

ISE

Industrial and
Systems Engineering

An Inexact Column-And-Constraint Generation Method To Solve Two-Stage Robust Optimization Problems

MAN YIU TSANG¹, KARMEL S. SHEHADEH¹, AND FRANK E. CURTIS¹

¹Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA

ISE Technical Report 22T-008



An inexact column-and-constraint generation method to solve two-stage robust optimization problems

Man Yiu Tsang^a, Karmel S. Shehadeh^{a,*}, Frank E. Curtis^a

^a*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA*

Abstract

We propose a new inexact column-and-constraint generation (i-C&CG) method to solve two-stage robust optimization problems. The method allows solutions to the master problems to be inexact, which is desirable when solving large-scale and/or challenging problems. It is equipped with a backtracking routine that controls the trade-off between bound improvement and inexactness. Importantly, this routine allows us to derive theoretical finite convergence guarantees for our i-C&CG method. Numerical experiments demonstrate computational advantages of our i-C&CG method over state-of-the-art column-and-constraint generation methods.

Keywords: Two-stage stochastic optimization, robust optimization, column-and-constraint generation

1. Introduction

Robust optimization (RO) is a methodology for formulating optimization problems in which some parameters are uncertain, but belong to a given *uncertainty set*. In RO, one optimizes a system by hedging against the worst-case scenario of uncertain parameters within the predefined uncertainty set. For example, two-stage RO models are employed when some decisions are made before the uncertainty is revealed (i.e., first-stage problem) and some are made after the uncertainty is realized (i.e., second-stage problem). Two-stage RO models have received substantial attention in various application domains because of their ability to provide solutions that are robust to perturbations within the uncertainty set [1, 12, 21]. We refer readers to [7] for a recent survey.

Various solution methods have been proposed to obtain exact solutions to two-stage RO models under the master-subproblem framework. In this framework, a master problem and a subproblem are solved alternately. The master problem, as a relaxation of the RO model, provides a lower bound to the true optimal value, whereas the subproblem provides an upper bound. The algorithm terminates when the relative gap between the lower and upper bounds is less than a prescribed tolerance. One popular method is based on Benders' decomposition (BD), which constructs lower

*Corresponding author.

Email addresses: mat420@lehigh.edu (Man Yiu Tsang), kas720@lehigh.edu (Karmel S. Shehadeh), frank.e.curtis@lehigh.edu (Frank E. Curtis)

approximations of the objective function from a dual perspective, i.e., via second-stage dual variables [9, 16]. Another solution method is the column-and-constraint generation (C&CG) method proposed in [20]. Different from BD, C&CG constructs lower approximations of the objective function from a primal perspective, i.e., via second-stage variables and constraints. Results from [20] demonstrate the computational efficiency of C&CG over BD. Hence, C&CG has been employed widely to solve RO problems in many application domains [1, 6, 13, 23].

Recent research has been devoted to addressing the computational challenges arising from solving the subproblems within C&CG and to exploring relaxations of the assumptions adopted in [20]. Relaxing the relatively complete recourse assumption, Ayoub and Poss [2] derived an alternative mixed integer program (MIP) reformulation of the subproblem under a 0-1 polytope uncertainty set. In a similar line of research, Bertsimas and Shtern [3] developed a feasibility oracle and extended the convergence results in [20] from polyhedral to general compact uncertainty sets.

In various RO problems, the master problem is a large-scale (mixed) integer program. For this or other potential reasons, solving the master problem to optimality in each iteration can be challenging. However, discussions on computational issues associated with the master problem in the literature are sparse. The recent article [17] proposed an adaptive relative tolerance scheme when solving the master problem. Although computational results from [17] suggest an improvement in solution time by adopting their proposed scheme, the work did not provide theoretical guarantees concerning the accuracy and convergence of the proposed modification of C&CG.

In this paper, we propose a new inexact C&CG method for solving general two-stage RO problems. In our i-C&CG method, the master problems only need to be solved to a prescribed relative optimality gap or time limit, which may be the only tractable option when solving large-scale and/or challenging problems in practice. Our method involves a backtracking routine that controls the trade-off between bound improvement and inexactness. We derive theoretical guarantees and prove finite convergence of our approach. Numerical experiments on a scheduling problem demonstrate the computational advantages of our i-C&CG method over a state-of-the-art C&CG method.

The remainder of the paper is organized as follows. In Section 2, we present the general two-stage RO problem. In Section 3, we discuss the C&CG method, then, in Section 4, we introduce our proposed i-C&CG method and present its theoretical properties. Finally, we conduct computational experiments on an operating room scheduling problem in Section 5.

2. Two-stage RO Problem and Assumptions

We consider a two-stage stochastic problem using $\mathbf{x} \in \mathbb{R}^n$ to denote first-stage variables, \mathcal{X} to denote the first-stage feasible region, and $\mathbf{y} \in \mathbb{R}^m$ to denote second-stage variables. These variables can be either continuous or discrete. Let ξ be a random vector defined on the measurable space $(\mathbb{R}^l, \mathcal{B})$, where \mathcal{B} is the Borel σ -field. Our problem of interest is the two-stage robust linear optimization problem

$$v^* = \min_{\mathbf{x} \in \mathcal{X}} \left\{ \mathbf{c}^\top \mathbf{x} + \max_{\boldsymbol{\xi} \in \Xi} \min_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}, \boldsymbol{\xi})} \mathbf{q}^\top \mathbf{y} \right\}, \quad (1)$$

where $\Xi \subseteq \mathbb{R}^l$ is an uncertainty set and $\mathcal{Y}(\mathbf{x}, \boldsymbol{\xi}) = \{\mathbf{y} \in \mathbb{R}_+^m \mid \mathbf{T}\mathbf{x} + \mathbf{W}\mathbf{y} + \mathbf{C}\boldsymbol{\xi} \geq \mathbf{h}\}$ is the second-stage feasibility set. The parameters $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{q} \in \mathbb{R}^m$, $\mathbf{h} \in \mathbb{R}^r$, $\mathbf{T} \in \mathbb{R}^{r \times n}$, $\mathbf{W} \in \mathbb{R}^{r \times m}$, and $\mathbf{C} \in \mathbb{R}^{r \times l}$ are assumed to be known. For notational convenience, we use $Q(\mathbf{x}, \boldsymbol{\xi})$ to denote the value of the second-stage (recourse) problem for a given $(\mathbf{x}, \boldsymbol{\xi}) \in \mathcal{X} \times \Xi$, i.e., $Q(\mathbf{x}, \boldsymbol{\xi}) = \min_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}, \boldsymbol{\xi})} \mathbf{q}^\top \mathbf{y}$.

We make a few standard assumptions on problem (1). First, as in [20], we assume that the uncertainty set Ξ is either a finite set or a bounded polyhedron (i.e., a polytope). This is a mild assumption that is satisfied by various popular uncertainty sets (e.g., the budgeted uncertainty set [4]). Second, we assume that problem (1) has relatively complete recourse, i.e., for any $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$, the set $\mathcal{Y}(\mathbf{x}, \boldsymbol{\xi})$ is non-empty. Third, we assume problem (1) has an optimal solution with finite objective value, which holds trivially when \mathcal{X} is compact. Various applications, such as facility location problems and scheduling problems, fulfill the second and third assumptions [1, 12]. Finally, we assume that for all $\mathbf{x} \in \mathcal{X}$ and $\boldsymbol{\xi} \in \Xi$, the second-stage cost has $Q(\mathbf{x}, \boldsymbol{\xi}) \geq K$ for some $K > 0$. This is satisfied in many settings in which $Q(\mathbf{x}, \boldsymbol{\xi})$ represents actual implementation costs.

Remark 1. Although our discussions are based on the two-stage RO problem, we remark that several two-stage distributionally robust optimization problems can be reformulated in the form of (1); see Appendix A. Thus, our proposed i-C&CG method can solve such problems as well.

3. The Column-and-constraint Generation (C&CG) Method

Algorithm 1 summarizes the steps of the C&CG method from [20]. At each iteration j , a master problem (2) is solved using a subset of scenarios $\mathcal{S} \subseteq \Xi$ to obtain a solution \mathbf{x}^j . Since only a subset of the scenarios is considered, the master problem (2) is a relaxation of (1). Thus, its optimal value serves as a valid lower bound LB on the optimal value v^* of (1). Second, given the master problem solution \mathbf{x}^j , a scenario $\boldsymbol{\xi}^* \in \Xi$ is identified and an upper bound UB is computed from solving the subproblem (3). With the updated LB and UB , the current relative optimality gap is computed, i.e., $(UB - LB)/UB$. If this gap is less than the prescribed tolerance ε , then the algorithm terminates and returns the (nearly) optimal solution. Otherwise, the scenario set \mathcal{S} is enlarged by $\boldsymbol{\xi}^*$, the master problem is re-solved, and the method continues. Adding a new scenario in the master problem is equivalent to adding new second-stage variables and constraints.

A key computational challenge for the C&CG method is the need to solve the (enlarged) master and subproblems to optimality in each iteration. As mentioned earlier, most existing research focuses on solution methods and reformulations for the subproblem. In this paper, we assume that the maximin subproblem (3) can be solved by an optimality oracle (see, e.g., [3, 20]). Moreover, if the uncertainty set Ξ is a polytope, then the oracle outputs an extreme point of Ξ as the optimal solution $\boldsymbol{\xi}^*$ in step 2. Under the assumption that Ξ is either a finite set or a polytope, the C&CG method terminates in a finite number of iterations [3, 20].

Algorithm 1: Column-and-constraint generation (C&CG) method

Initialization: Set $LB = 0$, $UB = \infty$, $\varepsilon \in (0, 1)$, $\mathcal{S} = \emptyset$, $j = 1$.

1. Master problem. Solve the master problem:

$$\underset{\mathbf{x} \in \mathcal{X}, \delta}{\text{minimize}} \quad \mathbf{c}^\top \mathbf{x} + \delta \tag{2a}$$

$$\text{subject to} \quad \delta \geq Q(\mathbf{x}, \boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \in \mathcal{S}, \tag{2b}$$

$$\mathbf{c}^\top \mathbf{x} + \delta \geq 0. \tag{2c}$$

Record the optimal solution (\mathbf{x}^j, δ^j) and set $LB \leftarrow \mathbf{c}^\top \mathbf{x}^j + \delta^j$.

2. Subproblem. Solve the subproblem for fixed $\mathbf{x} = \mathbf{x}^j$:

$$\max_{\boldsymbol{\xi} \in \Xi} \min_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^j, \boldsymbol{\xi})} \mathbf{q}^\top \mathbf{y}. \tag{3}$$

Record the optimal solution $\boldsymbol{\xi}^*$ and value D^j .

Set $UB \leftarrow \min \{UB, \mathbf{c}^\top \mathbf{x}^j + D^j\}$.

3. Optimality test. If $(UB - LB)/UB < \varepsilon$, then terminate and return \mathbf{x}^j .

4. Scenario set enlargement.

Enlarge the scenario set $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{\xi}^*\}$.

Update $j \leftarrow j + 1$ and go back to step 1.

Note that the first-stage problem in many real-life applications involves a large number of integer-valued variables. For this and other potential reasons, solving the master problem (2) to optimality can be challenging. Moreover, once the method encounters a challenging master problem at some iteration, the subsequent master problems are potentially even more challenging to solve because of the enlarged scenario set. On the other hand, if the master problem is only solved to a prescribed (large) relative gap, then the value $\mathbf{c}^\top \mathbf{x}^j + \delta^j$ might not be a valid lower bound on v^* , so allowing inexactness is not something that can be done naively. These observations motivate our proposed i-C&CG method presented in the next section.

4. Inexact Column-and-constraint Generation (i-C&CG) Method

In this section, we present our i-C&CG method that aims to address computational difficulties encountered when solving challenging master problems in the C&CG method. In Section 4.1, we discuss the details of our i-C&CG approach. In Section 4.2, we derive theoretical properties of our i-C&CG approach and prove a finite convergence guarantee for it. Finally, in Section 4.3, we demonstrate the computational flexibility of our i-C&CG method and present some variants of it.

4.1. i-C&CG method

Algorithm 2 summarizes our proposed i-C&CG method, which shares a similar structure with the previously stated C&CG method except for a few additional simple updates in step 1 and the additional backtracking routine in step 3. In step 1.1, the algorithm solves the master problem

(4) to within a relative optimality gap tolerance of ε_{MP}^j and records the best feasible solution. Observe that as long as (1) has an optimal solution \mathbf{x}^* with finite objective value v^* and \bar{L} is finite, the master problem is guaranteed to be feasible. Indeed, one can easily verify that $(\mathbf{x}, \delta) = (\mathbf{x}^*, \max\{\bar{L}, v^*\} - \mathbf{c}^\top \mathbf{x}^*)$ is a feasible solution. See Remark 2 below for further discussion on the role played by \bar{L} in constraint (4c). In step 1.2, we obtain a lower bound L^j and upper bound U^j on the optimal value v_j^* from the solver, which are used in the backtracking routine to ensure convergence. If L^j is a valid lower bound, then ℓ is set as the current iteration index to indicate that it is the most recent iteration with such a valid bound (see Proposition 1). In step 1.3, the algorithm sets \bar{L} to U^j , which may accelerate the lower bound improvement for the next master problem; again, see Remark 2 below. Step 2 of the i-C&CG method is the same as that for the C&CG method, where the subproblem is solved to obtain a scenario $\boldsymbol{\xi}^*$ and compute a valid upper bound.

The backtracking routine in step 3 balances the computational gains and inaccuracies from solving the master problems inexactly. In particular, we shall show that this backtracking routine can adapt the inexactness tolerance in order to guarantee finite convergence of the method. The routine can be described as follows. First, if the *actual* relative gap $(\bar{U} - L^\ell)/\bar{U}$ is less than the prescribed tolerance ε , then the algorithm terminates and returns \mathbf{x}^j as the (nearly) optimal solution of the overall problem. Otherwise, the algorithm proceeds to an *exploitation* or *exploration* step based on the value of the *inexact* relative gap $(\bar{U} - U^j)/\bar{U}$. In an exploration step, the algorithm proceeds to step 4 and enlarges the current scenario set as in the C&CG method. In an exploitation step, the algorithm exploits knowledge of the current best valid lower bound L^ℓ and proceeds to step 1 by solving the master problem based on L^ℓ with a reduced relative gap tolerance ε_{MP}^j . This corrects any inaccuracies from solving prior master problems.

Remark 2. In step 1.3, one may set \bar{L} to L^j (instead of U^j) to ensure the lower bound validity and thus, convergence. In this case, since L^j is a valid lower bound on v^* , the constraint (4c) is always valid and the algorithm will always set ℓ (which tracks the most recent iteration that L^ℓ provides a valid lower bound) to the current iteration index j . On the other hand, setting \bar{L} to U^j (as stated in the algorithm) may accelerate the exploration process and help the solver escape from proving the optimality of a given feasible solution. We have found this to be computationally effective when some master problems in early iterations are challenging and the lower bounds that they provide may be very loose. In any case, the fact that the exploitation step sets $\bar{L} \leftarrow L^\ell$ ensures convergence as we will show in Section 4.2.

Remark 3. If $\varepsilon_{MP}^j = 0$ and $\tilde{\varepsilon} < \varepsilon$, then $L^j = U^j$ since the master problem is solved to optimality. Thus, in this case, \bar{L} is a valid lower bound and the algorithm sets $\ell \leftarrow j$. As a result, the backtracking routine reduces to checking the termination condition. Hence, our i-C&CG method is a generalization of the C&CG method with an additional valid lower bound constraint (4c).

Algorithm 2: Inexact column-and-constraint (i-C&CG) method

Initialization: $\bar{L} \leftarrow 0, \bar{U} \leftarrow \infty, \varepsilon \in (0, 1), \tilde{\varepsilon} \in (0, \varepsilon/(1 + \varepsilon)), \{\varepsilon_{MP}^j \in (0, 1)\}_{j \in \mathbb{N}}, \alpha \in (0, 1),$
 $\mathcal{S} \leftarrow \emptyset, j \leftarrow 1, \ell \leftarrow 0.$

1. Master problem.

1.1. Solve the master problem to within a relative optimality gap of ε_{MP}^j :

$$v_j^* = \underset{\mathbf{x} \in \mathcal{X}, \delta}{\text{minimize}} \quad \mathbf{c}^\top \mathbf{x} + \delta \quad (4a)$$

$$\text{subject to} \quad \delta \geq Q(\mathbf{x}, \boldsymbol{\xi}), \quad \forall \boldsymbol{\xi} \in \mathcal{S}, \quad (4b)$$

$$\mathbf{c}^\top \mathbf{x} + \delta \geq \bar{L}. \quad (4c)$$

Record the best feasible solution (\mathbf{x}^j, δ^j) found.

1.2. Record a lower bound $L^j \geq \bar{L}$ and upper bound $U^j = \mathbf{c}^\top \mathbf{x}^j + \delta^j$ of v_j^* .

If $L^j > \bar{L}$, then set $\ell \leftarrow j$.

1.3. Set $\bar{L} \leftarrow U^j$.

2. Subproblem. Solve the subproblem (3) for fixed $\mathbf{x} = \mathbf{x}^j$.

Record the optimal solution $\boldsymbol{\xi}^*$ and value D^j .

Set $\bar{U} \leftarrow \min \{\bar{U}, \mathbf{c}^\top \mathbf{x}^j + D^j\}$.

3. Optimality test and backtracking routine.

If $(\bar{U} - L^\ell)/\bar{U} < \varepsilon$, then terminate and return \mathbf{x}^j ; otherwise, do the following.

- **Exploitation:** If $(\bar{U} - U^j)/\bar{U} < \tilde{\varepsilon}$, then set $j \leftarrow \ell$ and $\bar{L} \leftarrow L^\ell$.
Set $\varepsilon_{MP}^j \leftarrow \alpha \varepsilon_{MP}^j$ for all $j \geq \ell$ and go back to step 1.
- **Exploration:** If $(\bar{U} - U^j)/\bar{U} \geq \tilde{\varepsilon}$, then go to step 4.

4. Scenario set enlargement.

Enlarge the scenario set $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{\xi}^*\}$.

Update $j \leftarrow j + 1$ and go back to step 1.

4.2. Theoretical Properties

In this section, we study the theoretical properties of the i-C&CG method. In particular, we derive an upper bound on the actual relative gap in the presence of inexactness (Proposition 2 and Corollary 3) and prove a finite convergence property (Theorem 4). First, in Proposition 1, we show the validity of L^ℓ as a lower bound on v^* (see Appendix B for a proof).

Proposition 1. *At any iteration, the value L^ℓ is a valid lower bound on the optimal value v^* to problem (1), i.e., $L^\ell \leq v^*$.*

Since \bar{U} is an upper bound on v^* , together with Proposition 1, these results justify the use of the *actual* relative gap $(\bar{U} - L^\ell)/\bar{U}$ as the termination condition. That is, if the algorithm terminates, then the actual relative gap computed based on a valid lower and a valid upper bound on v^* is less than the prescribed tolerance ε . Next, Proposition 2 provides an upper bound on the actual relative gap when the algorithm reaches the exploitation step.

Proposition 2. *If the i-C \mathcal{E} CG method reaches the exploitation step at iteration j , i.e., the actual relative gap satisfies $(\bar{U} - L^\ell)/\bar{U} \geq \varepsilon$, but the inexact relative gap satisfies $(\bar{U} - U^j)/\bar{U} < \tilde{\varepsilon}$, then the actual relative gap is at most $(1 - \tilde{\varepsilon})^{-1} \prod_{k=\ell}^j (1 - \varepsilon_{MP}^k)^{-1} - 1$.*

Proof. First, we consider the case that $j = \ell$. Since $(U^\ell - L^\ell)/U^\ell \leq \varepsilon_{MP}^\ell$ follows from the inexact solution of the master problem (4), it follows that $U^\ell \leq L^\ell/(1 - \varepsilon_{MP}^\ell)$ and thus,

$$\frac{U^\ell - L^\ell}{L^\ell} \leq \frac{1}{1 - \varepsilon_{MP}^\ell} - 1 =: \varepsilon' > 0. \quad (5)$$

Then, we have

$$\frac{U^\ell}{1 - \tilde{\varepsilon}} > \bar{U} \geq v^* \geq L^\ell \geq \frac{U^\ell}{1 + \varepsilon'}, \quad (6)$$

where the first inequality follows from $(\bar{U} - U^\ell)/\bar{U} < \tilde{\varepsilon}$, the second inequality follows since \bar{U} is an upper bound for v^* , the third inequality is a consequence of Proposition 1, and the last inequality follows from (5). Using the chain of inequalities (6), we obtain the desired inequality:

$$\frac{\bar{U} - L^\ell}{\bar{U}} < \frac{1}{\bar{U}} \left(\frac{U^\ell}{1 - \tilde{\varepsilon}} - \frac{U^\ell}{1 + \varepsilon'} \right) \leq \frac{1 + \varepsilon'}{1 - \tilde{\varepsilon}} - 1, \quad (7)$$

where the first inequality follows from the facts that (6) shows $\bar{U} < U^\ell/(1 - \tilde{\varepsilon})$ and $L^\ell \geq U^\ell/(1 + \varepsilon')$, and the second inequality follows from the fact that (6) shows $U^\ell/\bar{U} \leq 1 + \varepsilon'$. Hence, from (7), the desired conclusion follows in the case that $j = \ell$.

Next, we consider the case that $j > \ell$. Note that, for all $k \in \{\ell, \dots, j - 1\}$ in step 1, we have $(U^{k+1} - L^{k+1})/U^{k+1} \leq \varepsilon_{MP}^{k+1}$ from solving the master problem to within a relative gap of ε_{MP}^{k+1} . Moreover, step 1.2 implies $L^{k+1} = U^k$ for all $k \in \{\ell, \dots, j - 1\}$. Therefore,

$$U^{k+1} \leq \frac{L^{k+1}}{1 - \varepsilon_{MP}^{k+1}} = \frac{U^k}{1 - \varepsilon_{MP}^{k+1}} \quad (8)$$

for all $k \in \{\ell, \dots, j - 1\}$. Applying inequality (8), we obtain

$$U^j \leq U^{j-1} \cdot \frac{1}{1 - \varepsilon_{MP}^j} \leq \dots \leq U^\ell \cdot \prod_{k=\ell+1}^j \frac{1}{1 - \varepsilon_{MP}^k} \leq L^\ell \cdot \prod_{k=\ell}^j \frac{1}{1 - \varepsilon_{MP}^k},$$

where the last inequality follows from $(U^\ell - L^\ell)/U^\ell \leq \varepsilon_{MP}^\ell$. This in turn implies

$$\frac{U^j - L^\ell}{L^\ell} \leq \prod_{k=\ell}^j \frac{1}{1 - \varepsilon_{MP}^k} - 1 =: \varepsilon'' > 0. \quad (9)$$

Therefore, the chain of inequalities similar to (6) is as follows:

$$\frac{U^j}{1 - \tilde{\varepsilon}} > \bar{U} \geq v^* \geq L^\ell \geq \frac{U^j}{1 + \varepsilon''}, \quad (10)$$

where the last inequality follows from (9). Hence, we can derive

$$\frac{\bar{U} - L^\ell}{\bar{U}} < \frac{1}{\bar{U}} \left(\frac{U^j}{1 - \tilde{\varepsilon}} - \frac{U^j}{1 + \varepsilon''} \right) \leq \frac{1 + \varepsilon''}{1 - \tilde{\varepsilon}} - 1, \quad (11)$$

where the first inequality follows from the fact that (10) shows $\bar{U} < U^j/(1 - \tilde{\varepsilon})$ and $L^\ell > U^j/(1 + \varepsilon'')$,

and the second inequality follows from the fact that (10) shows $U^j/\bar{U} \leq 1 + \varepsilon''$. Hence, from (11), the desired conclusion also follows in the case that $j > \ell$. \square

Proposition 2 quantifies the effect of inexact solves of master problems on the actual relative gap. In Corollary 3, we provide a bound on the actual relative gap if the algorithm updates \bar{L} to L^j in step 1.3, which is equivalent to the C&CG method with inexact solves of master problems only [17] (see Appendix C for a proof).

Corollary 3. *Assume that in step 1.3, \bar{L} is updated as L^j instead of U^j . At iteration j , if the actual relative gap satisfies $(\bar{U} - L^j)/\bar{U} \geq \varepsilon$, but the inexact relative gap satisfies $(\bar{U} - U^j)/\bar{U} < \tilde{\varepsilon}$ in step 3, then the actual relative gap is at most $(1 - \tilde{\varepsilon})^{-1}(1 - \varepsilon_{MP}^j)^{-1} - 1$.*

Recall that in the backtracking routine (step 3), if the termination condition is not satisfied, then the algorithm proceeds to either the exploration or the exploitation step. If the inexact relative gap is large (i.e., greater than $\tilde{\varepsilon}$), then the algorithm explores possible new valid lower bounds on v^* . Otherwise, only a relatively small improvement in the lower bound could be achieved by the exploration step, and our i-C&CG method switches to exploiting the best current lower bound. Proposition 2 shows that the actual relative gap is bounded by $\tilde{\varepsilon}$ (the backtracking routine parameter) and ε_{MP}^k (the relative gap from solving master problems). Therefore, to close the actual relative gap, the algorithm reduces ε_{MP}^k in every exploitation step. Finally, we leverage the results in Proposition 2 to show the finite convergence of our proposed i-C&CG method in Theorem 4.

Theorem 4. *If $\tilde{\varepsilon} < \varepsilon/(1 + \varepsilon)$, then Algorithm 2 terminates in a finite number of iterations.*

Proof. We first show that every visit to step 4 (via the exploration step) enlarges the scenario set \mathcal{S} . Equivalently, we want to prove that if ξ^* in step 2 belongs to the current scenario set \mathcal{S} (i.e., $\xi^* \in \mathcal{S}$), then we will not proceed to the exploration step. Consider master problem (4) at some iteration j with current scenario set \mathcal{S} . Assume that $\xi^* \in \arg \max_{\xi \in \Xi} Q(\mathbf{x}^j, \xi)$ belongs to \mathcal{S} . Then, by the definition of U^j as an upper bound of v_j^* , we have

$$\mathbf{c}^\top \mathbf{x}^j + \max_{\xi \in \Xi} Q(\mathbf{x}^j, \xi) = \mathbf{c}^\top \mathbf{x}^j + \max_{\xi \in \mathcal{S}} Q(\mathbf{x}^j, \xi) \leq U^j,$$

where the first equation follows from $\xi^* \in \mathcal{S}$. This implies that \bar{U} updated in step 2 satisfies

$$\bar{U} - U^j \leq \mathbf{c}^\top \mathbf{x}^j + \max_{\xi \in \Xi} Q(\mathbf{x}^j, \xi) - U^j \leq 0.$$

Thus, the algorithm will not proceed to the exploration step.

Next, we show that Algorithm 2 terminates in a finite number of iterations. Note that in the i-C&CG method, if the termination criterion is not met, then the algorithm visits either the exploitation or the exploration step. Since we proved that every exploration step enlarges the scenario set, under the assumption that Ξ is a finite set or a polytope (with a finite number of extreme points), the number of exploration steps is finite. Hence, to verify the finite convergence

property, it suffices to show that for any fixed scenario set $\tilde{\mathcal{S}} \subset \Xi$, the master problem (4) is solved at most finitely many times (with possibly different \bar{L} values). Suppose, on the contrary, that master problem (4) with scenario set $\tilde{\mathcal{S}}$ is solved infinitely many times. This can happen only when neither the termination nor the exploration step is visited. That is, starting from the first re-visit of the master problem with scenario set $\tilde{\mathcal{S}}$ via an exploitation step, Algorithm 2 proceeds with the exploitation step forever. However, from Proposition 2, the actual relative gap is bounded by $1/[(1 - \tilde{\varepsilon})(1 - \varepsilon_{MP}^{\ell})] - 1$. Since the value of ε_{MP}^{ℓ} is reduced in each exploitation step, it follows that ε_{MP}^{ℓ} converges to zero as the i-C&CG method continues with the exploitation step. Therefore, the actual relative gap $1/[(1 - \tilde{\varepsilon})(1 - \varepsilon_{MP}^{\ell})] - 1$ converges to $\tilde{\varepsilon}/(1 - \tilde{\varepsilon})$, which is less than ε by our assumption. This implies that after a sufficiently large number of iterations, the termination condition will be satisfied, which contradicts that master problem (4) with scenario set $\tilde{\mathcal{S}}$ is solved infinitely many times. \square

Remark 4. We provide the following guidelines for choosing the i-C&CG method parameters, namely, $\tilde{\varepsilon}$, $\{\varepsilon_{MP}^j\}$, and α . First, recall that $\tilde{\varepsilon}$ determines whether exploitation or exploration is performed. As suggested by Theorem 4, one can set $\tilde{\varepsilon} < \varepsilon(1 + \varepsilon)$ and choose $\tilde{\varepsilon}$ close to this upper bound to favor exploitation. Second, recall that $\{\varepsilon_{MP}^j\}$ and α control the extent of the inexactness allowed by the method. If ε_{MP}^j is large (e.g., greater than ε), then a value of α that shrinks ε_{MP}^j at a relatively fast rate is preferred. In contrast, if ε_{MP}^j is comparable with ε , then a value of α close to 1 that shrinks ε_{MP}^j more slowly is preferred. Finally, we emphasize that there is no one set of parameters that yields the best performance for all problems, and indeed, such a flexibility allows the i-C&CG method to adapt to problems of different structures.

4.3. Variants of i-C&CG

Algorithm 2 provides a general framework to handle challenging master problems. That is, our proposed i-C&CG is flexible, allowing users to customize the algorithm for specific problems to achieve better computational performance. In this section, we discuss two variants of our i-C&CG method that provide additional flexibility for practical use (see Appendix D).

The first variant provides additional controls on the trade-off in step 3 by allowing users to impose an exploitation frequency f^{exploit} . Specifically, one can enforce the algorithm to proceed to the exploitation step when ℓ does not change for f^{exploit} iterations. This mechanism remedies the situation that the valid lower bound information is not exploited for a long time, i.e., $\ell \ll j$.

The second variant allows users to impose a time limit τ for solving the master problem. That is, one can run a solver for the master problem that terminates either when the relative gap of ε_{MP}^j is reached or when the solution time exceeds τ . Due to this additional source of inexactness, in the exploitation step, the algorithm can increase the time limit by a factor of $\beta > 0$. As a result, the algorithm establishes an adaptive time limit for solving the master problems that increases with each exploitation step. This variant could be useful to accelerate the lower bound improvement if

some intermediate master problems are challenging.

5. Numerical Results

In this section, we use a two-stage distributionally robust operating room (OR) scheduling problem recently studied in [18] to compare the performance of the C&CG and i-C&CG methods.

5.1. A distributionally robust operating room scheduling problem (DRORSP)

We start by introducing the DRORSP setting as in [18]. Let I be a set of surgeries to schedule and R be a set of ORs. Each surgery $i \in I$ has a random duration d_i where the support of $\mathbf{d} = (d_1, \dots, d_{|I|})^\top$ is $\Xi = \{\mathbf{d} \in \mathbb{R}^{|I|} \mid \underline{d}_i \leq d_i \leq \bar{d}_i, i \in I\}$. The fixed cost of opening an OR is c^f , and a per-unit overtime cost c^v is incurred if an OR operates beyond the working hour T . In the DRORSP, given the sets of surgeries I and ORs R , the OR manager aims to make the following decisions simultaneously: (a) decide which OR(s) to open, and (b) assign each surgery to an open OR. The objective is to minimize the fixed cost of opening ORs plus the worst-case expected cost associated with OR overtime. As in [18], we define the following ambiguity set

$$\mathcal{P} = \{\mathbb{Q} \in \mathcal{D}(\Xi) \mid \mathbb{E}_{\mathbb{Q}}(d_i) = \mu_i, \mathbb{E}_{\mathbb{Q}}|d_i - \mu_i| \leq \nu_i, i \in I\}, \quad (12)$$

where $\mathcal{D}(\Xi)$ is the set of probability measures with support Ξ . Ambiguity set (12) consists of all probability measures with support Ξ such that, for all $i \in I$, the mean is μ_i and mean absolute deviation (MAD) is less than ν_i .

For each $r \in R$, we define a binary variable x_r that equals 1 if OR r is open, and is 0 otherwise. In addition, we define a binary decision variable $y_{i,r}$ that equals 1 if surgery i is assigned to OR r , and is 0 otherwise. For any $a \in \mathbb{R}$, we write $(a)^+ = \max\{a, 0\}$. The DRORSP can be stated as:

$$\begin{aligned} & \underset{\mathbf{x} \in \{0,1\}^{|R|}, \mathbf{y} \in \{0,1\}^{|I| \times |R|}}{\text{minimize}} && \sum_{r \in R} c^f x_r + c^v \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[\sum_{r \in R} \left(\sum_{i \in I} y_{ir} d_i - T \right)^+ \right] && (13a) \\ & \text{subject to} && y_{ir} \leq x_r, \quad \forall i \in I, r \in R, && (13b) \\ & && \sum_{r \in R} y_{ir} = 1, \quad \forall i \in I. && (13c) \end{aligned}$$

Objective (13a) is a sum of the fixed cost of opening ORs and the worst-case expected overtime cost. Constraint (13b) ensures that surgeries are assigned to open ORs only, and constraints (13c) require that every surgeries is assigned to exactly one OR.

In [18], the authors demonstrated the challenges of solving problem (13) exactly and developed linear decision rules to approximate solutions to problem instances. For our experiments, we implemented the C&CG and i-C&CG methods to solve various instances of problem (13). In [Appendix E](#), we present the detailed derivations of an equivalent reformulation of problem (13) and the associated master problem and subproblem, as well as standard symmetry-breaking constraints included in the master problem to break symmetry in the solution space of first-stage decisions.

5.2. Test Instances and Experimental Setup

We use three years of surgery duration data for six different surgery types from [11] and parameter settings from the literature to generate various DRORSP instances as follows. First, we sample 500 data points to estimate the mean, MAD, and lower and upper bounds of the surgery durations for each surgery type. In particular, we set the lower and upper bounds as the (20th, 80th) or (10th, 90th) percentiles of the empirical distribution. Second, for each combination of $|I| \in \{20, 21, \dots, 25\}$ and $|R| \in \{7, 10\}$, we generate the number of surgeries for each type from a multinomial distribution with probability being equal to the estimated surgery type proportion from the data set. Third, we set $T = 480$ minutes and consider two sets of weights for the multi-criteria objective function $(c^f, c^v) \in \{(1, 1/30), (1, 1/120)\}$ as in [5]. Finally, for each combination of $|I|$, $|R|$, percentiles, and (c^f, c^v) , we generate and solve 5 instances for a total of 240 instances.

For algorithmic parameters, we set the final relative gap $\varepsilon \in \{2\%, 5\%\}$ for both the C&CG and i-C&CG methods. In the i-C&CG method, we set the initial master relative gap tolerance ε_{MP} to 2% and the inexact relative gap $\tilde{\varepsilon}$ to 1.5%. In addition, we impose an initial solver time limit of 300 seconds (s) for solving the master problems. If the time limit is exceeded, then we set U^j as the best objective value found. At each exploitation step, ε_{MP}^j is decreased by a factor of $\alpha = 0.8$ and the time limit is increased by 600 seconds. We implemented both the C&CG and i-C&CG methods with the AMPL modeling language and use CPLEX (version 20.1.0.0) as the solver with its default settings. We conducted all the experiments on a computer with an Intel Xeon Silver processor with a 2.10 GHz CPU and 128 Gb memory.

5.3. Experimental Results

We focus our discussion on the cost structure $(c^f, c^v) = (1, 1/30)$; the results for $(c^f, c^v) = (1, 1/120)$ are similar (see Appendix F). Figure 1 shows the time performance profiles for the C&CG and i-C&CG methods under two different lower and upper bound estimates. The curves represent the percentage of instances solved to a relative gap of $\varepsilon = 2\%$ (dotted line) or $\varepsilon = 5\%$ (solid line) within a given time $t \in [0, 7200]$. Figure 1 clearly illustrates that the i-C&CG method can solve more instances than the C&CG method. For example, when we use (20th, 80th) percentiles, the C&CG method can only solve about 60% of the instances, but the i-C&CG method can solve up to 90% of them with $\varepsilon = 5\%$ within the 2-hour time limit. Similarly, when we use (10th, 90th) percentiles, the C&CG method can solve only 20% of the instances, but the i-C&CG method can solve at least 50% of them.

Next, we analyze the final relative gap reported from the algorithms for instances that terminate at the 2-hour time limit. The curves in Figure 2 show the proportion of instances solved to a final relative gap less than a certain percentage when $\varepsilon = 2\%$. It is clear that the final relative gaps from the i-C&CG method are significantly less than those from the C&CG method. For most instances, the final relative gaps from the i-C&CG method are less than 10%. In contrast, the final relative gaps from the C&CG method are greater than 80% for at least 70% of the instances.

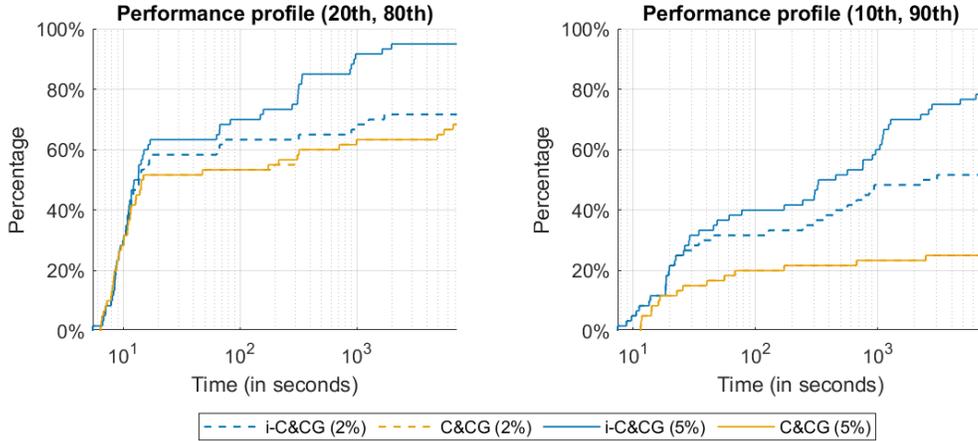


Figure 1: Time performance profile with $(c^f, c^v) = (1, 1/30)$ under two different lower and upper bound estimates: left – (20th, 80th) percentiles; right – (10th, 90th) percentiles

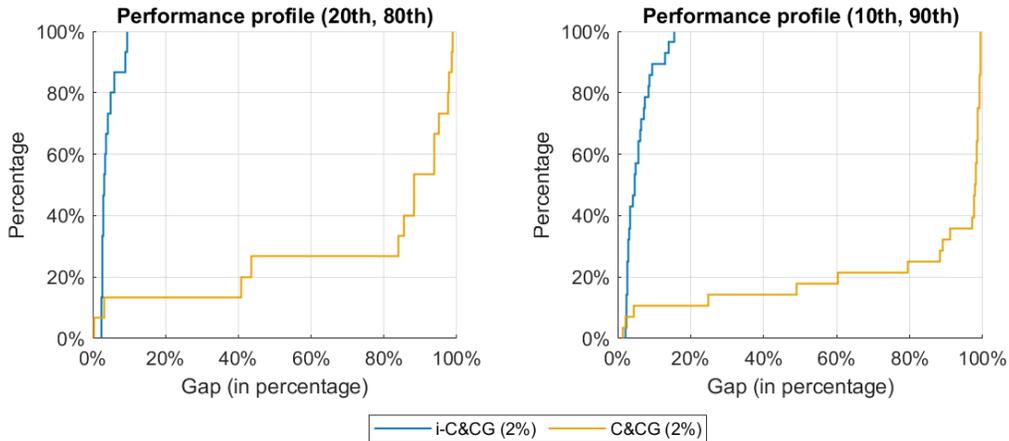


Figure 2: Gap performance profile for instances that both C&CG and i-C&CG exceed 2-hour time limit with $(c^f, c^v) = (1, 1/30)$ under two different lower and upper bound estimates: left – (20th, 80th) percentiles; right – (10th, 90th) percentiles

We attribute the differences in performance of the two algorithms to the following. Since the master problem of the DRORSP is a challenging MILP, solving the master problem requires significant computational effort. By allowing inexact solutions to challenging master problems, the i-C&CG method can solve problem instances more efficiently to a small relative gap. In contrast, we observe that the C&CG method spends a significant amount of time solving some master problems (in early iterations), thus terminating with a large final relative gap for most instances.

- [1] An, Y., Zeng, B., Zhang, Y., Zhao, L., 2014. Reliable p -median facility location problem: Two-stage robust models and algorithms. *Transportation Research Part B: Methodological* 64, 54–72.
- [2] Ayoub, J., Poss, M., 2016. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science* 13, 219–239.
- [3] Bertsimas, D., Shtern, S., 2018. A scalable algorithm for two-stage adaptive linear optimization. arXiv preprint arXiv:1807.02812 .
- [4] Bertsimas, D., Sim, M., 2004. The price of robustness. *Operations Research* 52, 35–53.
- [5] Denton, B.T., Miller, A.J., Balasubramanian, H.J., Huschka, T.R., 2010. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations Research* 58, 802–816.
- [6] Du, B., Zhou, H., Leus, R., 2020. A two-stage robust model for a reliable p -center facility location problem. *Applied Mathematical Modelling* 77, 99–114.
- [7] Gabrel, V., Murat, C., Thiele, A., 2014. Recent advances in robust optimization: An overview. *European Journal of Operational Research* 235, 471–483.
- [8] Jiang, R., Shen, S., Zhang, Y., 2017. Integer programming approaches for appointment scheduling with random no-shows and service durations. *Operations Research* 65, 1638–1656.
- [9] Jiang, R., Zhang, M., Li, G., Guan, Y., 2012. Benders’ decomposition for the two-stage security constrained robust unit commitment problem, in: *IIE Annual Conference Proceedings, Institute of Industrial and Systems Engineers (IISE)*. p. 1.
- [10] Liu, Y., Pichler, A., Xu, H., 2019. Discrete approximation and quantification in distributionally robust optimization. *Mathematics of Operations Research* 44, 19–37.
- [11] Mannino, C., Nilssen, E., Nordlander, T., 2010. SINTEF ICT: MSS-Adjusts Surgery data. <https://www.sintef.no/Projectweb/Health-care-optimization/Testbed/>. Accessed: 2022-04-18.
- [12] Neyshabouri, S., Berg, B.P., 2017. Two-stage robust optimization approach to elective surgery and downstream capacity planning. *European Journal of Operational Research* 260, 21–40.
- [13] Ruiz, C., Conejo, A.J., 2015. Robust transmission expansion planning. *European Journal of Operational Research* 242, 390–401.
- [14] Shapiro, A., Dentcheva, D., Ruszczyński, A., 2014. *Lectures on Stochastic Programming: Modeling and Theory*. Second ed., SIAM.

- [15] Shehadeh, K.S., Sanci, E., 2021. Distributionally robust facility location with bimodal random demand. *Computers & Operations Research* 134, 105257.
- [16] Thiele, A., Terry, T., Epelman, M., 2009. Robust linear optimization with recourse. Technical Report. Available in Optimization Online.
- [17] Tönissen, D.D., Arts, J.J., Shen, Z.J.M., 2021. A column-and-constraint generation algorithm for two-stage stochastic programming problems. *TOP* 29, 781–798.
- [18] Wang, Y., Zhang, Y., Tang, J., 2019. A distributionally robust optimization approach for surgery block allocation. *European Journal of Operational Research* 273, 740–753.
- [19] Xu, H., Liu, Y., Sun, H., 2018. Distributionally robust optimization with matrix moment constraints: Lagrange duality and cutting plane methods. *Mathematical Programming* 169, 489–529.
- [20] Zeng, B., Zhao, L., 2013. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters* 41, 457–461.
- [21] Zhang, X., Liu, X., 2022. A two-stage robust model for express service network design with surging demand. *European Journal of Operational Research* 299, 154–167.
- [22] Zhang, Y., Shen, S., Li, B., Mathieu, J.L., 2017. Two-stage distributionally robust optimal power flow with flexible loads, in: *2017 IEEE Manchester PowerTech, IEEE*. pp. 1–6.
- [23] Zugno, M., Conejo, A.J., 2015. A robust optimization approach to energy and reserve dispatch in electricity markets. *European Journal of Operational Research* 247, 659–671.

Appendix A. Two-stage Distributionally Robust Optimization Examples

By defining an ambiguity set \mathcal{P} (a subset of all probability measures with support Ξ), the two-stage distributionally robust optimization problem has the form

$$\min_{\mathbf{x} \in \mathcal{X}} \left\{ \mathbf{c}^\top \mathbf{x} + \sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[\min_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}, \boldsymbol{\xi})} \mathbf{q}^\top \mathbf{y} \right] \right\}. \quad (\text{A.1})$$

In the following, we provide two examples that under specific choices of the ambiguity set \mathcal{P} and the support Ξ , problem (A.1) can be reformulated in the form of (1). Thus, one can apply the C&CG and i-C&CG methods to solve these models.

Example 1. Mean-support ambiguity set $\mathcal{P}(\boldsymbol{\mu}) = \{\mathbb{P} \in \mathcal{P}(\Xi) \mid \mathbb{E}_{\mathbb{P}}(\boldsymbol{\xi}) = \boldsymbol{\mu}\}$ captures the support Ξ and mean $\boldsymbol{\mu}$ of the random vector $\boldsymbol{\xi}$. Due to its intuitive inputs, this ambiguity set has been employed in various applications; see, e.g., [8, 15]. Under the Slater-type condition [19], the worst-case expectation in (A.1) is equivalent to its dual:

$$\min_{\boldsymbol{\lambda} \in \mathbb{R}^l} \left\{ \boldsymbol{\mu}^\top \boldsymbol{\lambda} + \sup_{\boldsymbol{\xi} \in \Xi} \left\{ Q(\mathbf{x}, \boldsymbol{\xi}) - \boldsymbol{\xi}^\top \boldsymbol{\lambda} \right\} \right\}. \quad (\text{A.2})$$

We remark that the Slater-type condition is weaker than the classical Slater condition; see, e.g., discussions in [10]. The former is commonly used to ensure strong duality in distributionally robust optimization problems with moment-based ambiguity sets; see, e.g., [19]. Hence, problem (A.2) reduces to the form of problem (1) with $(\mathbf{x}, \boldsymbol{\lambda})$ as the first-stage decision.

Example 2. The mean-absolute-deviation ambiguity set $\mathcal{P}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \{\mathbb{P} \in \mathcal{P}(\Xi) \mid \mathbb{E}_{\mathbb{P}}(\boldsymbol{\xi}) = \boldsymbol{\mu}, \mathbb{E}_{\mathbb{P}}|\boldsymbol{\xi} - \boldsymbol{\mu}| \leq \boldsymbol{\sigma}\}$ captures the support Ξ and mean $\boldsymbol{\mu}$ information, and requires the absolute deviation from mean is no more than $\boldsymbol{\sigma}$ [18, 22]. Again, under the Slater-type condition [19], we can reformulate the worst-case expectation in (A.1) as

$$\min_{\boldsymbol{\lambda} \in \mathbb{R}^l, \boldsymbol{\rho} \in \mathbb{R}^l} \left\{ \boldsymbol{\mu}^\top \boldsymbol{\lambda} + \boldsymbol{\rho}^\top \boldsymbol{\sigma} + \sup_{\boldsymbol{\xi} \in \Xi} \left\{ Q(\mathbf{x}, \boldsymbol{\xi}) - \boldsymbol{\xi}^\top \boldsymbol{\lambda} - \boldsymbol{\rho}^\top |\boldsymbol{\xi} - \boldsymbol{\mu}| \right\} \right\}, \quad (\text{A.3})$$

where the absolute value is computed entry-wisely. We can introduce auxiliary variables to linearize terms in absolute value, leading to the form of problem (1).

Appendix B. Proof of Proposition 1

To prove the validity of L^ℓ in Proposition 1, we first provide Proposition 5, which addresses the inexactness associated with the additional constraint (4c).

Proposition 5. *If the value \bar{L} in (4c) is greater than the optimal value v^* of problem (1) at some iteration j (i.e., $\bar{L} > v^*$), the optimal value v_j^* of (4) is \bar{L} .*

Proof. First, note that constraint (4c) implies $v_j^* \geq \bar{L}$. For the opposite direction that $v_j^* \leq \bar{L}$, it suffices to show that there exists a feasible solution with objective \bar{L} . We claim that $(\mathbf{x}, \delta) = (\mathbf{x}^*, \bar{L} - \mathbf{c}^\top \mathbf{x}^*)$ is feasible to (4) with objective value \bar{L} , where \mathbf{x}^* is an optimal solution to problem

(1). To verify our claim, we only need to show that (4b) is satisfied. Indeed, since $v^* < \bar{L}$ by our assumption, we have

$$c^\top x^* + \max_{\xi \in \Xi} Q(x^*, \xi) = v^* < \bar{L}. \quad (\text{B.1})$$

Therefore, we obtain the desired inequality

$$\max_{\xi \in \mathcal{S}} Q(x^*, \xi) \leq \max_{\xi \in \Xi} Q(x^*, \xi) < \bar{L} - c^\top x^* = \delta,$$

where the first inequality follows from $\mathcal{S} \subseteq \Xi$ and the second one follows from (B.1). \square

Proof of Proposition 1. Suppose, on the contrary, that L^ℓ is not a valid lower bound for v^* , i.e., $L^\ell > v^*$. Consider the master problem (4) at iteration ℓ and its optimal value v_ℓ^* . Therefore, \bar{L} in this proof is the right-hand-side value of (4c) when solving this master problem. First, we have $L^\ell > \bar{L}$ from the only updating criterion for ℓ in step 1.2. Next, consider the following two cases. If $\bar{L} > v^*$, by Proposition 5, we have the inequality $v_\ell^* = \bar{L} < L^\ell$, contradicting that L^ℓ is a lower bound on v_ℓ^* . If $\bar{L} \leq v^*$, the solution $(\mathbf{x}, \delta) = (\mathbf{x}^*, \max_{\xi \in \Xi} Q(\mathbf{x}^*, \xi))$ is feasible to the master problem with objective v^* , where \mathbf{x}^* is an optimal solution to problem (1). Therefore, this leads to the same contradiction that $v_\ell^* \leq v^* < L^\ell$. This concludes that L^ℓ is a valid lower bound. \square

Appendix C. Proof of Corollary 3

Proof. Note that the new updating rule $\bar{L} \leftarrow U^j$ guarantees that \bar{L} is always a valid lower bound on v^* . Therefore, we can set $\ell = j$ at each iteration and the desired results follow from the first part of the proof in Proposition 2. \square

Appendix D. Pseudocode for i-C&CG variants

We present the pseudo codes for the two variant of i-C&CG discussed in Section 4.3 in the following two subsections, respectively. For the ease of reading, we highlight additional elements in these two variants, when compared with the general i-C&CG framework presented in Algorithm 2, in orange.

Appendix D.1. Exploitation Frequency Variant

Algorithm 3 presents the exploitation frequency variant of the i-C&CG method. In this variant, we have an additional parameter f^{exploit} to control the exploitation frequency. Specifically, in the backtracking routine, if $j - \ell > f^{\text{exploit}}$ (i.e., exploitation has not been reached at least f^{exploit} times), the algorithm proceeds to the exploitation step. In such a case ($\ell \ll j$), the scenario set is substantially enlarged, and thus, it is likely that a new valid lower bound could be found via exploitation.

Appendix D.2. Time Limit Variant

Algorithm 4 presents the time limit variant of the i-C&CG method. In this variant, there are two additional parameters, τ and β , to control the time limit of the master problem and the increase for the time limit, respectively. Specifically, in step 1.1, we impose a time limit τ when solving the

Algorithm 3: Inexact column-and-constraint (i-C&CG) method, exploitation frequency variant

Initialization: $\bar{L} \leftarrow 0, \bar{U} \leftarrow \infty, \varepsilon \in (0, 1), \tilde{\varepsilon} \in (0, \varepsilon/(1 + \varepsilon)), \{\varepsilon_{MP}^j \in (0, 1)\}_{j \in \mathbb{N}}, \alpha \in (0, 1),$
 $\mathcal{S} \leftarrow \emptyset, j \leftarrow 1, \ell \leftarrow 0, f^{\text{exploit}} \in \mathbb{N}.$

1. Master problem.

1.1. Solve the master problem (4) to within a relative optimality gap of ε_{MP}^j .
 Record the best feasible solution (\mathbf{x}^j, δ^j) found.

1.2. Record a lower bound $L^j \geq \bar{L}$ and upper bound $U^j = \mathbf{c}^\top \mathbf{x}^j + \delta^j$ of v_j^* .
 If $L^j > \bar{L}$, then set $\ell \leftarrow j$.

1.3. Set $\bar{L} \leftarrow U^j$.

2. Subproblem. Solve the subproblem (3) for fixed $\mathbf{x} = \mathbf{x}^j$.

Record the optimal solution $\boldsymbol{\xi}^*$ and value D^j .

Set $\bar{U} \leftarrow \min \{\bar{U}, \mathbf{c}^\top \mathbf{x}^j + D^j\}$.

3. Optimality test and backtracking routine.

If $(\bar{U} - L^\ell)/\bar{U} < \varepsilon$, then terminate and return \mathbf{x}^j ; otherwise, do the following.

- **Exploitation:** If $(\bar{U} - U^j)/\bar{U} < \tilde{\varepsilon}$ or $j - \ell > f^{\text{exploit}}$, then set $j \leftarrow \ell$ and $\bar{L} \leftarrow L^\ell$.

Set $\varepsilon_{MP}^j \leftarrow \alpha \varepsilon_{MP}^j$ for all $j \geq \ell$ and go back to step 1.

- **Exploration:** If $(\bar{U} - U^j)/\bar{U} \geq \tilde{\varepsilon}$, then go to step 4.

4. Scenario set enlargement.

Enlarge the scenario set $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{\xi}^*\}$.

Update $j \leftarrow j + 1$ and go back to step 1.

master problem. This is useful when the relative gap ε_{MP}^j is difficult to achieve. To control the inaccuracies due to the imposed time limit, in the exploitation step, the algorithm increases the time limit τ by β .

Appendix E. Details of the DRORSP

Appendix E.1. Master-subproblem Framework

In this section, we derive the master problem and subproblem for the DRORSP that facilitates the use of the C&CG and i-C&CG methods. In view of (13), we define the second-stage recourse function as

$$Q(\mathbf{y}, \mathbf{d}) := \sum_{r \in R} \left(\sum_{i \in I} y_{ir} d_i - T \right)^+ = \underset{\mathbf{w}}{\text{minimize}} \quad \sum_{r \in R} w_r \quad (\text{E.1a})$$

$$\text{subject to} \quad w_r \geq \sum_{i \in I} y_{ir} d_i - T, \quad \forall r \in R, \quad (\text{E.1b})$$

$$w_r \geq 0, \quad \forall r \in R. \quad (\text{E.1c})$$

We first reformulate the inner maximization problem in (13) over $\mathbb{P} \in \mathcal{P}$ as defined in (12). As shown in [18], this inner maximization problem is equivalent to its dual presented in Proposition 6.

Algorithm 4: Inexact column-and-constraint (i-C&CG) method, time limit variant

Initialization: $\bar{L} \leftarrow 0$, $\bar{U} \leftarrow \infty$, $\varepsilon \in (0, 1)$, $\tilde{\varepsilon} \in (0, \varepsilon/(1 + \varepsilon))$, $\{\varepsilon_{MP}^j \in (0, 1)\}_{j \in \mathbb{N}}$, $\alpha \in (0, 1)$,
 $\mathcal{S} \leftarrow \emptyset$, $j \leftarrow 1$, $\ell \leftarrow 0$, $\tau > 0$, $\beta > 0$.

1. Master problem.

1.1. Solve the master problem (4) to within a relative optimality gap of ε_{MP}^j
or terminate if the time limit τ is exceeded.

Record the best feasible solution (\mathbf{x}^j, δ^j) found.

1.2. Record a lower bound $L^j \geq \bar{L}$ and upper bound $U^j = \mathbf{c}^\top \mathbf{x}^j + \delta^j$ of v_j^* .

If $L^j > \bar{L}$, then set $\ell \leftarrow j$.

1.3. Set $\bar{L} \leftarrow U^j$.

2. Subproblem. Solve the subproblem (3) for fixed $\mathbf{x} = \mathbf{x}^j$.

Record the optimal solution $\boldsymbol{\xi}^*$ and value D^j .

Set $\bar{U} \leftarrow \min \{\bar{U}, \mathbf{c}^\top \mathbf{x}^j + D^j\}$.

3. Optimality test and backtracking routine.

If $(\bar{U} - L^\ell)/\bar{U} < \varepsilon$, then terminate and return \mathbf{x}^j ; otherwise, do the following.

- **Exploitation:** If $(\bar{U} - U^j)/\bar{U} < \tilde{\varepsilon}$, then set $j \leftarrow \ell$ and $\bar{L} \leftarrow L^\ell$.

Set $\varepsilon_{MP}^j \leftarrow \alpha \varepsilon_{MP}^j$ for all $j \geq \ell$, set $\tau \leftarrow \tau + \beta$, and go back to step 1.

- **Exploration:** If $(\bar{U} - U^j)/\bar{U} \geq \tilde{\varepsilon}$, then go to step 4.

4. Scenario set enlargement.

Enlarge the scenario set $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{\xi}^*\}$.

Update $j \leftarrow j + 1$ and go back to step 1.

Proposition 6. The problem $\sup_{\mathbf{Q} \in \mathcal{P}} \mathbb{E}_{\mathbf{Q}}[Q(\mathbf{y}, \mathbf{d})]$ with \mathcal{P} defined in (12) is equivalent to

$$\underset{\boldsymbol{\eta}, \boldsymbol{\varphi}}{\text{minimize}} \quad \sum_{i \in I} (\mu_i \eta_i + \sigma_i \varphi_i) + \sup_{\mathbf{d} \in \Xi} \left\{ Q(\mathbf{y}, \mathbf{d}) - \sum_{i \in I} (d_i \eta_i + |d_i - \mu_i| \varphi_i) \right\} \quad (\text{E.2a})$$

$$\text{subject to} \quad \varphi_i \geq 0, \quad \forall i \in I. \quad (\text{E.2b})$$

Proof. First, note that Ξ is compact. Moreover, $Q(\mathbf{x}, \mathbf{d})$ is a continuous function in \mathbf{d} , as well as $\phi_i^{\text{eq}}(\mathbf{d}) := d_i$ and $\phi_i^{\text{ineq}}(\mathbf{d}) := |d_i - \mu_i|$ for all $i \in I$. By strong duality of moment problems (see, e.g., [14]), the problem $\sup_{\mathbf{Q} \in \mathcal{P}} \mathbb{E}_{\mathbf{Q}}[Q(\mathbf{y}, \mathbf{d})]$ is equivalent to

$$\underset{\boldsymbol{\eta}, \boldsymbol{\varphi}, \theta}{\text{minimize}} \quad \sum_{i \in I} (\mu_i \eta_i + \sigma_i \varphi_i) + \theta \quad (\text{E.3a})$$

$$\text{subject to} \quad \sum_{i \in I} (d_i \eta_i + |d_i - \mu_i| \varphi_i) + \theta \geq Q(\mathbf{y}, \mathbf{d}), \quad \forall \mathbf{d} \in \Xi, \quad (\text{E.3b})$$

$$\varphi_i \geq 0, \quad \forall i \in I. \quad (\text{E.3c})$$

From (E.3b), we have

$$\theta \geq Q(\mathbf{y}, \mathbf{d}) - \sum_{i \in I} (d_i \eta_i + |d_i - \mu_i| \varphi_i), \quad \forall \mathbf{d} \in \Xi.$$

Since θ is unrestricted and the objective is to minimize θ , this shows the equivalence between (E.3b)

and (E.2). □

In view of Proposition 6, the DRORSP (13) is equivalent to

$$\underset{\mathbf{x}, \mathbf{y}, \boldsymbol{\eta}, \boldsymbol{\varphi}, \delta}{\text{minimize}} \quad \sum_{r \in R} c^f x_r + c^v \sum_{i \in I} (\mu_i \eta_i + \sigma_i \varphi_i) + c^v \delta \quad (\text{E.4a})$$

$$\text{subject to} \quad y_{ir} \leq x_r, \quad \forall i \in I, r \in R, \quad (\text{E.4b})$$

$$\sum_{r \in R} y_{ir} = 1, \quad \forall i \in I, \quad (\text{E.4c})$$

$$\delta \geq Q(\mathbf{d}, \boldsymbol{\xi}) - \sum_{i \in I} (d_i \eta_i + |d_i - \mu_i| \varphi_i), \quad \forall \mathbf{d} \in \Xi, \quad (\text{E.4d})$$

$$x_r \in \{0, 1\}, y_{ir} \in \{0, 1\}, \quad \forall i \in I, r \in R. \quad (\text{E.4e})$$

Therefore, given a subset of scenario $\mathcal{S} \in \Xi$, the master problem is given by

$$\underset{\mathbf{x}, \mathbf{y}, \boldsymbol{\eta}, \boldsymbol{\varphi}, \delta, \mathbf{w}}{\text{minimize}} \quad \sum_{r \in R} c^f x_r + c^v \sum_{i \in I} (\mu_i \eta_i + \sigma_i \varphi_i) + c^v \delta \quad (\text{E.5a})$$

$$\text{subject to} \quad y_{ir} \leq x_r, \quad \forall i \in I, r \in R, \quad (\text{E.5b})$$

$$\sum_{r \in R} y_{ir} = 1, \quad \forall i \in I, \quad (\text{E.5c})$$

$$\delta \geq \sum_{r \in R} w_r^k - \sum_{i \in I} (d_i^k \eta_i + |d_i^k - \mu_i| \varphi_i), \quad \forall k \in \mathcal{S}, \quad (\text{E.5d})$$

$$w_r^k \geq \sum_{i \in I} y_{ir} d_i^k - T, \quad \forall r \in R, k \in \mathcal{S}, \quad (\text{E.5e})$$

$$x_r \in \{0, 1\}, y_{ir} \in \{0, 1\}, \quad \forall i \in I, r \in R. \quad (\text{E.5f})$$

Finally, we provide a tractable MILP reformulation of the subproblem in Proposition 7.

Proposition 7. *Let $\underline{\Delta}_i = \mu_i - \underline{d}_i$ and $\overline{\Delta}_i = \overline{d}_i - \mu_i$ for all $i \in I$. The subproblem $\sup_{\mathbf{d} \in \Xi} \left\{ Q(\mathbf{y}, \mathbf{d}) - \sum_{i \in I} (d_i \eta_i + |d_i - \mu_i| \varphi_i) \right\}$ is equivalent to*

$$\underset{\boldsymbol{\pi}, \mathbf{b}, \boldsymbol{\zeta}}{\text{maximize}} \quad -T \sum_{r \in R} \pi_r + \sum_{i \in I} \left[\mu_i \left(\sum_{r \in R} \pi_r y_{ir} - \eta_i \right) - \underline{\Delta}_i \left(\sum_{r \in R} \zeta_{ir}^1 y_{ir} - \eta_i b_i^1 \right) + \overline{\Delta}_i \left(\sum_{r \in R} \zeta_{ir}^2 y_{ir} - \eta_i b_i^2 \right) - \varphi_i \left(b_i^1 \underline{\Delta}_i + b_i^2 \overline{\Delta}_i \right) \right] \quad (\text{E.6a})$$

$$\text{subject to} \quad 0 \leq \pi_r \leq 1, \quad \forall r \in R, \quad (\text{E.6b})$$

$$\zeta_{ir}^1 \geq 0, \zeta_{ir}^1 \geq \pi_r + b_i^1 - 1, \zeta_{ir}^1 \leq \pi_r, \zeta_{ir}^1 \leq b_r^1, \quad \forall i \in I, r \in R, \quad (\text{E.6c})$$

$$\zeta_{ir}^2 \geq 0, \zeta_{ir}^2 \geq \pi_r + b_i^2 - 1, \zeta_{ir}^2 \leq \pi_r, \zeta_{ir}^2 \leq b_r^2, \quad \forall i \in I, r \in R, \quad (\text{E.6d})$$

$$b_i^1 + b_i^2 \leq 1, \quad \forall i \in I, \quad (\text{E.6e})$$

$$b_i^1 \in \{0, 1\}, b_i^2 \in \{0, 1\}, \quad \forall i \in I. \quad (\text{E.6f})$$

Proof. First, by LP strong duality, we have

$$Q(\mathbf{y}, \mathbf{d}) = \underset{\boldsymbol{\pi}}{\text{maximize}} \quad \sum_{r \in R} \pi_r \left(\sum_{i \in I} y_{ir} d_i - T \right) \quad (\text{E.7a})$$

$$\text{subject to} \quad 0 \leq \pi_r \leq 1, \quad \forall r \in R. \quad (\text{E.7b})$$

Therefore, we can reformulate the subproblem as a single maximization problem:

$$\underset{\boldsymbol{\pi}, \mathbf{d}}{\text{maximize}} \quad -T \sum_{r \in R} \pi_r + \sum_{i \in I} \left[\left(\sum_{r \in R} \pi_r y_{ir} - \eta_i \right) d_i - \varphi_i |d_i - \mu_i| \right] \quad (\text{E.8a})$$

$$\text{subject to} \quad 0 \leq \pi_r \leq 1, \quad \forall r \in R, \quad (\text{E.8b})$$

$$\underline{d}_i \leq d_i \leq \bar{d}_i, \quad \forall i \in I. \quad (\text{E.8c})$$

Since the objective function of (E.8) is piecewise linear in d_i with two pieces on $[\underline{d}_i, \mu_i]$ and $[\mu_i, \bar{d}_i]$, we have that the optimal solution $d_i^* \in \{\underline{d}_i, \mu_i, \bar{d}_i\}$. Let b_i^1 and b_i^2 be two binary variables and let $d_i = \mu_i - b_i^1 \underline{\Delta}_i + b_i^2 \bar{\Delta}_i$. Then, problem (E.8) is equivalent to

$$\underset{\boldsymbol{\pi}, \mathbf{b}}{\text{maximize}} \quad -T \sum_{r \in R} \pi_r + \sum_{i \in I} \left[\mu_i \left(\sum_{r \in R} \pi_r y_{ir} - \eta_i \right) - \underline{\Delta}_i \left(\sum_{r \in R} \pi_r b_i^1 y_{ir} - \eta_i b_i^1 \right) \right. \\ \left. + \bar{\Delta}_i \left(\sum_{r \in R} \pi_r b_i^2 y_{ir} - \eta_i b_i^2 \right) - \varphi_i \left(b_i^1 \underline{\Delta}_i + b_i^2 \bar{\Delta}_i \right) \right] \quad (\text{E.9a})$$

$$\text{subject to} \quad 0 \leq \pi_r \leq 1, \quad \forall r \in R, \quad (\text{E.9b})$$

$$b_i^1 + b_i^2 \leq 1, \quad \forall i \in I, \quad (\text{E.9c})$$

$$b_i^1 \in \{0, 1\}, b_i^2 \in \{0, 1\}, \quad \forall i \in I. \quad (\text{E.9d})$$

Note that (E.9) is non-linear due to the quadratic terms $b_i^1 \pi_r$ and $b_i^2 \pi_r$ in the objective. Defining $\zeta_{ir}^1 = b_i^1 \pi_r$ and $\zeta_{ir}^2 = b_i^2 \pi_r$ and introducing the McCormick inequalities, we can reformulate (E.9) into (E.6). \square

Appendix E.2. Symmetry-breaking Constraints

As in [18], we apply the following symmetry-breaking constraints in the model under the realistic assumption that $|I| \geq |R|$ (see [5] for a detailed explanation):

$$x_t \geq x_{r+1}, \quad \forall r \in \{1, \dots, |R| - 1\}, \quad (\text{E.10})$$

$$\sum_{r=1}^i y_{ir} = 1, \quad \forall i \in \{1, \dots, |R| - 1\}, \quad (\text{E.11})$$

$$\sum_{r=j}^{\min\{i, |R|\}} y_{ir} \leq \sum_{u=j-1}^{i-1} y_{u, j-1} \quad \forall j \in \{2, \dots, |R|\}, i \in \{j, \dots, |R|\}. \quad (\text{E.12})$$

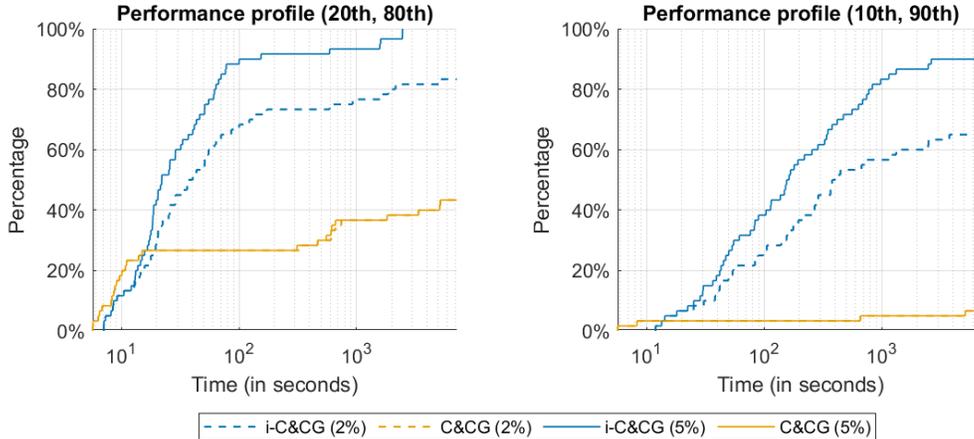


Figure F.3: Time performance profile with $(c^f, c^v) = (1, 1/120)$ under two different lower and upper bound estimates: left – (20th, 80th) percentiles; right – (10th, 90th) percentiles

Appendix F. Additional Computational Results

We provide additional computational results under cost structure $(c^f, c^v) = (1, 1/120)$. Figure F.3 shows the time performance profile under two different lower and upper bound estimates. Similar to the observations in Section 5.3, while the computational performance between the C&CG and i-C&CG methods for easier instances is similar, we observe that the i-C&CG method is more efficient in solving challenging instances. When we use (20th, 80th) percentiles, the C&CG method can only solve about 40% of the instances, but the i-C&CG method can solve more than 80% of them (indeed, all the instances when $\varepsilon = 5\%$) within the 2-hour time limit. When we use (10th, 90th) percentiles, the C&CG method can solve only less than 10% of the instances while the i-C&CG method can solve up to 90% with $\varepsilon = 5\%$.

Finally, we also compare the final relative gap of instances that both the C&CG and i-C&CG methods terminate due to the time limit. Figure F.4 shows the corresponding results when $(c^f, c^v) = (1, 1/120)$ and $\varepsilon = 2\%$. Similar to the observations in Section 5.3, the final relative gaps from the i-C&CG method are smaller than that of the C&CG method. For instance, when we use (20th, 80th) percentiles, all the final relative gaps are within 10%, but those from the C&CG method are still greater than 20% (and many of them are even greater than 80%). We observe similar results when using (10th, 90th) percentiles. These results further conclude that our proposed i-C&CG method could be efficient when solving challenging instances (i.e., with difficult master problems).

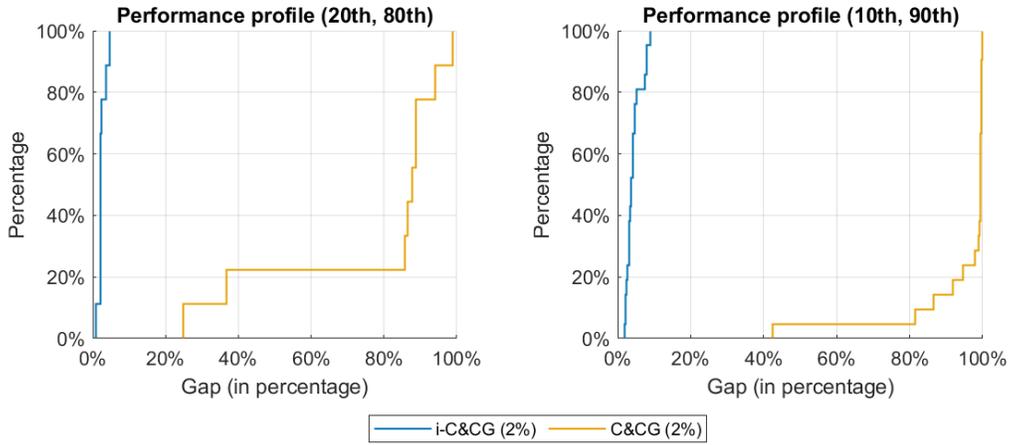


Figure F.4: Gap performance profile for instances that both the C&CG and i-C&CG methods exceed the 2-hour time limit with $(c^f, c^v) = (1, 1/120)$ under two different lower and upper bound estimates: left – (20th, 80th) percentiles; right – (10th, 90th) percentiles