



HAL
open science

A semi-infinite constraint generation algorithm for adjustable robust optimization

Patxi Flambard, Ayşe N Arslan, Boris Detienne

► **To cite this version:**

Patxi Flambard, Ayşe N Arslan, Boris Detienne. A semi-infinite constraint generation algorithm for adjustable robust optimization. ROADEF 2025 - 26ème congrès annuel de la société française de recherche opérationnelle et d'aide à la décision, Feb 2025, Champ-sur-Marne, France. hal-04845833v2

HAL Id: hal-04845833

<https://hal.science/hal-04845833v2>

Submitted on 14 Feb 2025

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A semi-infinite constraint generation algorithm for adjustable robust optimization

Patxi Flambard¹, Ayse N. Arslan¹, Boris Detienne^{1,2}

¹ Centre Inria de l'université de Bordeaux

{patxi.flambard@inria.fr, ayse-nur.arslan@inria.fr}

² Université de Bordeaux

boris.detienne@u-bordeaux.fr

Keywords : *adjustable robust optimization, constraint generation, constraint-and-column generation, semi-infinite constraints, location-transportation problem.*

1 Introduction

Robust optimization has emerged as a prevalent method for dealing with uncertain parameters in optimization. Static robust optimization problems with polyhedral or convex uncertainty sets lead to deterministic equivalent formulations that are often in the same complexity class as their deterministic counterparts [4, 2]. However, robust optimization problems in which some of the decisions can be delayed until after the uncertain parameters are revealed, known as adjustable robust optimization, still pose theoretical and numerical challenges. In this paper, we focus on adjustable robust optimization problems with fixed and continuous recourse. Although these problems have received significant attention from the research community with various approaches for their exact and approximate solution [1, 3, 5, 6], existing exact algorithms still do not perform well. In this paper, we propose a novel reformulation of the problem with an exponential number of semi-infinite constraints and develop a decomposition algorithm to solve it. We then show through numerical experiments that this algorithm significantly outperforms the state-of-the-art exact algorithms on a location-transportation problem.

Section 2 describes our methodological contributions, Section 3 presents our numerical results, and we finally provide conclusions and future research directions in Section 4.

2 Methodological development

We study the exact solution of linear adjustable robust optimization problems with continuous and fixed recourse formalized as follows:

$$\begin{aligned} \min_{x \in \mathcal{X}} c^\top x + \max_{\xi \in \Xi} \min_{y \in \mathbb{R}_+^{n_y}} 0 & \quad (P) \\ \text{s.t. } Wy & \geq h(\xi) - T(\xi)x. \end{aligned}$$

Here $x \in \mathcal{X}$ denotes the first-stage variables, $\xi \in \Xi$ denotes the uncertain parameters and $y \in \mathcal{Y}(x, \xi) := \{y \in \mathbb{R}_+^{n_y} | Wy \geq h(\xi) - T(\xi)x\}$ denotes the second stage variables with their respective feasible regions. Although (P) is written with a second-stage feasibility problem, it subsumes the cases where the second-stage problem is an optimization problem. Throughout, we assume that \mathcal{X} is compact, Ξ is a polytope, $T(\cdot)$ and $h(\cdot)$ are affine functions of $\xi \in \Xi$, and (P) has an optimal solution.

In the remainder of this section, we first recall the two classical algorithms, namely constraint generation (CG) [3] and constraint-and-column generation (C&CG) [6]. We then present a novel reformulation of (P) and develop a decomposition algorithm to solve it.

2.1 CG and C&CG algorithms

We start by reformulating (P) as a monolithic mixed-integer linear programming problem with an exponential number of constraints (and variables). To do so, we first integrate the effect of uncertainty into the constraints of the first-stage variables, x , so as to obtain an equivalent nonlinear monolithic reformulation of (P) :

$$\min_{x \in \mathcal{X}} c^\top x \quad (1)$$

$$\text{s.t. } \max_{\xi \in \Xi} \min_{\substack{y \in \mathbb{R}_+^{n_y} \\ Wy \geq h(\xi) - T(\xi)x}} 0 \leq 0. \quad (2)$$

(1)-(2) is equivalent to (P) since, for given $x \in \mathcal{X}$, the left-hand-side of (2) is equal to $+\infty$ if and only if there exists $\xi \in \Xi$ such that the inner minimization problem in (P) is infeasible.

We next use linear programming duality on the inner minimization problem of constraint (2). Since the objective function of the primal problem is zero, the corresponding dual feasible set, $\Pi := \{\pi \in \mathbb{R}_+^{m_y} \mid W^\top \pi \leq \mathbf{0}\}$, is a pointed polyhedral cone. Moreover, the dual solution $\pi = \mathbf{0}$ is feasible for any $W \in \mathbb{R}^{m_y \times n_y}$. We conclude that the primal is infeasible if and only if a dual solution with a strictly positive objective value exists. We, therefore, obtain:

$$\begin{aligned} & \min_{x \in \mathcal{X}} c^\top x \\ & \text{s.t. } \max_{\xi \in \Xi} \max_{\pi \in \Pi} \pi^\top (h(\xi) - T(\xi)x) \leq 0. \end{aligned} \quad (3)$$

We finally transform the maximization over $\xi \in \Xi$ and $\pi \in \Pi$ into a finite but exponential number of constraints. Indeed, restricting Ξ to its extreme points, $\text{ext}(\Xi)$, is sufficient to ensure a valid formulation [3]. We can further restrict Π to its normalized extreme directions, denoted $\text{dir}(\Pi)$, since the left-hand-side of each constraint is a linear programming problem defined over a pointed cone and the right-hand-side is $\mathbf{0}$. We then obtain the following reformulation of (P) on which the CG algorithm is based:

$$\begin{aligned} & \min_{x \in \mathcal{X}} c^\top x \\ & \text{s.t. } \pi^\top (h(\xi) - T(\xi)x) \leq 0 \quad \forall (\xi, \pi) \in \text{ext}(\Xi) \times \text{dir}(\Pi). \end{aligned} \quad (4)$$

The reformulation of (P) on which the C&CG algorithm is based can, similarly, be derived from (1)-(2) by introducing second-stage variables y^ξ for each realization $\xi \in \text{ext}(\Xi)$ and creating $|\text{ext}(\Xi)|$ copies of the recourse problem to obtain:

$$\begin{aligned} & \min_{x \in \mathcal{X}, y \in \mathbb{R}_+^{|\text{ext}(\Xi)| \times n_y}} c^\top x \\ & \text{s.t. } Wy^\xi \geq h(\xi) - T(\xi)x \quad \forall \xi \in \text{ext}(\Xi). \end{aligned} \quad (5)$$

Both CG and C&CG algorithms will iteratively solve and reinforce a so-called master program, which is a relaxation of (4) or (5), until the obtained first-stage solution $\hat{x} \in \mathcal{X}$ is proven to be feasible and hence optimal for (P) . To prove the feasibility of a given first-stage relaxation solution $\hat{x} \in \mathcal{X}$, a separation subproblem is solved:

$$\max_{\xi \in \Xi} \max_{\pi \in \Pi} \pi^\top (h(\xi) - T(\xi)\hat{x}). \quad (SP(\hat{x}, \Xi, \Pi))$$

If the objective value of $(SP(\hat{x}, \Xi, \Pi))$ is 0, then \hat{x} is feasible for (3), hence also feasible for (4) and (5), and optimal for (P) . Otherwise, the master problem should be reinforced using an optimal solution of $(SP(\hat{x}, \Xi, \Pi))$ in order to cut off the current infeasible solution \hat{x} .

Algorithm 1: Unified framework for CG and C&CG algorithms

```

1  $(MP) := \min_{x \in \mathcal{X}} c^\top x$ 
2 do
3   Let  $\hat{x}$  be an optimal solution of  $(MP)$ 
4   Let  $(\xi^*, \pi^*)$  be an optimal solution of  $(SP(\hat{x}, \Xi, \Pi))$ 
5   if  $\pi^{*\top}(h(\xi^*) - T(\xi^*)\hat{x}) > 0$  then
6     CG : add constraint  $\pi^{*\top}(h(\xi^*) - T(\xi^*)x) \leq 0$  to  $(MP)$ 
7     C&CG : add  $y^{\xi^*} \in \mathbb{R}^{n_y}$  with constraints  $Wy^{\xi^*} \geq h(\xi^*) - T(\xi^*)x$  to  $(MP)$ 
8 while  $\pi^{*\top}(h(\xi^*) - T(\xi^*)\hat{x}) > 0$ 
Output: An optimal solution of  $(P) : \hat{x}$ 

```

Details of the CG and C&CG algorithms are presented in Algorithm 1 in a unified manner. Assuming that $(SP(\hat{x}, \Xi, \Pi))$ returns extreme points and extreme directions of Ξ and Π , respectively, for $\hat{x} \in \mathcal{X}$, the CG and C&CG algorithms converge in at most $|\text{ext}(\Xi)| \times |\text{dir}(\Pi)|$ and $|\text{ext}(\Xi)|$ iterations, respectively. In addition to an improved worst-case complexity, the C&CG algorithm often has better numerical performance compared to CG [6]. This can be explained by the quality of the dual bound obtained by the solution of the C&CG master problem compared to that of CG. Indeed, by construction, the dual bound obtained by solving the C&CG master is always larger than that of the CG master provided that the two contain the same uncertainty realizations $\tilde{\Xi}$ [6]. On the other hand, adding an entire second-stage system to the master problem at each iteration within the C&CG algorithm can hinder the solution of the master problem, especially if many realizations are needed for convergence, or the second-stage system has a complex structure.

2.2 Semi-infinite constraint generation algorithm

The proposed algorithm aims to combine the advantages of CG and C&CG algorithms, that is, maintaining a smaller master problem while providing good-quality dual bounds. We start by deriving a reformulation of (P) on which the algorithm will be based. To do so, we change the maximum operator over $\xi \in \Xi$ and $\pi \in \Pi$ in (3) to enumeration operators. We further restrict Ξ to its extreme points $\text{ext}(\Xi)$ to obtain equivalently:

$$\begin{aligned}
 & \min_{x \in \mathcal{X}} c^\top x & (6) \\
 & \text{s.t. } \pi^\top(h(\xi) - T(\xi)x) \leq 0 & \forall \xi \in \text{ext}(\Xi), \forall \pi \in \Pi.
 \end{aligned}$$

Model (6) is composed of an exponential number of semi-infinite constraints due to the fact that Π is a polyhedral cone. To address the exponential nature of the constraint, we employ a decomposition algorithm based on a relaxation of (6), similar to CG and C&CG algorithms, defined by restricting the set of constraints to $\tilde{\Xi} \subseteq \text{ext}(\Xi)$:

$$\begin{aligned}
 & \min_{x \in \mathcal{X}} c^\top x & (7) \\
 & \text{s.t. } \pi^\top(h(\xi) - T(\xi)x) \leq 0 & \forall \xi \in \tilde{\Xi}, \forall \pi \in \Pi.
 \end{aligned}$$

Then, to address the semi-infinite nature of the constraints, we can interpret them as classical static robust constraints. One approach is to apply LP duality theory to obtain the master problem of the C&CG algorithm with Ξ relaxed to $\tilde{\Xi}$. The second approach is to generate the semi-infinite constraints dynamically as needed. This approach restricts the set of constraints relative to each realization $\xi \in \tilde{\Xi}$ to $\tilde{\Pi}^\xi \subseteq \text{dir}(\Pi)$. Let us further denote by $\tilde{\Pi} := (\tilde{\Pi}^\xi)_{\xi \in \tilde{\Xi}}$ the family of sets of generated dual solutions associated with realizations in $\tilde{\Xi}$. The master problem of the constraint generation algorithm solving (7) can be written as:

$$\begin{aligned}
 & \min_{x \in \mathcal{X}} c^\top x & (8) \\
 & \text{s.t. } \pi^\top(h(\xi) - T(\xi)x) \leq 0 & \forall \xi \in \tilde{\Xi}, \forall \pi \in \tilde{\Pi}^\xi.
 \end{aligned}$$

Any solution $\hat{x} \in \mathcal{X}$ of (8) is feasible for (7) if for all $\xi \in \tilde{\Xi}$ we have that $z_{\text{SP}}(\hat{x}, \xi) := \max_{\pi \in \Pi} \pi^\top (h(\xi) - T(\xi)\hat{x}) \leq 0$. We refer to this linear problem as $(SP(\hat{x}, \{\xi\}, \Pi))$. Otherwise, for some $\hat{\xi} \in \tilde{\Xi}$ we have that $z_{\text{SP}}(\hat{x}, \hat{\xi}) > 0$ and $\pi^* \in \arg \max_{\pi \in \Pi} \pi^\top (h(\hat{\xi}) - T(\hat{\xi})\hat{x})$ should be added to $\tilde{\Xi}$. When no further violated cuts are found, the optimal value of (8) coincides with that of (7). The proposed algorithm builds upon this perspective in order to further strengthen the master problem (8) without resorting to the solution of bilinear separation problems. Indeed, when a violated cut $\pi^{*\top} (h(\hat{\xi}) - T(\hat{\xi})\hat{x}) \leq 0$ is identified, one can perform a cut selection operation in order to determine if a more deeply violated cut can be found by solving the linear program $(SP(\hat{x}, \Xi, \{\pi^*\})) : \max_{\xi \in \Xi} \pi^{*\top} (h(\xi) - T(\xi)\hat{x})$, where the optimal value is necessarily strictly positive. If the optimal solution of this problem, ξ^* , is different from $\hat{\xi}$, then we will obtain a cut that is potentially deeper at \hat{x} . Further, ξ^* can now be incorporated into $\tilde{\Xi}$ (if it is not already present) in order to expand the set of realizations from which violated cuts can be generated. In the following, we call this improved version of the constraint generation algorithm the *augmented constraint generation algorithm*.

Algorithm 2: Semi-Infinite Constraint Generation (SICG) algorithm

```

1  $\tilde{\Xi} \leftarrow \emptyset$ ,  $(MP) := \min_{x \in \mathcal{X}} c^\top x$ 
2 do
3   Run Algorithm 3 (which actualizes  $(MP)$ ,  $\tilde{\Xi}$ , and  $\hat{x}$ )
4   Let  $(\xi^*, \pi^*)$  be an optimal solution of  $(SP(\hat{x}, \Xi, \Pi))$ 
5   if  $\pi^{*\top} (h(\xi^*) - T(\xi^*)\hat{x}) > 0$  then
6      $\perp$  add constraint  $\pi^{*\top} (h(\xi^*) - T(\xi^*)x) \leq 0$  to  $(MP)$ ;  $\tilde{\Xi} \leftarrow \tilde{\Xi} \cup \{\xi^*\}$ 
7 while  $\pi^{*\top} (h(\xi^*) - T(\xi^*)\hat{x}) > 0$ 
Output: An optimal solution of  $(P) : \hat{x}$ 

```

Algorithm 3: Augmented constraint generation algorithm

```

1 do
2    $feasible \leftarrow true$ 
3   Let  $\hat{x}$  be an optimal solution of  $(MP)$ 
4   for  $\hat{\xi} \in \tilde{\Xi}$  do
5     Let  $\pi^*$  be an optimal solution of  $(SP(\hat{x}, \{\hat{\xi}\}, \Pi))$  // cut generation phase
6     if  $\pi^{*\top} (h(\hat{\xi}) - T(\hat{\xi})\hat{x}) > 0$  then
7        $feasible \leftarrow false$ 
8       Let  $\xi^*$  be an optimal solution of  $(SP(\hat{x}, \Xi, \{\pi^*\}))$  // cut selection phase
9       Add  $\pi^{*\top} (h(\xi^*) - T(\xi^*)x) \leq 0$  to  $(MP)$ 
10      if  $\xi^* \notin \tilde{\Xi}$  then  $\tilde{\Xi} \leftarrow \tilde{\Xi} \cup \{\xi^*\}$  // pool expansion phase
11 while  $\neg feasible$ 

```

The overall SICG algorithm is thus composed of two decomposition algorithms, Algorithm 2 that solves (P) , and the augmented constraint generation algorithm, Algorithm 3, that solves the master problem of the latter using cutting planes. In Algorithm 2, the relaxed master problem is iteratively solved and reinforced until its solution $\hat{x} \in \mathcal{X}$ is proven feasible, similarly to CG and C&CG algorithms. The main difference lies in solving the master problem (7) using the augmented constraint generation algorithm. Further, when a violated constraint is identified through the solution of the bilinear separation problem $(SP(\hat{x}, \Xi, \Pi))$, $\tilde{\Xi}$ is expanded with ξ^* , leading implicitly to the addition of a semi-infinite constraint in Line 6. In Algorithm 3, the feasibility of the current optimal solution of (8) with respect to the already generated semi-infinite constraints in $\tilde{\Xi}$ is tested at Line 6. When a violated constraint is found, the cut selection phase is performed at Line 8, potentially exhibiting a new realization ξ^* , which

augments the pool of semi-infinite constraints associated with $\tilde{\Xi}$.

Under the assumption that the subproblems return extreme points and extreme directions of Ξ and Π , respectively, for $\hat{x} \in \mathcal{X}$, it can be shown that Algorithm 2 converges in at most $\text{ext}(\Xi)$ iterations and returns an optimal solution of (P) . Although the worst-case complexity of Algorithm 2 is the same as that of the C&CG algorithm we may expect a better numerical performance. To see why this would be the case, we first point out that both the solution of (7) and the master problem of C&CG with $\tilde{\Xi} \subseteq \Xi$ yield the same dual bound. While C&CG uses an entire second-stage system for each realization in $\tilde{\Xi}$ the proposed algorithm uses only cutting planes. Further in Line 10 of Algorithm 3 we may expand the set $\tilde{\Xi}$ which has the potential to reduce the number of times a bilinear separation problem needs to be solved. We address the effects of these improvements in our numerical section.

3 Numerical study

In this section, we test the algorithms presented in Section 2 on the robust location-transportation problem, on which the C&CG algorithm is the state-of-the-art approach [6].

In this problem, a subset of locations among a set I of potential locations should be selected (fixed cost $s_i \geq 0$ for $i \in I$), and their capacity should be chosen (variable cost $a_i \geq 0$ and a maximum capacity $c_i \geq 0$ for $i \in I$) with the aim of serving the uncertain demand of a set J of customers. This demand is modeled as $\bar{d}_j + \hat{d}_j \delta_j$, with $\bar{d}_j, \hat{d}_j \geq 0$, where uncertain parameters δ_j for $j \in J$ are governed by the uncertainty set $B_J^\Delta := \{\delta \in [0, 1]^{|J|} \mid \sum_{j \in J} \delta_j \leq \Delta\}$. We further consider that the chosen capacity of an open facility $i \in I$ can be reduced by an uncertain factor γ_i , with γ_i for $i \in I$ governed by the uncertainty set $B_I^\Gamma := \{\gamma \in [0, 1]^{|I|} \mid \sum_{i \in I} \gamma_i \leq \Gamma\}$. The available capacity at each facility $i \in I$ can then be used to satisfy the demand of customer $j \in J$ at a unit cost f_{ij} with unsatisfied demand costing $\hat{f} = \max_{(i,j) \in I \times J} f_{ij}$ per unit.

Let x_i, z_i, h_j , and y_{ij} represent the choice of opening facility $i \in I$, the capacity implemented at $i \in I$, the unsatisfied demand of customer $j \in J$, and the quantity transported from $i \in I$ to $j \in J$, respectively. The robust location-transportation problem can be formulated as:

$$\begin{aligned} \min_{\substack{x \in \{0,1\}^{|I|}, z \in \mathbb{R}_+^{|I|} \\ z_i \leq c_i x_i \quad \forall i \in I}} \sum_{i \in I} s_i x_i + a_i z_i + \max_{\substack{\gamma \in B_I^\Gamma \\ \delta \in B_J^\Delta}} \min_{y \in \mathbb{R}_+^{|I| \times |J|}, h \in \mathbb{R}_+^{|J|}} \sum_{(i,j) \in I \times J} f_{ij} y_{ij} + \sum_{j \in J} \hat{f} h_j \\ \text{s.t.} \quad \sum_{j \in J} y_{ij} \leq z_i (1 - \gamma_i) \quad \forall i \in I \\ \sum_{i \in I} y_{ij} + h_j \geq \bar{d}_j + \hat{d}_j \delta_j \quad \forall j \in J \end{aligned}$$

In our numerical study, we compare the proposed algorithm, denoted SICG hereafter, with the CG and C&CG algorithms. To do so, we randomly generate 80 instances with 30 facilities and 30 customers according to the procedure presented in [6]. We vary Γ and Δ in $\{0, 6, 6.5\}$ and $\{10, 10.5\}$, respectively. To assess the contribution of each algorithmic element, we also report the performance of versions without the pool expansion and the cut selection phases (replaced by: $\xi^* \leftarrow \hat{\xi}$), denoted by SICG-noPE, and SICG-noPECS, respectively. All algorithms are implemented in Julia 1.9 and run on Cascade Lake Intel Xeon Skylake Gold 6240 @ 2.6 GHz CPU machines on a single thread with a 5 hour time limit.

Our results, provided in Figure 1, show that C&CG performs significantly better than CG, as previously observed in the literature [6]. Further, algorithms C&CG and SICG-noPECS behave very similarly since the master problem of C&CG is equivalent to (7) for given $\tilde{\Xi} \subseteq \Xi$. The cut selection phase proves its efficiency by reducing the number of constraints and uncertainty realizations required for convergence but does not significantly decrease the number of times the bilinear separation problem is solved. This subproblem is clearly the bottleneck of all algorithms, as it constitutes a significant percentage of their total execution time. It follows that decreasing the number of times it is solved can have a considerable impact on the overall performance, which is reflected by the superior performance of the complete SICG algorithm.

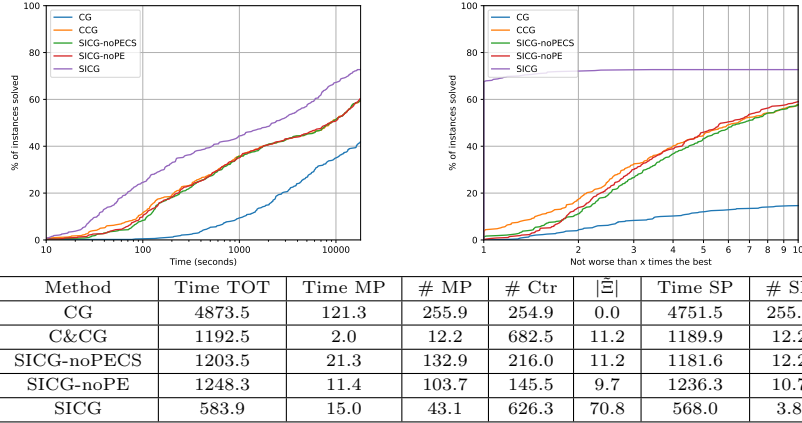


FIG. 1: On the top, performance profiles for all instances. On the bottom, data in the table is averaged over instances solved by all methods (201/480). Time TOT is the total time. Time MP includes the master solution time (model (MP)) and, in SICG variants, the time to solve the linear separation problems. #MP is the number of times (MP) is solved. Time SP and #SP are, respectively, the time to solve bilinear separation problems and the number of times they are solved. #Ctr is the number of constraints and $|\tilde{\Xi}|$ is the number of uncertainty realizations added to the master.

Indeed, the pool expansion phase augments $\tilde{\Xi}$ without resorting to the bilinear separation problem, dividing the number of times it must be solved by 3.2 on average.

4 Conclusion

In this paper, we present a semi-infinite constraint generation algorithm to solve adjustable robust optimization problems with continuous and fixed recourse. Our numerical results demonstrate the superior numerical performance of this algorithm with respect to the state-of-the-art on a location-transportation problem. They further emphasize the numerical difficulty of the bilinear separation problem that needs to be repeatedly solved in all algorithms revealing the need for further research into the solution of such problems as well as alternative ways of certifying feasibility/optimality in the context of adjustable robust optimization.

References

- [1] Aharon Ben-Tal, Alexander Goryashko, Elana Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical programming*, 99(2):351–376, 2004.
- [2] Aharon Ben-Tal and Arkadi Nemirovski. Robust convex optimization. *Mathematics of operations research*, 23(4):769–805, 1998.
- [3] Dimitris Bertsimas, Eugene Litvinov, Xu Andy Sun, Jinye Zhao, and Tongxin Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE transactions on power systems*, 28(1):52–63, 2012.
- [4] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.
- [5] Angelos Georghiou, Angelos Tsoukalas, and Wolfram Wiesemann. A primal–dual lifting scheme for two-stage robust optimization. *Operations Research*, 68(2):572–590, 2020.
- [6] Bo Zeng and Long Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.