}, \bar{\pi}, \bar{q})$ is feasible for (12). Substitution in (11a) and (12a) shows that the two solutions have the same objective value. Thus, the optimal value of (12) is at most that of (11).

Conversely, let $(\bar{x}, \bar{y}, \bar{\pi}, \bar{q})$ be optimal for (12). For every s with $\bar{\pi}_s = 1$, constraint (12f) implies $\bar{q}^s = \bar{x}$, and (12c) therefore implies (11b). For s with $\bar{\pi}_s = 0$, implication (11b) imposes no restriction. Together with (12b), this shows that $(\bar{x}, \bar{y}, \bar{\pi})$ is feasible for (11). It remains to compare the objectives. When $\bar{\pi}_s = 1$, the contribution in (12a) equals that in (11a). When $\bar{\pi}_s = 0$, constraint (12e) gives $\bar{q}^s = 0$, and the remaining optimization over y^s is $\min\{(d^s)^\top y^s : W^s y^s \geq 0\}$. Subproblem dual feasibility gives $\eta^s \geq 0$ with $(W^s)^\top \eta^s = d^s$; hence every feasible y^s satisfies $(d^s)^\top y^s = (\eta^s)^\top W^s y^s \geq 0$. Since $y^s = 0$ is feasible, this problem has optimal value zero, and an optimal solution to (12) may be chosen with $\bar{y}^s = 0$ whenever $\bar{\pi}_s = 0$. Such scenarios contribute zero to both objectives. Consequently, $(\bar{x}, \bar{y}, \bar{\pi})$ has the same objective value in (11) as $(\bar{x}, \bar{y}, \bar{\pi}, \bar{q})$ has in (12). The optimal value of (11) is therefore no greater than that of (12). $\square$

3.3 Column and constraint generation algorithm

We introduce a column and constraint generation algorithm for solving the LP relaxation of (7). In each iteration, the algorithm solves a restricted master problem (RMP) of the form (7) but with X replaced with a (small) subset X' , e.g., as initialized from the scenario-based heuristic. The key step, analogous to pricing in a column generation algorithm, is to determine if there is any $v \in X$ such that adding the decision variables σ_v and ρ_{sv} for $s \in [N]$ (along with their connecting constraints) has potential to improve the RMP optimal value. If such v is identified, the associated variables and constraints are added and the RMP is re-solved. Otherwise, the algorithm terminates with an optimal solution of the LP relaxation.

By replacing the binary requirements in the assignment formulation (7) with nonnegativity constraints, and dropping the $1/N$ factor in the objective, we can write the dual of the continuous relaxation of (7) as follows:

$$\max_{\lambda, \tau, \mu} \sum_{s \in [N]} \lambda_s + K\mu \quad (13a)$$

$$\text{s.t. } \lambda_s - \tau_{sv} \leq f(\omega^s, v), \quad \forall v \in X_s, s \in [N], \quad (13b)$$

$$\mu + \sum_{s \in [N]} \tau_{sv} \leq 0, \quad \forall v \in X, \quad (13c)$$

$$\tau_{sv} \geq 0, \quad \forall v \in X_s, s \in [N], \quad (13d)$$

$$\mu \leq 0. \quad (13e)$$

Note that nonnegativity constraints and (7b) imply that $\rho \leq 1$. The σ variables can take larger values; however, this would never be advantageous, due to the cardinality constraint (7d). Here λ and τ are dual variables associated with constraints (7b) and (7c), respectively, and μ is associated with constraint (7d). In the dual of the RMP, X' replaces X in (13).

Let $(\{\lambda_s^*\}_{s \in [N]}, \{\tau_{sv}^*\}_{s \in [N], v \in X'}, \mu^*)$ be an optimal RMP dual solution. We wish to determine if there exists $v \in X \setminus X'$ such that some of the dual constraints (13b), (13c), (13d) indexed by v are violated. In

other words, we seek to determine if there exists $v \in X \setminus X'$ such that the system

$$\tau_{sv} \geq \max\{0, \lambda_s^* - f(\omega^s, v)\} \quad \forall s \in [N], \quad (14a)$$

$$\sum_{s \in [N]} \tau_{sv} \leq -\mu^*, \quad (14b)$$

is infeasible. Note that while $(\mu^*, \{\lambda_s^*\}_{s \in [N]})$ are fixed, the values of $\{\tau_{sv}\}_{s \in [N]}$ are not assigned, since they correspond to the primal constraints (7c) associated to $v \notin X'$, that were not included in the RMP. These primal constraints are active since we can set the primal variables $(\sigma_v, \{\rho_{sv}\}_{s \in [N]})$ to be nonbasic and therefore equal to zero. System (14) associated to v is infeasible if and only if

$$\sum_{s \in [N]} \max\{0, \lambda_s^* - f(\omega^s, v)\} + \mu^* > 0.$$

Thus, to check if there exists $v \in X \setminus X'$ associated to violated dual constraints, we solve

$$\max_{x \in X} \sum_{s \in [N]} \max\{0, \lambda_s^* - f(\omega^s, x)\}. \quad (15)$$

Note that scenarios such that $\lambda_s^* - f(\omega^s, x) < 0$ do not contribute to the objective function. By using the definition of $f(\omega^s, x)$ in (2), we rewrite

$$\begin{aligned} \lambda_s^* - f(\omega^s, x) &= \lambda_s^* - (c^s)^\top x - \min_y \{(d^s)^\top y^s : W^s y^s \geq h^s - T^s x\} \\ &= \max_y \{(\lambda_s^* - (c^s)^\top x - (d^s)^\top y^s) : W^s y^s \geq h^s - T^s x\}. \end{aligned}$$

To solve (15), for each $s \in [N]$ we introduce a binary variable π_s . Setting $\pi_s = 1$ selects the term of scenario s . Thus, problem (15) is equivalent to

$$\max_{x, y, \pi} \sum_{s \in [N]} (\lambda_s^* - (c^s)^\top x - (d^s)^\top y^s) \pi_s \quad (16a)$$

$$x \in X, \pi \in \{0, 1\}^N, \quad (16b)$$

$$\pi_s = 1 \Rightarrow T^s x + W^s y^s \geq h^s, \quad \forall s \in [N]. \quad (16c)$$

As in Section 3.2, bounded scenario-wise feasible regions permit a big- M /Fortet linearization of (16), introducing $(n + p)N$ auxiliary variables and $O((n + p)N)$ constraints. We instead use the following nonanticipativity-inspired reformulation, which introduces a copy q^s of x for each scenario, enforces $q^s = x$ when $\pi_s = 1$ and $q^s = 0$ otherwise, and avoids both big- M constraints and products involving y^s :

$$\max_{x, y, \pi, q} \sum_{s \in [N]} (\lambda_s^* \pi_s - (c^s)^\top q^s - (d^s)^\top y^s) \quad (17a)$$

$$x \in X, \pi \in \{0, 1\}^N, \quad (17b)$$

$$T^s q^s + W^s y^s \geq h^s \pi_s, \quad \forall s \in [N], \quad (17c)$$

$$A q^s \geq b \pi_s, \quad \forall s \in [N], \quad (17d)$$

$$0 \leq q^s \leq \mathbf{1} \pi_s, \quad \forall s \in [N], \quad (17e)$$

$$-\mathbf{1}(1 - \pi_s) \leq x - q^s \leq \mathbf{1}(1 - \pi_s), \quad \forall s \in [N]. \quad (17f)$$

Lemma 2. Formulation (17) solves the pricing problem (15).

Proof. First, let $(\tilde{x}, \{\tilde{y}^s\}_{s \in [N]}, \tilde{\pi})$ be a feasible solution to (16). We define $\bar{x} = \tilde{x}$, $\bar{\pi} = \tilde{\pi}$, $\bar{q}^s = \tilde{\pi}_s \tilde{x}$ for $s \in [N]$, and $\bar{y}^s = \tilde{y}^s \tilde{\pi}_s$ for $s \in [N]$. We argue that $(\bar{x}, \{\bar{y}^s\}_{s \in [N]}, \bar{\pi}, \{\bar{q}^s\}_{s \in [N]})$ is a feasible solution to (17). Constraints (17b) are trivially satisfied. For every $s \in [N]$ with $\bar{\pi}_s = 1$, (17d), (17e) and (17f) hold by construction, and (17c) is satisfied since $(\tilde{x}, \{\tilde{y}^s\}_{s \in [N]}, \tilde{\pi})$ is a feasible solution to (16). For every $s \in [N]$ with $\bar{\pi}_s = 0$, $\bar{q}^s = 0$ implies that constraints (17d) and (17e) are trivially satisfied. Moreover, (17f) are redundant. Since $\bar{y}^s = 0$, also constraints (17c) are satisfied. Moreover, $(\bar{x}, \{\bar{y}^s\}_{s \in [N]}, \bar{\pi}, \{\bar{q}^s\}_{s \in [N]})$ has the same objective value as $(\tilde{x}, \{\tilde{y}^s\}_{s \in [N]}, \tilde{\pi})$ has in (16). This demonstrates the optimal value of (17) is at most the optimal value of (16) (and how an optimal solution of (16) can be translated to a feasible solution of (17)).

Next, we argue that (17) has an optimal solution $(\bar{x}, \{\bar{y}^s\}_{s \in [N]}, \bar{\pi}, \{\bar{q}^s\}_{s \in [N]})$ that satisfies $\bar{y}^s = 0$ for all $s \in [N]$ with $\bar{\pi}_s = 0$. Consider $s \in [N]$ in which $\bar{\pi}_s = 0$. Then $\bar{q}^s = 0$ by (17e), and the problem of optimizing y^s (with $\bar{\pi}_s = 0$ fixed) reduces to $\max\{-(d^s)^\top y^s : W^s y^s \geq 0\}$. The subproblem dual feasibility assumption implies this problem is bounded, and since the feasible region is a cone, the optimal value must be 0 and hence we can assume $\bar{y}^s = 0$ since this is a feasible solution that attains the optimal value.

Now, starting from $(\bar{x}, \{\bar{y}^s\}_{s \in [N]}, \bar{\pi}, \{\bar{q}^s\}_{s \in [N]})$, we construct a feasible solution $(\tilde{x}, \{\tilde{y}^s\}_{s \in [N]}, \tilde{\pi})$ to (16) by setting $\tilde{x} = \bar{x}$ and $\tilde{\pi} = \bar{\pi}$. For $s \in [N]$ with $\bar{\pi}_s = 1$ we set $\tilde{y}^s = \bar{y}^s$, whereas for $s \in [N]$ with $\bar{\pi}_s = 0$ we set $\tilde{y}^s = \bar{y}^s = 0$; the indicator constraint (16c) is vacuous for these scenarios, so $(\tilde{x}, \{\tilde{y}^s\}_{s \in [N]}, \tilde{\pi})$ is feasible for (16) and the objective value is unchanged. For $s \in [N]$ with $\bar{\pi}_s = 1$, constraints (17f) give $\bar{q}^s = \bar{x}$, so (16c) follows from (17c) and the objective term of scenario s is the same in (16a) and (17a). Hence $(\tilde{x}, \{\tilde{y}^s\}_{s \in [N]}, \tilde{\pi})$ is feasible for (16) with the same objective value as $(\bar{x}, \{\bar{y}^s\}_{s \in [N]}, \bar{\pi}, \{\bar{q}^s\}_{s \in [N]})$ in (17). $\square$

The pricing problem (15) can be solved by feeding (17) directly into the solver (we call this method CCG-MIP). We also derive a specialized branch-and-bound algorithm for solving (15) and refer to the CCG method using this as CCG-BB.

A branch-and-bound algorithm for the pricing problem. We first rewrite the pricing problem (15) as

$$\Phi^* = \max_{x \in X} \Phi(x), \quad \text{where} \quad \Phi(x) := \sum_{s \in [N]} \max\{0, \lambda_s^* - f(\omega^s, x)\}, \quad (18)$$

We define a *node* as an ordered pair $u = (F_0, F_1)$ of disjoint subsets $F_0, F_1 \subseteq [p]$ of first-stage variables, interpreted as variables fixed to 0 and 1 respectively. The *region* of u is $X(u) := \{x \in X : x_j = 0 \ \forall j \in F_0, \ x_j = 1 \ \forall j \in F_1\}$. The *root* is $u_0 := (\emptyset, \emptyset)$, so that $X(u_0) = X$. The value of the node is $a(u) := \max\{\Phi(x) : x \in X(u)\}$. The bound attached to a node is obtained by relaxing the requirement that all scenarios share one first-stage decision. For a scenario $s \in [N]$ and a node $u = (F_0, F_1)$, we define

$$g_s(u) = \min (c_s)^\top x + (d_s)^\top y \quad (19a)$$

$$\text{s.t. } T^s x + W^s y \geq h^s \quad (19b)$$

$$x \in X(u) \quad (19c)$$

We set $g_s(u) := +\infty$ if (19) is infeasible, and write $g(u) := (g_s(u))_{s \in [N]}$. Evaluating a node corresponds to solving the N problems (19), which are independent for a given u and can be solved in parallel across scenarios. Let

$$\tilde{\Phi}(u) := \sum_{s \in [N]} \max\{0, \lambda_s^* - g_s(u)\}. \quad (20)$$

Lemma 3. For every node u , $\tilde{\Phi}(u)$ is an upper bound on $a(u)$: $\tilde{\Phi}(u) \geq a(u)$.

Proof. If $X(u) = \emptyset$ the claim is immediate. Otherwise, fix any $x \in X(u)$. For each $s \in [N]$, we have $g_s(u) \leq f(\omega^s, x)$ by definition. This implies that $\lambda_s^* - f(\omega^s, x) \leq \lambda_s^* - g_s(u)$. Taking max preserves the inequality and we get $\max\{0, \lambda_s^* - g_s(u)\} \geq \max\{0, \lambda_s^* - f(\omega^s, x)\}$. Summing over all s gives $\tilde{\Phi}(u) \geq \Phi(x)$. Maximizing over $x \in X(u)$ yields $\tilde{\Phi}(u) \geq a(u)$. $\square$

Lemma 3 shows we can use $\tilde{\Phi}(u)$ as a bound within a branch-and-bound algorithm. A node u is fathomed if $\tilde{\Phi}(u) \leq \hat{L}$, where $\hat{L}$ is the value of the best solution of (18) found so far (a lower bound on Φ^*). Let $A(u) = \{s \in [N] : g_s(u) < +\infty\}$ denote the *active* set of scenarios that are feasible at node u . Every node with $A(u) = \emptyset$ is fathomed, because $\tilde{\Phi}(u) = 0 \leq \hat{L}$. When a node u is not fathomed, let $x^s(u)$ denote an optimal first-stage solution of (19) for every $s \in A(u)$, and define

$$\bar{x}(u) := \frac{1}{|A(u)|} \sum_{s \in A(u)} x^s(u).$$

The next lemma states that an integral scenario average certifies the optimality of the node.

Lemma 4. *Let u be a node with $A(u) \neq \emptyset$, and suppose $\bar{x}(u) \in \{0, 1\}^p$. Then $x^s(u) = \bar{x}(u)$ for every $s \in A(u)$ and $\Phi(\bar{x}(u)) = \tilde{\Phi}(u)$.*

Proof. Since $x^s(u)$ is binary and $\bar{x}(u)$ is the average of binary vectors, $\bar{x}(u)$ is binary if and only if all the $x^s(u)$ coincide, hence $x^s(u) = \bar{x}(u)$ for every $s \in A(u)$. As $x^s(u)$ optimizes (19), for all $s \in A(u)$ we have $f(\omega^s, x^s(u)) = g_s(u)$. For all $s \notin A(u)$, we have $g_s(u) = \infty$ and $f(\omega^s, \bar{x}(u)) = \infty$ (since $\bar{x}(u) \in X(u)$), so the scenario contributes 0 to $\Phi(x)$ and $\tilde{\Phi}(u)$. Summing over all s gives us $\Phi(\bar{x}) = \tilde{\Phi}(u)$. $\square$

If $\bar{x}(u)$ is fractional, we branch on the *most-fractional* index j (closest to $1/2$) creating children $u^- := (F_0 \cup \{j\}, F_1)$ and $u^+ := (F_0, F_1 \cup \{j\})$. Every branching step fixes one previously free index, so the depth of the tree is at most p and the search is finite. The algorithm starts from u_0 with $\hat{L} = \Phi(x^*)$ for an arbitrary $x^* \in X$, processes the open node with largest upper bound, and terminates when no open nodes remain.

Two more devices are used in our branch-and-bound algorithm. First, a *rounding heuristic*: at a node with fractional $\bar{x}(u)$ we form x^R by rounding $\bar{x}(u)$ componentwise and evaluate the complete node $u^R := (\{j : \bar{x}_j(u) < 1/2\}, \{j : \bar{x}_j(u) \geq 1/2\})$. Second, node evaluations are *cached* on the key (F_0, F_1) : distinct branching orders reach the same fixings, and the rounded node u^R is frequently revisited, so caching $(g(u), (x^s(u))_s)$ avoids repeated scenario solves. The complete method is stated in Algorithm 1.

4 Computational Experiments

We present the results of a detailed computational study designed to evaluate the value of recourse, quality of the scenario based heuristic, and the performance of the compact, assignment and subset formulations and their associated solution methods.

All MIP and LP problems are created in Julia using JuMP as the modeling language and solved with Gurobi v11.0.3. Experiments are carried out on a heterogeneous cluster of servers. Unless otherwise indicated, we use ‘Gurobi work units’ (GWU) to evaluate computational effort of a method. GWU is a platform independent measure of computational effort used by Gurobi when solving a model which is calibrated so that one GWU is approximately one second of CPU time. For a method that requires solving many problems with Gurobi, we compute the total GWU for that method by summing the GWU over all problems. The implementation details of all methods are provided in Appendix B.

Test problems. We test our methods on single- and multi-dimensional knapsack and capacitated facility location problems. In the knapsack problems, item profits, weights, and capacities are random. We use two

Algorithm 1 Branch-and-bound for the assignment-formulation pricing problem (15)

Input: duals λ^* , first-stage set X , scenario data $\{(c^s, d^s, T^s, W^s, h^s)\}_{s \in [N]}$

Output: optimal solution x^* to (15)

```
1: choose any  $x^* \in X$  and set  $\hat{L} \leftarrow \Phi(x^*)$ 
2: evaluate the root: solve (19) for all  $s \in [N]$  at  $u_0 = (\emptyset, \emptyset)$ , obtaining  $(g(u_0), (x^s(u_0))_s)$  and  $\tilde{\Phi}(u_0)$ 
3:  $\mathcal{O} \leftarrow \{(u_0, \tilde{\Phi}(u_0))\}$  ▷ open nodes with their upper bounds
4: while  $\mathcal{O} \neq \emptyset$  do
5:   pop  $(u, \tilde{\Phi}_{\max})$  from  $\mathcal{O}$  with  $\tilde{\Phi}_{\max}$  maximum ▷ best-first search
6:   if  $\tilde{\Phi}_{\max} \leq \hat{L}$  then continue
7:   retrieve or compute  $(g(u), (x^s(u))_s)$  and form  $\tilde{\Phi}(u)$  by (20)
8:   if  $\tilde{\Phi}(u) \leq \hat{L}$  then continue
9:   compute  $A(u)$  and  $\bar{x}(u)$ 
10:  if  $\bar{x} \in \{0, 1\}^p$  then ▷ Lem. 4: node solved exactly
11:    if  $\tilde{\Phi}(u) > \hat{L}$  then  $(x^*, \hat{L}) \leftarrow (\bar{x}, \tilde{\Phi}(u))$ 
12:    continue
13:  end if
14:   $x^R \leftarrow \text{round}(\bar{x})$ ; evaluate the node  $u^R$  ▷ rounding heuristic
15:  if  $\tilde{\Phi}(u^R) > \hat{L}$  then  $(x^*, \hat{L}) \leftarrow (x^R, \tilde{\Phi}(u^R))$ 
16:  select most fractional branching variable  $j^*$ 
17:  push  $((F_0 \cup \{j^*\}, F_1), \tilde{\Phi}(u))$  and  $((F_0, F_1 \cup \{j^*\}), \tilde{\Phi}(u))$  onto  $\mathcal{O}$ 
18: end while
19: return  $x^*$ 
```

instance families: (i) randomly generated instances following [Malaguti et al. \(2022\)](#) and (ii) perturbed OR-LIB knapsack benchmarks. The Malaguti single-dimensional knapsack instances are classified as “small” and “large”, see Appendix C. The multi-dimensional knapsack instances from both (i) and (ii) have five knapsack constraints. We consider two types of knapsack problem. In the no recourse (NR) model, the selected solution must be feasible to the knapsack constraints. In the recourse model (R), violation of the knapsack constraints is allowed but penalized in the objective at a unit cost α through continuous recourse variables (the value of α is given in Appendix C). The Malaguti instances are only of type NR (following [Malaguti et al. \(2022\)](#)), whereas the OR-LIB instances contain both R and NR instances. For facility location, we use two base instances with 25 and 50 potential facilities and mixed Bernoulli–Gaussian demands. We vary the number of binary variables n (items or facilities), scenarios N , and parameter K . Appendix C provides the formulations, base instances, parameter ranges, and data-generation details.

4.1 Value of recourse

We investigate the value of adding recourse to the K -adaptable model by demonstrating the profit gain it enables, relative to the no-recourse model, on one single-knapsack (SK) and one multi-dimensional knapsack (MK) instance with $n = 10$ items and $K = 4$. For model NR, a candidate set is generated by optimizing the NR model over a training set of 100 scenarios. For model R, several candidate sets are generated by optimizing the R model over the same training scenarios as the NR model, but with varying values of the penalty α , yielding candidate sets with different trade-offs between profit and constraint violation. An independently generated set of 1000 test scenarios is used to estimate the average profit and constraint violation of each candidate set (the same test set is used for each candidate set). When evaluating an R candidate set under a test scenario, the solution in the candidate set that maximizes profit minus α times violation is selected, where the value of α equals that used to generate the candidate set. When evaluating an NR candidate set under a test scenario, the solution in the candidate set that minimizes violation is selected (with ties broken by profit). Each point in Figure 1 corresponds to a single candidate set, with the y-coordinate representing

the average profit of the candidate set and the x -coordinate representing the average constraint violation divided by the average capacity of the constraint (so, e.g., 0.05 indicates the constraint is violated by about 5% of its capacity on average). The NR candidate set does not have zero violation on the test set due to the mismatch between training and test scenarios caused by sampling. We find that allowing modest violations as in the R model can yield substantial profit gains. On the SK instance, an average relative violation of about 2.4% raises average profit by 10.8%; on the MK instance, an average relative violation of about 4.7% raises average profit by 41.1%. Moreover, as α increases, violations approach zero and R converges to NR. We found qualitatively similar results on several other instances.

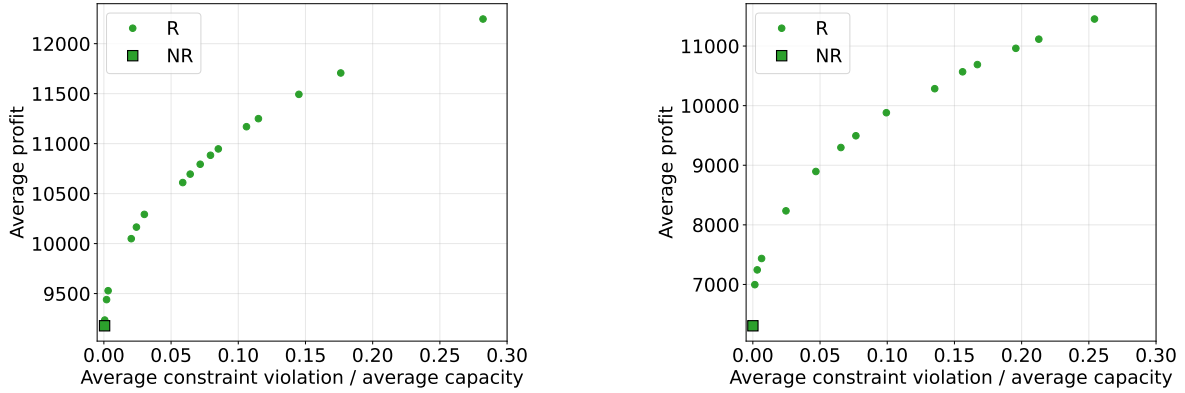


Figure 1: R (with recourse) vs. NR (no recourse) frontiers for $K = 4$ on the SK instance (left) and the MK instance (right): average profit vs. average relative constraint violation.

4.2 Scenario-based heuristic quality

We next investigate the quality of the candidate sets generated from the scenario-based heuristic described in Section 3.1. As a baseline, we compare against the (unattainable) wait-and-see value, which is computed by finding the optimal solution to each scenario separately. We fix a sample of N scenarios from the underlying distribution to obtain the SAA of problem $(K\text{-SR})$. The L scenarios used by the heuristic to generate the solution pool Q are sampled independently from the same underlying distribution. The N scenarios are used in the objective of the assignment formulation used to select a candidate set of K solutions from Q , and for comparison we also compute the optimal candidate set of K solutions (not restricted to Q) over this set of N scenarios.

Heuristic in recourse setting. We first present results for the K -adaptable model with recourse. Figure 2 plots the average gap between the solution obtained by the scenario-based heuristic and the wait-and-see bound as a function of K for single-knapsack instances with $N = 100$ scenarios, multi-knapsack instances with $N = 100$ scenarios, and facility location instances with $N = 25$ scenarios. Each panel shows one curve for each $L \in \{100, 200, 300\}$ and one curve for the optimal solution; the gap is averaged over instances with $n \in \{10, 20, 30, 40, 50\}$ for the knapsack problems, and over instances with $n \in \{25, 50\}$ for facility location. We find that in all these cases the scenario-based heuristic yields near-optimal solutions, with smaller gap to optimality as L grows. We also find that on these instances the average gap to the wait-and-see bound is small when $K = 10$.

Heuristic in no recourse setting. In the knapsack instances with no recourse the scenario-based heuristic is no longer guaranteed to yield a feasible solution: for every scenario, the candidate set must contain a solution that satisfies all the constraints without any recourse adjustment, so a pool of L single-scenario solutions may fail to admit a feasible K -adaptable assignment. For $L \in \{20, 50, 100\}$, the heuristic failed to find a feasible solution on 65, 34, and 30 of 162 Malaguti single-knapsack instances and 19, 6, and 1 of 96

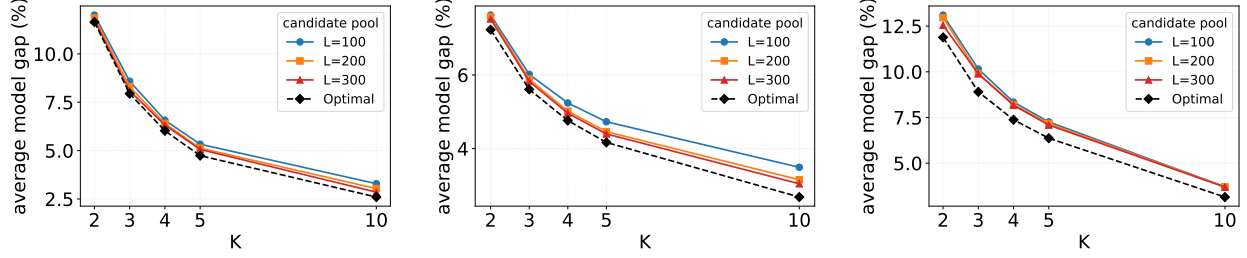


Figure 2: Average model gap (%) of the K -adaptable heuristic solution relative to the wait-and-see bound versus K , for single knapsack (left), multi-knapsack (center), and facility location (right).

multi-knapsack instances, respectively. For $L \in \{100, 200, 300\}$ on OR-LIB, the corresponding counts were 12, 8, and 8 of 60 single-knapsack instances and 49, 45, and 33 of 60 multi-knapsack instances. Thus, we find that the scenario-based heuristic is most effective on instances in which recourse variables are present that guarantee feasibility in every scenario.

K	Malaguti				OR-LIB								FL-R	
	SK-NR		MK-NR		SK-NR		MK-NR		SK-R		MK-R			
	(3)	(4)	(3)	(4)	(3)	(4)	(3)	(4)	(3)	(4)	(3)	(4)	(3)	(4)
2	0.0	0.0	2.4	1.7	1.4	0.1	4.1	0.2	8.4	0.6	19.4	0.2	43.2	3.4
3	1.0	0.6	5.7	5.7	2.5	1.6	8.5	3.5	2.1	1.7	21.3	1.4	52.1	6.4
4	1.9	0.6	8.2	6.3	7.3	2.6	15.0	5.7	2.7	2.5	25.7	2.2	54.7	6.6
5	2.7	0.8	8.2	5.7	10.4	2.8	19.8	7.9	4.0	2.7	21.0	2.8	53.5	6.1

Table 1: Average ending optimality gap (%) of the two compact formulations (3) and (4). NR and R denote the models without and with recourse. SK: Single Knapsack, MK: Multi-knapsack, FL: Facility Location.

4.3 Exact solution methods

We next investigate the performance of the exact solution methods, including the two compact formulations introduced in Section 2.1, the proposed CCG algorithm for solving the LP relaxation of the assignment formulation, and the column generation method for solving the subset formulation.

Compact Formulations. Table 1 presents the optimality gaps obtained when the work limits are reached for the two compact formulations: formulation (3), which extends the formulation proposed by Malaguti et al. (2022) to the recourse setting, and our new formulation (4). For the Malaguti-style instances, the reported values are averaged over $n \in \{10, 15\}$ and $N \in \{10, 15, 20\}$ for the single-knapsack problem (we only consider “small” instances here), and over $N \in \{10, 15, 20, 25\}$ for the multi-knapsack problem. For the OR-LIB instances, the averages are computed over $n \in \{10, 20, 30, 40, 50\}$ and $N \in \{25, 50, 100\}$ for both the single- and multi-knapsack problems. For the facility-location instances, the reported values are averaged over $n \in \{25, 50\}$ and $N \in \{25, 50\}$. The work limits are 600, 12000, and 30000 GWU for the Malaguti, OR-LIB, and facility-location instances, respectively. These results indicate that the new formulation (4) is much more promising for solving these instances. On the other hand, they also indicate that this formulation is limited in its ability to fully close the optimality gap.

Overall comparison. We next compare three exact approaches, the compact formulation (4), the assignment formulation (AF) solved by our CCG method, and the subset (SF) formulation solved with column generation. Work limits are 600 GWU for Malaguti small single-knapsack and multi-knapsack instances, 7200 GWU for Malaguti large single knapsack instances, 12000 GWU for OR-LIB instances, and 30000 GWU for facility

location instances. For both AF and SF, we restrict our attention to solving their LP relaxations, rather than embedding them in a full branch-and-price framework. We find that after solving their LP relaxations, solving the current restricted master problem with integrality enforced yields average integrality gaps of only 0.26%, 0.01%, and 0.02% for CCG-MIP on the Malaguti, OR-LIB, and facility-location instances, respectively, suggesting little benefit from branch-and-price.

We first empirically compare the two SF pricing formulations (11) and (12) on the Malaguti-style single-knapsack (SK) and multi-knapsack (MK) instances. For each instance, we run the column-generation algorithm with each pricing formulation and record the total Gurobi work units spent solving the pricing subproblem. Figure 3 plots these values against one another: for each instance, the horizontal coordinate is the pricing work units under formulation (11) and the vertical coordinate is the pricing work units under formulation (12). Points below the diagonal correspond to instances on which (12) consumes fewer work units than (11).

For the single-knapsack instances (Figure 3, left), the two formulations are broadly comparable: the points lie close to the diagonal, and neither formulation dominates the other across the instance set. For the multi-knapsack instances (Figure 3, right), however, formulation (12) is substantially more efficient, consuming fewer work units than (11) on the majority of instances.

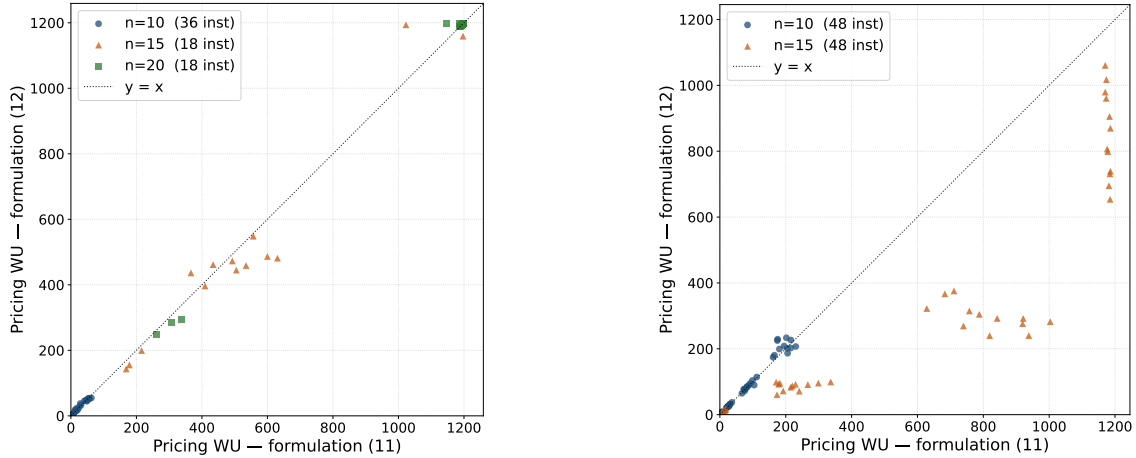


Figure 3: Comparison of the two SF pricing formulations (11) (x -axis) and (12) (y -axis), measured in Gurobi work units on the Malaguti-style single-knapsack (left) and multi-knapsack (right) instances.

Based on this comparison, we use formulation (12) for the SF pricing problem. Because AF pricing has an analogous structure, we likewise use formulation (17) rather than a mixed-integer linearization of (16) to solve the AF pricing. Returning to the comparison of the three exact approaches, we first consider the knapsack instances. Figure 4 displays performance profiles of these three methods on the two classes of these instances (Malaguti and OR-LIB). These plots display, for each method, the number of instances on which the GWU required by that method is within a factor of τ of the best method. We find that the assignment formulation dominates: AF CCG-MIP is the fastest method on 188 of the 258 Malaguti instances and 198 of the 240 OR-LIB instances. The compact formulation is competitive on a nontrivial share of the smaller instances, achieving the fastest solution on 38 Malaguti and 21 OR-LIB instances. The subset formulation is uniformly outperformed and is never the fastest method on any instance in either benchmark.

We next compare AF and SF in terms of ending optimality gaps for instances not solved within the work limit. AF solves substantially more instances to optimality: 86% versus 39% on Malaguti instances and 91% versus 38% on OR-LIB instances. Figure 5 displays a scatter plot of the ending optimality gaps of the two methods on the set of instances where both methods reach the work limit. We find that AF achieves

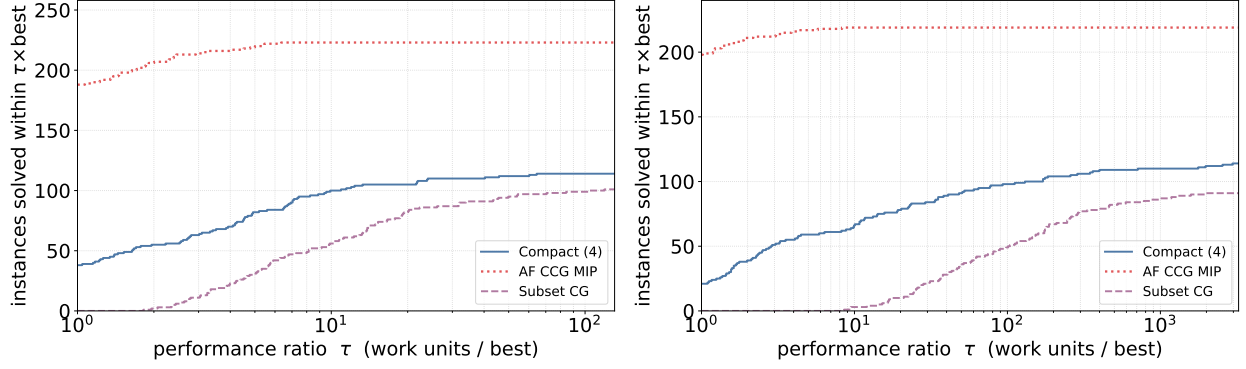


Figure 4: Performance profiles of the three exact methods: the compact formulation (4), AF CCG-MIP, and Subset CG, measured in total Gurobi work units, on the Malaguti (left, 258 NR instances) and OR-LIB (right, 120 NR and 120 R instances) benchmarks. For each algorithm, the curve plots the number of instances on which it solved the problem within a factor of τ of the fastest method.

significantly smaller optimality gap than SF on these instances.

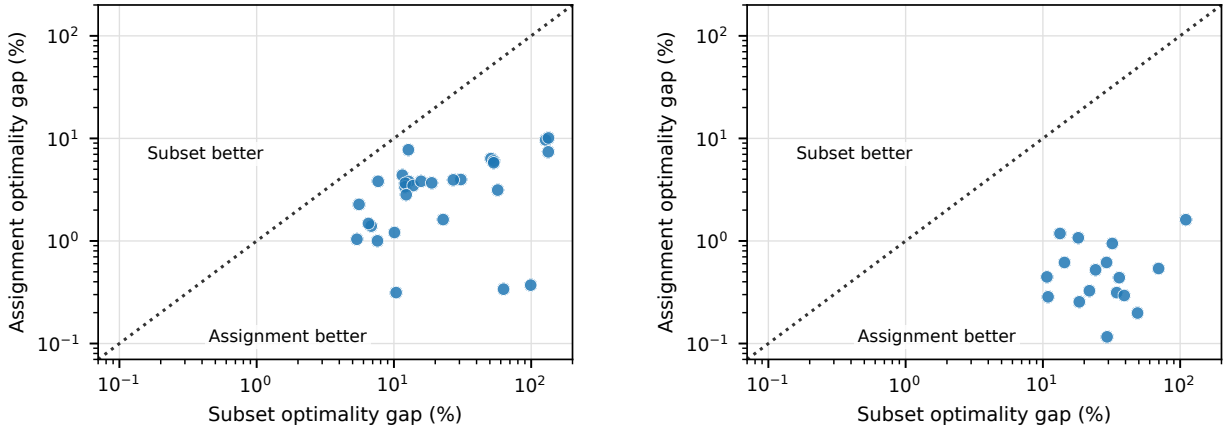


Figure 5: Optimality gaps of AF and SF on instances where both hit the work limit: Malaguti (left), OR-LIB (right).

Figure 6 displays box plots of the number of iterations of the AF and SF methods, where each box summarizes the distribution of the number of iterations required for instances having a fixed number of scenarios. Here, we include instances for which at least one method reached optimality. Whenever SF converges, AF also converges; therefore, all reported AF counts are terminating iteration counts, whereas SF counts are lower bounds on the true number of iterations SF would need. These results illustrate a key strength of the AF method: it requires considerably fewer iterations than SF, and the number of iterations is less sensitive to the number of scenarios.

We next compare the AF and SF methods on the facility location instances. Table 2 displays the number of instances solved (of the four tested with $K \in \{2, 3, 4, 5\}$), the average work units (the limit of 30000 units is used when an instance hits the limit), the average ending optimality gap, and the median number of iterations for these two methods, summarized by base instance (cap91 vs cap111) and number of scenarios. For $N = 25$, we find that AF CCG-MIP solves all cap91 and cap111 instances within the work limit, while SF solves only 3 of the 4 cap91 instances and none of the cap111 instances. Neither formulation solves any of the instances with $N = 50$. AF yields better ending optimality gaps than SF within the work limit, but the ending gap is higher than that obtained by the compact formulation (4) (Table 1). We again find that AF has

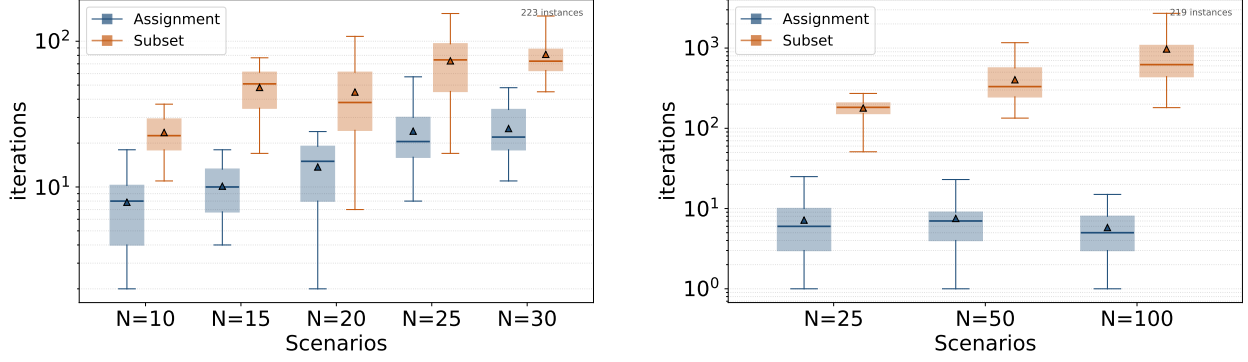


Figure 6: Iterations of the assignment (AF) vs. subset (SF) formulations on Malaguti (left) and OR-LIB (right) instances.

lower median iteration counts than SF.

Instance	Solved (of 4)				Work units				Optimality Gap (%)				Median iterations			
	$N = 25$		$N = 50$		$N = 25$		$N = 50$		$N = 25$		$N = 50$		$N = 25$		$N = 50$	
	AF	SF	AF	SF	AF	SF	AF	SF	AF	SF	AF	SF	AF	SF	AF	SF
cap91	4	3	0	0	10127	21461	WL	WL	0.0	2.9	14.4	21.4	7	85	5*	124*
cap111	4	0	0	0	7882	WL	WL	WL	0.0	13.9	9.6	23.3	12	104*	2*	122*

Table 2: AF vs. SF on the facility-location instances: number of instances solved to optimality, total Gurobi work units, average optimality gap, and median iterations at termination. * marks lower bounds for instances not solved to optimality.

CCG-MIP versus CCG-BB. The pricing problem (15) can be solved either by passing the MIP formulation (17) directly to Gurobi (CCG-MIP, Section 3.3) or by the customized branch-and-bound method (CCG-BB). Since CCG-BB incurs substantial Julia-side, API, and tree-management overhead not captured by Gurobi work units, we compare these two methods using wall-clock time, with limits of 1800 and 3600 seconds for SK and MK respectively. In this experiment we focus on the OR-LIB knapsack instances since these include the largest of our test instances. These experiments were run on an Apple M4 workstation with 16 GB RAM.

Figure 7 compares run times and optimality gaps at the time limit on the OR-LIB knapsack instances; the left panel compares the ratio of solution times of the two methods for varying problem size and number of scenarios, whereas the right panel shows a scatter plot of ending optimality gaps on instances that neither method solved within the time limit. Both methods solve all instances with $n \leq 30$, with the exception of four instances with $N = 100$ in the multi-knapsack problem class that CCG-MIP is unable to solve. CCG-BB is generally faster when $n \leq 30$, particularly on instances with more scenarios. On the other hand, CCG-MIP is more effective on instances with larger n and a modest number of scenarios: at $N = 25$ and $n = 40$ and 50 , it solves 16 instances for both cases, while CCG-BB solves 12 and 4 for $n = 40$ and $n = 50$, respectively. When both dimensions are large, both struggle; at $N = 100$ and $n = 40$ and 50 , CCG-MIP solves only 4 out of 16 and 2 out of 16 instances respectively, while CCG-BB solves none. Table 3 compares these methods on the facility location instances (the five instances in each row correspond to $K \in \{2, 3, 4, 5, 10\}$; these experiments were run on an Apple M4 workstation with a time limit of 8 hours). These results follow a similar pattern: CCG-BB solves all cap91 instances (with $n = 25$) at both scenario counts but no cap111 instances (with $n = 50$), whereas CCG-MIP solves all cap91 and cap111 instances at $N = 25$ but none at $N = 50$. Thus, CCG-BB is preferable for instances with a smaller number of binary variables, where it still solves all instances as the number of scenarios grows; CCG-MIP is preferable for instances with more binary variables, but its performance degrades on instances with a large number of scenarios.

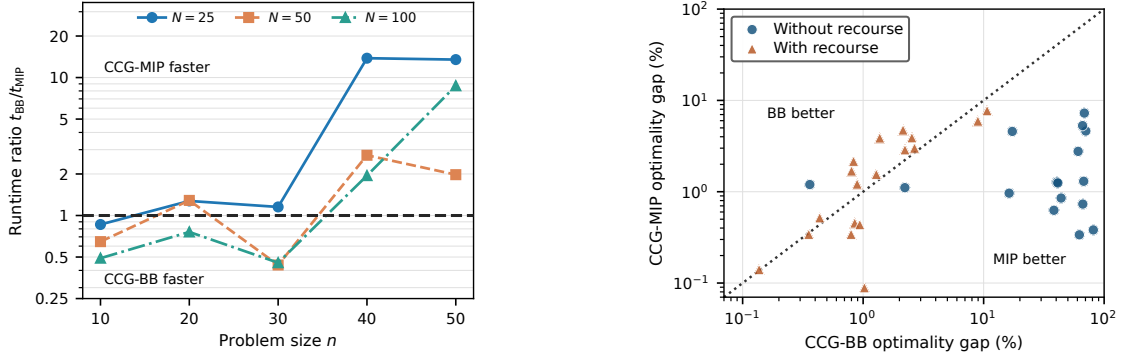


Figure 7: Comparison of CCG-MIP and CCG-BB on the OR-LIB instances. In the left panel, each point is the geometric mean of t_{BB}/t_{MIP} for a fixed (n, N) over instances solved to optimality by at least one method. The right panel reports gaps where both methods fail to solve to optimality.

Instance	Solved (of 5)				Time (s)				Optimality gap (%)			
	$N = 25$		$N = 50$		$N = 25$		$N = 50$		$N = 25$		$N = 50$	
	MIP	BB	MIP	BB	MIP	BB	MIP	BB	MIP	BB	MIP	BB
cap91	5	5	0	5	9785	3584	TL	17102	0.00	0.00	15.78	0.00
cap111	5	0	0	0	9985	TL	TL	TL	0.00	21.61	10.34	24.87

Table 3: CCG-MIP versus CCG-BB on the facility-location instances: number of instances solved to optimality, wall-clock time in seconds (TL: time limit reached), and average optimality gap (%) at termination.

5 Conclusion and future work

The paper introduces K -adaptability with continuous recourse. The idea is to select a (small) candidate set of the discrete part of the decisions in advance, allowing both the choice of one of the discrete vectors in the candidate set and free optimization of the continuous recourse variables for a given realization of the parameters. We introduce mathematical formulations of the problem and develop exact and heuristic approaches to solve it. An interesting direction for future work is to extend the model to include a set of decisions that must be made before any uncertainty is revealed as in the recent model of [De Santis et al. \(2025\)](#).

References

- Ahmed S (2013) A scenario decomposition algorithm for 0–1 stochastic programs. *Oper. Res. Lett.* 41(6):565–569.
- Arslan AN, Poss M, Silva M (2022) Min-sup-min robust combinatorial optimization with few recourse solutions. *INFORMS J. Comput.* 34(4):2212–2228.
- Beasley JE (1990) OR-Library: distributing test problems by electronic mail. *J. Oper. Res. Soc.* 41(11):1069–1072.
- Bertsimas D, Caramanis C (2010) Finite adaptability in multistage linear optimization. *IEEE Trans. Autom. Control* 55(12):2751–2766.
- Bertsimas D, Dunning I (2016) Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research* 64(4):980–998.
- Bertsimas D, Georghiou A (2015) Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research* 63(3):610–627.
- Buchheim C, Kurtz J (2017) Min–max–min robust combinatorial optimization. *Math. Program.* 163(1):1–23.

- Buchheim C, Kurtz J (2018) Complexity of min–max–min robustness for combinatorial optimization under discrete uncertainty. *Discrete Optimization* 28:1–15.
- Buchheim C, Prunte J (2019) K-adaptability in stochastic combinatorial optimization under objective uncertainty. *Eur. J. Oper. Res.* 277(3):953–963.
- De Santis M, Kurtz J, Donnini F (2025) K-adaptability for two-stage stochastic optimization. Technical report, Optimization Online, URL <https://optimization-online.org/2025/11/k-adaptability-for-two-stage-stochastic-optimization/>.
- Ghahtarani A, Saif A, Ghasemi A, Delage E (2023) A double-oracle, logic-based Benders decomposition approach to solve the K -adaptability problem. *Comput. Oper. Res.* 155:106243.
- Goerigk M, Kurtz J, Poss M (2020) Min–max–min robustness for combinatorial problems with discrete budgeted uncertainty. *Discrete Appl. Math.* 285:707–725.
- Han E, Bandi C, Nohadani O (2023) On finite adaptability in two-stage distributionally robust optimization. *Operations Research* 71(1):88–106.
- Hanasusanto GA, Kuhn D, Wiesemann W (2015) K-adaptability in two-stage robust binary programming. *Operations Research* 63(4):877–891.
- Homem-de Mello T, Valencia J, Lagos F, Lagos G (2024) Forecasting outside the box: Application-driven optimal pointwise forecasts for stochastic optimization. *arXiv preprint arXiv:2411.03520*.
- Julien E, Postek K, Birbil Ş (2025) Machine learning for K -adaptability in two-stage robust optimization. *INFORMS J. Comput.* 37(3):644–665.
- Kurtz J (2024) Approximation guarantees for min-max-min robust optimization and K -adaptability under objective uncertainty. *SIAM Journal on Optimization* 34(2):2121–2149.
- Kurtz J (2026) Bounding the optimal number of policies for robust K -adaptability. *Math. Program.* Forthcoming.
- Laporte G, Louveaux FV, van Hamme L (1994) Exact solution to a location problem with stochastic demands. *Transportation Science* 28(2):95–103.
- Malaguti E, Monaci M, Prunte J (2022) K-adaptability in stochastic optimization. *Math. Program.* 196(1):567–595.
- Malaguti E, Monaci M, Prunte J (2026) Heuristic algorithms for K-adaptability. *Optimization Letters* 20:1279–1297.
- Misra S, Roald L, Ng Y (2022) Learning for constrained optimization: Identifying optimal active constraint sets. *INFORMS J. Comput.* 34(1):463–480.
- Pillac V, Gendreau M, Gu  ret C, Medaglia AL (2013) A review of dynamic vehicle routing problems. *Eur. J. Oper. Res.* 225(1):1–11.
- Postek K, den Hertog D (2016) Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS J. Comput.* 28(3):553–574.
- Rockafellar RT (1970) *Convex Analysis* (Princeton, NJ: Princeton University Press).
- Shapiro A, Dentcheva D, Ruszczy  ski A (2014) *Lectures on stochastic programming: modeling and theory* (SIAM).
- Subramanyam A, Gounaris CE, Wiesemann W (2020) K-adaptability in two-stage mixed-integer robust optimization. *Math. Program. Comput.* 12(2):193–224.

Appendix A: Proof of Proposition 1: a counterexample under right-hand-side uncertainty

The idea of the construction is the following. In the recourse problem (2), the first-stage decision x enters only through the term $T(\omega)x$ subtracted from the right-hand side $h(\omega)$, i.e., x acts as a *shift* of the right-hand side. The instance is designed so that (i) on average over the realizations, no first-stage decision improves on $x = 0$, i.e., $\mathbb{E}_\omega[f(\omega, x)] > \mathbb{E}_\omega[f(\omega, 0)]$ for every $x \in X \setminus \{0\}$, whereas (ii) for every single realization $\omega \in \Omega$ some first-stage decision does, i.e., $\min_{x \in X} f(\omega, x) \leq f(\omega, 0) - \frac{1}{16}$. Property (i) makes $\bar{x} = 0$ the unique optimal solution of (K-SR) with $K = 1$; property (ii) makes it suboptimal for every realization. Throughout, given a vector $w \in \mathbb{R}^2$ we write $w = (w_1, w_2)$, and we denote by $H_+ := \{z \in \mathbb{R}^2 : z_2 \geq 0\}$ the closed upper half-plane.

A.1 The instance

Let

$$v_1 = (2, -1), \quad v_2 = (0, \tfrac{1}{2}), \quad v_3 = (-2, -1),$$

let $V \in \mathbb{R}^{3 \times 2}$ be the matrix whose rows are $v_1^\top, v_2^\top, v_3^\top$, and let

$$\psi(w) := \max_{i \in [3]} v_i^\top w, \quad w \in \mathbb{R}^2, \quad (21)$$

be the support function of the triangle $Q := \text{conv}\{v_1, v_2, v_3\}$; ψ is piecewise linear and convex. Further let

$$b^1 := \left(\tfrac{1}{4}, \tfrac{1}{8}\right), \quad b^2 := \left(-\tfrac{1}{4}, \tfrac{1}{8}\right), \quad B := (b^1 \ b^2) = \begin{pmatrix} 1/4 & -1/4 \\ 1/8 & 1/8 \end{pmatrix},$$

so that $Be_1 = b^1$ and $Be_2 = b^2$. Note that

$$B\{0, 1\}^2 = \{0, b^1, b^2, b^1 + b^2\} \subset H_+, \quad (Bx)_2 = \frac{x_1 + x_2}{8} > 0 \quad \text{for } x \in \{0, 1\}^2 \setminus \{0\}. \quad (22)$$

We consider the instance of (1) with $p = 2, n = 1, m = 3$, and data

$$c(\omega) = 0, \quad d(\omega) = 1, \quad W(\omega) = \mathbf{1} \in \mathbb{R}^3, \quad T(\omega) = VB, \quad h(\omega) = V\omega,$$

where $X = \{0, 1\}^2$ (i.e., the constraints $Ax \geq b$ are vacuous) and the random vector ω is uniformly distributed on the triangle

$$\Omega := \text{conv}\{\omega^1, \omega^2, \omega^3\}, \quad \omega^1 = (2, \tfrac{1}{2}), \quad \omega^2 = (0, 1), \quad \omega^3 = (-2, \tfrac{1}{2}),$$

shown in Figure 8a. Only the right-hand side is parametric, and the set of attainable right-hand sides is $h(\Omega) = V\Omega$, which is compact and convex, being the image of a compact convex set under a linear map. Note that $\omega_2 \in [1/2, 1]$ for every $\omega \in \Omega$.

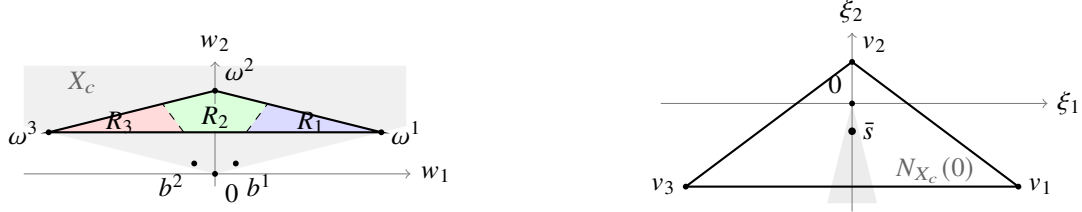
For $x \in X$ and a realization $\omega \in \Omega$, the recourse problem (2) has a single variable $y \in \mathbb{R}$ and reads

$$f(\omega, x) = \min \{y : \mathbf{1}y \geq V\omega - VBx\} = \max_{i \in [3]} v_i^\top (\omega - Bx) = \psi(\omega - Bx).$$

The recourse problem is feasible for every (ω, x) , and its dual feasible region is the unit simplex $\{\eta \in \mathbb{R}^3 : \mathbf{1}^\top \eta = 1, \eta \geq 0\}$, so both the scenario feasibility and the subproblem dual feasibility assumptions of Section 2 are satisfied. Defining the expected recourse cost of a shift $z \in \mathbb{R}^2$ as

$$\bar{\Psi}(z) := \mathbb{E}_\omega [\psi(\omega - z)], \quad z \in \mathbb{R}^2,$$

we have $\mathbb{E}_\omega[f(\omega, x)] = \bar{\Psi}(Bx)$ for every $x \in X$.



(a) The support Ω , its partition R_1, R_2, R_3 , the points b^1, b^2 , and the cone X_c of Corollary 1. (b) The triangle $Q = \text{conv}\{v_1, v_2, v_3\}$, the average subgradient $\bar{s}$, and the normal cone $N_{X_c}(0)$ of Remark 1.

Figure 8: Geometry of the counterexample of Proposition 1.

A.2 Average and pointwise behavior of the recourse cost

We first identify, for each realization, which of the three affine pieces of ψ is active. Within Ω , comparing the three linear forms shows that $v_2^\top \omega \geq v_1^\top \omega$ if and only if $\omega_1 \leq \frac{3}{4}\omega_2$, that $v_2^\top \omega \geq v_3^\top \omega$ if and only if $\omega_1 \geq -\frac{3}{4}\omega_2$, and that $v_1^\top \omega \geq v_3^\top \omega$ if and only if $\omega_1 \geq 0$. We accordingly partition Ω into

$$\begin{aligned} R_1 &:= \{\omega \in \Omega : \omega_1 > \frac{3}{4}\omega_2\}, \\ R_2 &:= \{\omega \in \Omega : |\omega_1| \leq \frac{3}{4}\omega_2\}, \\ R_3 &:= \{\omega \in \Omega : \omega_1 < -\frac{3}{4}\omega_2\}, \end{aligned}$$

so that $v_i^\top \omega = \psi(\omega)$ for all $\omega \in R_i, i \in [3]$ (Figure 8a), and we define $s : \Omega \rightarrow Q$ by $s(\omega) := v_i$ for $\omega \in R_i$. The vector $s(\omega)$ is a subgradient of ψ at ω : for every $w \in \mathbb{R}^2$,

$$\psi(w) \geq s(\omega)^\top w = s(\omega)^\top \omega + s(\omega)^\top (w - \omega) = \psi(\omega) + s(\omega)^\top (w - \omega). \quad (23)$$

We next compute the average subgradient $\bar{s} := \mathbb{E}_\omega[s(\omega)]$. The triangle Ω has area 1, so probabilities coincide with areas. The line $\omega_1 = \frac{3}{4}\omega_2$ intersects the bottom edge $\{\omega_2 = \frac{1}{2}\}$ of Ω at $(\frac{3}{8}, \frac{1}{2})$ and the edge joining ω^2 to ω^1 (contained in the line $\omega_1 = 4(1 - \omega_2)$) at $(\frac{12}{19}, \frac{16}{19})$. Hence R_1 is a triangle of area $\frac{1}{2} \cdot \frac{13}{8} \cdot \frac{13}{38} = \frac{169}{608}$, R_3 is its mirror image, and

$$\Pr(\omega \in R_1) = \Pr(\omega \in R_3) = \frac{169}{608}, \quad \Pr(\omega \in R_2) = 1 - \frac{338}{608} = \frac{135}{304}.$$

Since $\Pr(\omega \in R_1) = \Pr(\omega \in R_3)$ and the first coordinates of v_1 and v_3 are opposite while v_2 has first coordinate zero, the first coordinate of $\bar{s}$ vanishes, and

$$\bar{s} = \sum_{i \in [3]} \Pr(\omega \in R_i) v_i = \left(0, \frac{135}{304} \cdot \frac{1}{2} - \frac{338}{608}\right) = \left(0, -\frac{203}{608}\right).$$

Only the two facts $\bar{s}_1 = 0$ and $\bar{s}_2 < 0$ are used in the sequel; in particular, $\bar{s}$ lies in the interior of Q (Figure 8b).

The first lemma shows that, on average, shifting the right-hand side in any direction with positive second coordinate cannot help, so that the zero shift is optimal over the half-plane H_+ .

Lemma 5 (Average optimality on the half-plane). *For every $z \in \mathbb{R}^2$,*

$$\bar{\Psi}(z) \geq \bar{\Psi}(0) + \frac{203}{608} z_2.$$

In particular, 0 minimizes $\bar{\Psi}$ over H_+ , and $\bar{\Psi}(z) > \bar{\Psi}(0)$ for every $z \in H_+$ with $z_2 > 0$.

Proof. Fix $z \in \mathbb{R}^2$. Applying (23) with $w = \omega - z$ gives $\psi(\omega - z) \geq \psi(\omega) - s(\omega)^\top z$ for every $\omega \in \Omega$. Taking expectations with respect to ω yields $\bar{\Psi}(z) \geq \bar{\Psi}(0) - \bar{s}^\top z = \bar{\Psi}(0) + \frac{203}{608}z_2$, using $\bar{s} = (0, -\frac{203}{608})$. The two consequences are immediate. $\square$

The second lemma shows that, in contrast, for every single realization one of the two shifts b^1, b^2 strictly decreases ψ , by a margin that is uniform over the support.

Lemma 6 (Uniform pointwise improvement). *For every $\omega \in \Omega$,*

$$\min \{ \psi(\omega - b^1), \psi(\omega - b^2) \} \leq \psi(\omega) - \frac{1}{16}.$$

Proof. First let $\omega \in \Omega$ with $\omega_1 \geq 0$; we show $\psi(\omega - b^1) \leq \psi(\omega) - \frac{1}{16}$. We have

$$v_1^\top b^1 = \frac{3}{8}, \quad v_2^\top b^1 = \frac{1}{16}, \quad v_3^\top b^1 = -\frac{5}{8},$$

and therefore

$$\psi(\omega - b^1) = \max \left\{ v_1^\top \omega - \frac{3}{8}, v_2^\top \omega - \frac{1}{16}, v_3^\top \omega + \frac{5}{8} \right\}.$$

Since $v_i^\top \omega \leq \psi(\omega)$ for each i , the first two terms are at most $\psi(\omega) - \frac{3}{8}$ and $\psi(\omega) - \frac{1}{16}$, respectively. For the third term, using $\psi(\omega) \geq v_2^\top \omega$ together with $\omega_1 \geq 0$ and $\omega_2 \geq \frac{1}{2}$,

$$\psi(\omega) - v_3^\top \omega \geq v_2^\top \omega - v_3^\top \omega = \frac{3}{2}\omega_2 + 2\omega_1 \geq \frac{3}{4},$$

so that $v_3^\top \omega + \frac{5}{8} \leq \psi(\omega) - \frac{3}{4} + \frac{5}{8} = \psi(\omega) - \frac{1}{8}$. Combining the three bounds gives $\psi(\omega - b^1) \leq \psi(\omega) - \frac{1}{16}$.

Now let $\omega \in \Omega$ with $\omega_1 \leq 0$, and consider the reflection $\sigma(w_1, w_2) := (-w_1, w_2)$. Since σ swaps v_1 and v_3 and fixes v_2 , we have $\psi \circ \sigma = \psi$; moreover $\sigma(\Omega) = \Omega$ and $\sigma(b^1) = b^2$. Hence

$$\psi(\omega - b^2) = \psi(\sigma(\omega) - b^1) \leq \psi(\sigma(\omega)) - \frac{1}{16} = \psi(\omega) - \frac{1}{16},$$

where the inequality follows from the first part applied to $\sigma(\omega) \in \Omega$, which satisfies $\sigma(\omega)_1 \geq 0$. $\square$

The constant $\frac{1}{16}$ in Lemma 6 is attained at $\omega = \omega^2 = (0, 1)$, where $\psi(\omega^2) = \frac{1}{2}$ and $\psi(\omega^2 - b^1) = \psi(\omega^2 - b^2) = \frac{7}{16}$.

A.3 Proofs of Proposition 1 and Corollary 1

The two lemmas combine into a single statement about an arbitrary set of admissible shifts, from which both the binary and the convex case follow.

Lemma 7. *Let $Z \subseteq H_+$ be a set such that $Z \cap \{z \in \mathbb{R}^2 : z_2 = 0\} = \{0\}$ and $b^1, b^2 \in Z$. Then 0 is the unique minimizer of $\bar{\Psi}$ over Z , and yet*

$$\min_{z \in Z} \psi(\omega - z) \leq \psi(\omega) - \frac{1}{16} \quad \text{for every } \omega \in \Omega.$$

Proof. Every $z \in Z \setminus \{0\}$ satisfies $z_2 > 0$, so Lemma 5 gives $\bar{\Psi}(z) > \bar{\Psi}(0)$; this proves the first claim. For the second, fix $\omega \in \Omega$. Since $b^1, b^2 \in Z$, Lemma 6 gives $\min_{z \in Z} \psi(\omega - z) \leq \min\{\psi(\omega - b^1), \psi(\omega - b^2)\} \leq \psi(\omega) - \frac{1}{16}$. $\square$

Proof of Proposition 1. Consider the instance of Section A.1, for which $f(\omega, x) = \psi(\omega - Bx)$ and $\mathbb{E}_\omega[f(\omega, x)] = \bar{\Psi}(Bx)$. Let $Z := B\{0, 1\}^2 = \{0, b^1, b^2, b^1 + b^2\}$. By (22), $Z \subset H_+$, Z meets the line $\{z_2 = 0\}$ only at 0, and $b^1, b^2 \in Z$, so Lemma 7 applies. Since B is invertible, $x \mapsto Bx$ is a bijection between $X = \{0, 1\}^2$ and Z ; hence $\bar{x} = 0$ is the unique minimizer of $\mathbb{E}_\omega[f(\omega, x)]$ over X , i.e., the unique optimal solution of (K-SR) with $K = 1$, while for every realization $\omega \in \Omega$,

$$\min_{x \in X} f(\omega, x) = \min_{z \in Z} \psi(\omega - z) \leq \psi(\omega) - \frac{1}{16} = f(\omega, \bar{x}) - \frac{1}{16}.$$

In particular, $\bar{x} = 0$ is not an optimal solution of (1) for any realization $\omega \in \Omega$, i.e., $\bar{x} \notin X^*$. $\square$

The failure exhibited by Proposition 1 is resolved on average: each of the two alternatives $e_1, e_2 \in X$ improves the scenarios in one half of Ω but deteriorates those in the other half by a larger amount in expectation. For instance, $\psi(\omega^3 - b^1) = \psi(\omega^3) + \frac{5}{8}$.

Lemma 7 does not use the integrality of X , and this is the content of Corollary 1. Consider the variant of the instance in which the first-stage decision is the shift $z \in \mathbb{R}^2$ itself (so that $f(\omega, z) = \psi(\omega - z)$ and B is the identity), constrained to lie in the closed convex cone

$$X_c := \{z \in \mathbb{R}^2 : z_1 + 4z_2 \geq 0, -z_1 + 4z_2 \geq 0\} = \left\{z \in \mathbb{R}^2 : z_2 \geq \frac{|z_1|}{4}\right\},$$

shown in Figure 8a.

Corollary 1. *The point $z^* = 0$ is the unique minimizer of $\bar{\Psi}$ over X_c , yet for every $\omega \in \Omega$,*

$$\min_{z \in X_c} \psi(\omega - z) \leq \psi(\omega) - \frac{1}{16},$$

i.e., $z^* = 0$ does not minimize $\psi(\omega - z)$ over X_c for any realization $\omega \in \Omega$.

Proof. We verify the hypotheses of Lemma 7 for $Z = X_c$. Every $z \in X_c$ satisfies $z_2 \geq \frac{|z_1|}{4} \geq 0$, with $z_2 = 0$ only if $z = 0$; hence $X_c \subset H_+$ and X_c meets $\{z_2 = 0\}$ only at 0. Moreover $b^1, b^2 \in X_c$: indeed $b_1^1 + 4b_2^1 = \frac{1}{4} + \frac{1}{2} = \frac{3}{4} \geq 0$ and $-b_1^1 + 4b_2^1 = -\frac{1}{4} + \frac{1}{2} = \frac{1}{4} \geq 0$, and symmetrically for b^2 . $\square$

A.4 Remarks

Remark 1. Corollary 1 can also be read through first-order optimality conditions, which are available in the convex setting. Since ψ is the maximum of finitely many affine functions, its subdifferential is $\partial\psi(w) = \text{conv}\{v_i : v_i^\top w = \psi(w)\}$ (Rockafellar 1970, Theorem 23.8). The normal cone of X_c at 0 is $N_{X_c}(0) = \text{cone}\{(-1, -4), (1, -4)\} = \{\xi \in \mathbb{R}^2 : \xi_2 \leq -4|\xi_1|\}$ (Figure 8b), and $g := -\bar{s} \in \partial\bar{\Psi}(0)$ satisfies $-g = \bar{s} \in N_{X_c}(0)$, certifying that 0 minimizes $\bar{\Psi}$ over X_c . On the other hand, for a fixed ω , the point 0 minimizes the convex function $\psi(\omega - z)$ over X_c if and only if $\partial\psi(\omega) \cap N_{X_c}(0) \neq \emptyset$; within Ω , the possible subdifferentials of ψ are $\{v_1\}, \{v_2\}, \{v_3\}, \text{conv}\{v_1, v_2\}$, and $\text{conv}\{v_2, v_3\}$, none of which intersects $N_{X_c}(0)$. The obstruction is thus geometric: the average subgradient $\bar{s}$ lies in the interior of Q and in $N_{X_c}(0)$, while no subdifferential of ψ generated by a single realization in Ω reaches $N_{X_c}(0)$.

Remark 2. The result of Homem-de Mello et al. (2024) concerns continuous problems, so it is the convex variant of Corollary 1, rather than Proposition 1 itself, that should be compared with it. The instance of Corollary 1 is consistent with their result: taking the artificial scenario $\hat{\omega} = 0 \notin \Omega$, we have $\psi(\hat{\omega} - z) = \psi(-z) \geq 0 = \psi(0)$ for all $z \in \mathbb{R}^2$, because 0 lies in the interior of Q ; hence $z = 0$ minimizes $\psi(\hat{\omega} - z)$ over X_c . The corollary shows, however, that no scenario *within the support* has this property, and hence that in their theory the constructed scenario cannot, in general, be required to lie in the support of the random parameters—and, in particular, cannot be obtained by sampling from their distribution.

Appendix B: Implementation details of the exact methods

We compare the following four approaches:

- **Compact:** The compact formulations (3) and (4) are solved by Gurobi with default settings.
- **CCG-MIP:** The column-and-constraint generation algorithm applied to the assignment formulation, where each pricing subproblem is solved as a MIP by Gurobi. Pricing is executed in two phases. In the first phase, we impose a limit of 300 GWU on the pricing subproblem. If the limit is reached before optimality is proved, we check whether the incumbent constitutes an improving column; if so, we use Gurobi's solution pool feature to add up to the ten best improving solutions to the restricted master problem (RMP) and re-solve it. If no improving column is found within the first-phase budget, we remove the work limit and solve the pricing subproblem to optimality. The algorithm terminates when the pricing subproblem is solved to optimality and yields no improving column, thereby certifying optimality of the LP relaxation of the RMP. The RMP is initialized with the solution pool produced by the scenario-based heuristic of Section 3.1, i.e., the first-stage solutions of the single-scenario wait-and-see problems.
- **CCG-BB:** The column-and-constraint generation algorithm applied to the assignment formulation, using the same RMP initialization as CCG-MIP. The pricing subproblem is solved by the proposed customized branch-and-bound algorithm. At each node of the branch-and-bound tree, the per-scenario subproblems are independent given the branching decisions on the first-stage variables and are solved for each scenario.
- **Subset:** The column generation algorithm applied to the subset formulation. Pricing subproblems are solved as MIPs by Gurobi using the same two-phase pricing strategy as CCG-MIP. The RMP is initialized as follows: when solving the assignment formulation over Q in the scenario-based heuristic of Section 3.1, we keep the $r = 10$ best candidate sets instead of only the best one, and partition the scenario set according to which of the K solutions of each candidate set each scenario is assigned to. This yields $r \cdot K$ initial feasible subsets used to populate the RMP.

Appendix C: Test problems for computational study

We apply our models and test the computational performance of our solution methods on the knapsack problem (single and multi-dimensional) and capacitated facility location problems (Laporte et al. 1994).

C.1 Single dimensional and multi-dimensional knapsack problem

We are given a set of items I , a set of knapsacks M , and for each item $i \in I$ a profit p_i and a weight w_{im} in each knapsack $m \in M$. Each knapsack $m \in M$ has an associated capacity c_m . The goal is to select a subset of items that maximizes total profit without exceeding the capacity of any knapsack. We treat the profits p_i^s , weights w_{im}^s , and capacities c_m^s as depending on a scenario $s \in [N]$. We consider two versions of this model. In the recourse version, we allow some violation of the capacity constraints, which is penalized in the objective. For a fixed scenario s , the formulation is:

$$\max \quad \sum_{i \in I} p_i^s x_i - \alpha \sum_{m \in M} y_m^s \quad (24a)$$

$$\text{s.t.} \quad \sum_{i \in I} w_{im}^s x_i - y_m^s \leq c_m^s, \quad \forall m \in M, \quad (24b)$$

$$x_i \in \{0, 1\}, \quad \forall i \in I, \quad (24c)$$

$$y_m^s \geq 0, \quad \forall m \in M. \quad (24d)$$

Here, y_m^s is a shortfall variable that permits the capacity constraint in knapsack m to be violated under scenario s , at a penalty of α per unit of violation. The single-dimensional knapsack problem is recovered as the special case with $|M| = 1$, where a single capacity constraint applies across all items. The binary item selection variables are the variables for which we seek a candidate set of solutions of size at most K , and we allow the shortfall variables to be determined after a scenario is observed.

In the no-recourse version of the model, we require the knapsack constraints to be satisfied, without the option of violating them at a penalty. This corresponds to model (24) but with the y_m^s variables fixed to zero, and hence fits within the no-recourse model of Malaguti et al. (2022).

Data generation.

Malaguti-style instances. Our first set of instances follows the instance generation used in Malaguti et al. (2022). A scenario is generated as follows. For every item $i \in I$, we independently draw its profit $p_i(\omega) \sim \text{Unif}(0, 1)$. For every constraint $j \in [m]$ and item $i \in I$, we independently draw the corresponding weight $w_{ji}(\omega) \sim \text{Unif}(0, 1)$. The capacity of constraint j is then set to

$$c_j(\omega) = 0.75 \sum_{i \in I} w_{ji}(\omega). \quad (25)$$

This process is repeated independently N times to create an instance with N scenarios. In all three families below, scenarios are generated in this way, with $m = 1$ knapsack constraint for the two single-knapsack families and $m = 5$ for the multi-knapsack family. We consider three families: small single knapsack with $(n, N, K) \in \{10, 15\} \times \{10, 15, 20\} \times \{2, 3, 4, 5\}$, large single knapsack with $(n, N, K) \in \{15, 20, 25, 30, 50\} \times \{25, 30\} \times \{2, 5, 10\}$, and multi-knapsack with $(n, N, K) \in \{10, 15\} \times \{10, 15, 20, 25\} \times \{2, 3, 4, 5\}$. For each configuration (n, N, K) we generate three instances, giving 72 small single-knapsack, 90 large single-knapsack, and 96 multi-knapsack instances (258 in total).

OR-LIB instances. To create instances that do not have the item symmetry inherent in the instance generation approach of Malaguti et al. (2022) and also which exhibit some dependence in random variables, we also derive instances from the classical multi-knapsack instances of the OR-Library (Beasley 1990). Each OR-Library file is a collection of multiple, distinct benchmark instances. From mknap1.txt, we select two distinct benchmark instances: $n = 10$ and $n = 20$ items. Similarly, from mknap2.txt, we select three distinct benchmark instances WEISH01, WEISH06, and WEISH10, which have $n = 30$, $n = 40$, and $n = 50$ items, respectively. From each base instance we create two families of test instances. The multi-knapsack family retains the first $m = 5$ constraint rows of the base instance. The single-knapsack family is obtained by keeping only the first constraint row ($m = 1$).

Building on each base instance, we introduce a stochastic family by modeling the profits, weights, and capacities as random quantities perturbed around their deterministic values. We draw per-item standard deviations σ_i^p (profits) and σ_{ji}^w (weight of item i in constraint j), and per-constraint capacity scales σ_j^c . Each σ is uniformly generated between 0.2 and 0.3. These are drawn once and shared by all scenarios of the instance.

A scenario is generated as follows. We first draw standard Gaussian variates $\omega^{0,p} \sim \mathcal{N}(0, 1)$ and $\omega_j^{0,w}, \omega_j^{0,c} \sim \mathcal{N}(0, 1)$ for every constraint $j \in [m]$. The variate $\omega^{0,p}$ is a global profit perturbation shared by all items, while $\omega_j^{0,w}$ and $\omega_j^{0,c}$ are, respectively, the weight and capacity perturbations associated with constraint j . Then, independently for each item $i \in I$ and constraint $j \in [m]$, we draw standard Gaussian

variates $\omega_i^{1,P} \sim \mathcal{N}(0, 1)$ and $\omega_{ji}^{1,w} \sim \mathcal{N}(0, 1)$ and set

$$\begin{aligned} p_i(\omega) &= \max\{p_i(1 + \sigma_i^P(\omega_i^{1,P} + \omega^{0,P})), 0\}, & i \in I, \\ w_{ji}(\omega) &= \max\{w_{ji}(1 + \sigma_{ji}^w(\omega_{ji}^{1,w} + \omega_j^{0,w})), 0\}, & j \in [m], i \in I, \\ c_j(\omega) &= \max\{c_j(1 + \sigma_j^c \omega_j^{0,c}), 0\}, & j \in [m]. \end{aligned} \quad (26)$$

The single knapsack case follows the same construction with $m = 1$. Holding the sampled perturbation scales σ_i^P , σ_{ji}^w , and σ_j^c fixed across all scenarios, this scenario generation process is repeated N times with new independent Gaussian variates to create an instance with N scenarios. For both families, we use $N \in \{25, 50, 100\}$ in our tests. For the perturbed OR-Library family, the recourse penalty α in (24a) is set on a per-base-instance basis according to

$$\alpha = 2 \cdot \frac{\sum_{i \in I} p_i^{\text{base}}}{\sum_{m \in M} c_m^{\text{base}}}, \quad (27)$$

where p_i^{base} and c_m^{base} are the deterministic profits and capacities of the OR-Library base instance. We again have instances with $K \in \{2, 3, 4, 5\}$, and additionally $K = 10$ in the experiments of Section 4.2.

C.2 Capacitated facility location problem

Given a set of customers I and potential facilities J , the problem seeks to minimize total cost for opening facilities and for assigning customers to them, subject to capacity and demand satisfaction constraints. The deterministic formulation for a fixed scenario s is:

$$\min \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} c_{ij} y_{ij}^s + \sum_{i \in I} \alpha z_i^s \quad (28a)$$

$$\text{s.t.} \quad \sum_{j \in J} y_{ij}^s + z_i^s = d_i^s, \quad \forall i \in I, \quad (28b)$$

$$\sum_{i \in I} y_{ij}^s \leq u_j x_j, \quad \forall j \in J, \quad (28c)$$

$$x_j \in \{0, 1\}, \quad y_{ij}^s \geq 0, \quad z_i^s \geq 0, \quad \forall i \in I, j \in J. \quad (28d)$$

Here, x_j indicates whether facility j is opened, y_{ij}^s denotes the amount delivered from facility j to customer i , and z_i^s represents the shortage for customer i . The data include the fixed opening cost f_j , capacity u_j , unit delivery cost c_{ij} , and the demands d_i^s in scenario s . The penalty α discourages unserved demand.

We consider a parametric version of this problem with the demands d as the varying parameter (and hence describe the above model for a fixed demand scenario). We thus consider demands as the random variable ω . The facility opening variables x are the set of binary variables, for which we wish to select a candidate set of vectors of size at most K . We assume the variables (y^s, z^s) can be optimized after observing the nonnegative scenario d^s and so treat those as the continuous recourse variables.

Due to the presence of the shortfall variables z , model (28) is feasible for any choice of binary vector x , and hence the online problem of choosing (y, z) for a given candidate set is feasible for any scenario.

Data generation. We use the cap91 and cap111 instances from the ORLibrary (Beasley 1990) as our base instances. The cap91 base instance consists of 25 potential facility locations and 50 customers and cap111 consists of 50 potential facility locations and 50 customers. Building on the base instances, we introduce a stochastic family of instances by modeling the demands as drawn from a mixed Bernoulli–Gaussian distribution, which is described below.

A scenario is generated as follows. We first draw a standard deviation parameter $\sigma \sim \text{Unif}(0.1, 0.2)$ and a standard Gaussian variate $\omega^0 \sim \mathcal{N}(0, 1)$, both shared by all customers within the scenario; ω^0 represents a global perturbation common to every customer. Then, independently for each customer $i \in I$, we draw a standard Gaussian variate $\omega_i^1 \sim \mathcal{N}(0, 1)$ and a Bernoulli variate $\beta_i \sim \text{Bernoulli}(1/2)$, and set

$$d_i(\omega) = \beta_i \max\{0, d_i + \sigma d_i \omega_i^1 + \sigma d_i \omega^0\}. \quad (29)$$

The truncation keeps the perturbed demands nonnegative and the Bernoulli factor β_i removes each customer's demand independently with probability one half. This scenario generation process is repeated N times to create an instance with N scenarios. We set the unmet-demand penalty to $\alpha = 15$ for all facility location instances.

For each base instance, we create two randomly sampled instances, with $N = 25$ and $N = 50$ scenarios, and for each of these we use $K \in \{2, 3, 4, 5, 10\}$.