Operations Research Letters 41 (2013) 457–461

Contents lists available at ScienceDirect

Operations Research Letters

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

Solving two-stage robust optimization problems using a column-and-constraint generation method

Bo Zeng*, Long Zhao

Department of Industrial and Management Systems Engineering, University of South Florida, United States

ARTICLE INFO

Article history:
Received 27 August 2012
Received in revised form
5 April 2013
Accepted 11 May 2013
Available online 3 June 2013

Keywords:
Two-stage robust optimization
Cutting plane algorithm
Location-and-transportation

ABSTRACT

In this paper, we present a column-and-constraint generation algorithm to solve two-stage robust optimization problems. Compared with existing Benders-style cutting plane methods, the column-and-constraint generation algorithm is a general procedure with a unified approach to deal with optimality and feasibility. A computational study on a two-stage robust location-transportation problem shows that it performs an order of magnitude faster.

1. Introduction

Robust optimization (RO) [4–6,12,9,10] is a recent optimization approach to deal with data uncertainty. Because it is derived to hedge against any perturbation in the input data, a solution to a (single-stage) RO model tends to be overly conservative. To address this issue, *two-stage RO* (and the more general *multi-stage RO*), also known as robust adjustable or adaptable optimization, has been introduced and studied [3], where the second-stage problem is to model decision making after the first-stage decisions are made and the uncertainty is revealed. Due to the improved modeling capability, two-stage RO has become a popular decision making tool. Applications include network/transportation problems [1,16,13], portfolio optimization [17], and power system scheduling problems [21,15,8].

However, two-stage RO models are very difficult to compute. As shown in [3], even a simple two-stage RO problem could be NP-hard. To overcome the computational burden, two solution strategies have been studied. The first is the use of approximation algorithms, which assume that second-stage decisions are simple functions, such as affine functions, of the uncertainty; see examples in [7]. The second type of algorithms seeks to derive exact solutions in the line of Benders' decomposition method, i.e. they gradually construct the value function of the first-stage decisions using dual solutions of the second-stage decision problems [19,21,8,15,13]. So, we call them *Benders-dual cutting plane algorithms*.

In [21], we implement a different cutting plane strategy to solve a power system scheduling problem with an uncertain wind supply. This strategy does not create constraints using dual solutions of the second-stage decision problem; instead, it dynamically generates constraints with recourse decision variables in the primal space for an identified scenario, which is very different from the philosophy behind Benders-dual procedures. For this reason, it was denoted as a *primal cut* algorithm in [21], but actually it is a column-and-constraint generation procedure. In this study, we develop and present this solution procedure in a general setting and benchmark with a Benders-dual cutting plane procedure.

In the column-and-constraint generation procedure, the generated variables and constraints are very similar to those in a two-stage stochastic programming model. Also, when the uncertainty set is discrete and finite, by enumerating variables and constraints for each scenario in the set, an equivalent monolithic optimization formulation can be constructed [17]. However, to the best of our knowledge, except for the work in [21], no algorithm has been reported that uses these variables and constraints within a cutting plane procedure to solve two-stage RO problems. This is the first presentation of this cutting plane algorithm in a general setup and the first theoretical and systematic comparison of its performance with the Benders-dual cutting plane method.

2. Two-stage RO and Benders-dual cutting plane method

Although this solution strategy can be easily extended to nonlinear formulations, we focus on linear formulations in this paper, where both the first- and second-stage decision problems are linear optimization models and the uncertainty is either a finite discrete set or a polyhedron. Let $\mathbf{y}$ be the first-stage and $\mathbf{x}$ be the

* Corresponding author. Tel.: +1 813 974 5588.
E-mail addresses: bzeng@usf.edu (B. Zeng), longzhao@mail.usf.edu (L. Zhao).

0167-6377/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.orl.2013.05.003

second-stage decision variables, respectively. Unless mentioned explicitly, they can take either discrete or continuous values. The uncertainty set $\mathcal{U}$ could be a discrete set or a polyhedron. The general form of two-stage RO formulation is

$$\min_{\mathbf{y}} \mathbf{c}^T\mathbf{y} + \max_{u\in\mathcal{U}} \min_{\mathbf{x}\in F(\mathbf{y},u)} \mathbf{b}^T\mathbf{x} \tag{1}$$
$$\text{s.t. } \mathbf{Ay} \geq \mathbf{d}, \quad \mathbf{y} \in \mathbf{S_y}$$

where $F(\mathbf{y}, u) = \{\mathbf{x} \in \mathbf{S_x} : \mathbf{Gx} \geq \mathbf{h} - \mathbf{Ey} - \mathbf{M}u\}$ with $\mathbf{S_y} \subseteq \mathbb{R}^n_+$ and $\mathbf{S_x} \subseteq \mathbb{R}^m_+$. A few cutting plane based methods have been developed and implemented to derive the exact solution when $\mathbf{S_x} = \mathbb{R}^m_+$ [19,21,8,15]. Because they are designed in the line of Benders' decomposition [2,14] and make use of the dual information of the second-stage decision problem, we call them *Benders-dual cutting plane methods* or *Benders-dual methods* for short. We briefly describe them as follows.

Consider the case where the second-stage decision problem is a linear programming (LP) problem in $\mathbf{x}$. We first take *the relatively complete recourse* assumption that this LP is feasible for any given $\mathbf{y}$ and u. Let π be its dual variables. Then, we obtain its dual problem, which is a maximization problem and can be merged with the maximization over u. As a result, we have the following problem, which yields the subproblem in the Benders-dual method.

$$\mathbf{SP_1} : \mathcal{Q}(\mathbf{y}) = \max_{u,\pi}\{(\mathbf{h} - \mathbf{Ey} - \mathbf{M}u)^T\pi : \mathbf{G}^T\pi \leq \mathbf{b},$$
$$u \in \mathcal{U}, \pi \geq \mathbf{0}\}. \tag{2}$$

Note that the resulting problem in (2) is a bilinear optimization problem. Several solution strategies have been developed, either in a heuristic fashion [8] or for instances with specially-structured $\mathcal{U}$ [19,21,15,13]. Assume that, for given $\mathbf{y}^*_k$, an optimal solution (u^*_k, π^*_k) that solves $\mathcal{Q}(\mathbf{y}^*_k)$ can be obtained by a solution oracle. Then, a cutting plane in the form of

$$\eta \geq (\mathbf{h} - \mathbf{Ey} - \mathbf{M}u^*_k)^T\pi^*_k \tag{3}$$

can be generated. It can be included into the master problem, i.e.,

$$\mathbf{MP_1} : \min_{\mathbf{y},\eta} \mathbf{c}^T\mathbf{y} + \eta$$
$$\text{s.t. } \mathbf{Ay} \geq \mathbf{d} \tag{4}$$
$$\eta \geq (\mathbf{h} - \mathbf{Ey} - \mathbf{M}u^*_l)^T\pi^*_l, \quad \forall l \leq k$$
$$\mathbf{y} \in \mathbf{S_y}, \qquad \eta \in \mathbb{R},$$

which can compute an optimal solution $(\mathbf{y}^*_{k+1}, \eta^*_{k+1})$. Note that $\mathbf{c}^T\mathbf{y}^*_k + \mathcal{Q}(\mathbf{y}^*_k)$ provides an upper bound and $\mathbf{c}^T\mathbf{y}^*_{k+1} + \eta^*_{k+1}$ provides a lower bound to the optimal value of (1). Therefore, by iteratively introducing cutting planes (3) and computing $\mathbf{MP_1}$, lower and upper bounds will converge and an optimal solution of (1) can be obtained (see A1 in the Electronic Companion [11] for details). Note that π^*_k and u^*_k are extreme points (or discrete points) of their respective feasible sets. We have the following result regarding the algorithm's complexity.

Proposition 1. *Let p be the number of extreme points of $\mathcal{U}$ if it is a polyhedron or the cardinality of $\mathcal{U}$ if it is a discrete set. Let q be the number of extreme points of $\{\pi : \mathbf{G}^T\pi \leq \mathbf{b}, \pi \geq \mathbf{0}\}$. Then, the Benders-dual algorithm will generate an optimal solution to (1) in $O(pq)$ iterations.*

Compared with classical Benders' decomposition procedures [2,14], the generated cut in (3) can be treated as an *optimality cut*. In the cases where the relatively complete recourse assumption does not hold, Terry [18] and Jiang et al. [15] discuss the *feasibility cut* issue.

3. A column-and-constraint generation algorithm

In this section, we present another cutting plane procedure to solve two-stage RO problems. Because the generated cutting planes are defined by a set of created recourse decision variables in the forms of constraints of the recourse problem, the whole procedure is a column-and-constraint generation (C&CG) procedure. To make our exposition simple, we first mention an observation when $\mathcal{U}$ is a finite discrete set. Let $\mathcal{U} = \{u_1, \ldots, u_r\}$ and $\{\mathbf{x}^1, \ldots, \mathbf{x}^r\}$ be the corresponding recourse decision variables. Then, the two-stage RO in (1) in Section 2 can be reformulated as the following:

$$\min_{\mathbf{y}} \mathbf{c}^T\mathbf{y} + \eta \tag{5}$$
$$\text{s.t. } \mathbf{Ay} \geq \mathbf{d} \tag{6}$$
$$\eta \geq \mathbf{b}^T\mathbf{x}^l, \quad l = 1, \ldots, r \tag{7}$$
$$\mathbf{Ey} + \mathbf{Gx}^l \geq \mathbf{h} - \mathbf{M}u_l, \quad l = 1, \ldots, r \tag{8}$$
$$\mathbf{y} \in \mathbf{S_y}, \qquad \mathbf{x}^l \in \mathbf{S_x}, \quad l = 1, \ldots, r. \tag{9}$$

As a result, solving a two-stage RO problem reduces to solve an equivalent (probably large-scale) mixed integer program. When the uncertainty set is very large or is a polyhedron, developing the equivalent formulation by enumerating all the possible uncertain scenarios in $\mathcal{U}$ and deriving its optimal value is not practically feasible. Nevertheless, based on constraints in (7), it is straightforward that a formulation based on a partial enumeration, i.e., a formulation defined over a subset of $\mathcal{U}$, provides a valid relaxation (and, consequently, a lower bound) to the original two-stage RO (or its equivalent formulation). Hence, by expanding a partial enumeration by adding non-trivial scenarios gradually, stronger lower bounds can be expected. With this observation in mind, we were motivated to design a column-and-constraint generation procedure that expands a subset of $\mathcal{U}$ by identifying and including significant scenarios, i.e., generating the corresponding recourse decision variables and (7)–(8) on the fly.

Similar to the Benders-dual method, this column-and-constraint generation procedure is implemented in a master-subproblem framework. We assume that an oracle can solve the following subproblem in the procedure. It can either derive an optimal solution $(u^*, \mathbf{x}^*)$ with a finite optimal value $\mathcal{Q}(\mathbf{y})$ or identify some $u^* \in \mathcal{U}$ for which the second-stage decision problem is infeasible. $\mathcal{Q}(\mathbf{y})$ in the latter case is set to $+\infty$ by convention.

$$\mathbf{SP_2} : \mathcal{Q}(\mathbf{y}) = \{\max_{u\in\mathcal{U}} \min_{\mathbf{x}} \mathbf{b}^T\mathbf{x} : \mathbf{Gx} \geq \mathbf{h} - \mathbf{Ey} - \mathbf{M}u, \mathbf{x} \in \mathbf{S_x}\}. \tag{10}$$

Column-and-constraint generation (C&CG) *algorithm*

1. Set $LB = -\infty$, $UB = +\infty$, $k = 0$ and $\mathbf{O} = \emptyset$.
2. Solve the following master problem.

$$\mathbf{MP_2} : \min_{\mathbf{y},\eta} \mathbf{c}^T\mathbf{y} + \eta$$
$$\text{s.t. } \mathbf{Ay} \geq \mathbf{d}$$
$$\eta \geq \mathbf{b}^T\mathbf{x}^l, \quad \forall l \in \mathbf{O} \tag{11}$$
$$\mathbf{Ey} + \mathbf{Gx}^l \geq \mathbf{h} - \mathbf{M}u^*_l, \quad \forall l \leq k$$
$$\mathbf{y} \in \mathbf{S_y}, \qquad \eta \in \mathbb{R}, \qquad \mathbf{x}^l \in \mathbf{S_x} \quad \forall l \leq k.$$

   Derive an optimal solution $(\mathbf{y}^*_{k+1}, \eta^*_{k+1}, \mathbf{x}^{1*}, \ldots, \mathbf{x}^{k*})$ and update $LB = \mathbf{c}^T\mathbf{y}^*_{k+1} + \eta^*_{k+1}$.
3. Call the oracle to solve subproblem $\mathbf{SP_2}$ in (10) and update $UB = \min\{UB, \mathbf{c}^T\mathbf{y}^*_{k+1} + \mathcal{Q}(\mathbf{y}^*_{k+1})\}$.
4. If $UB - LB \leq \epsilon$, return $\mathbf{y}^*_{k+1}$ and terminate. Otherwise, do
   (a) if $\mathcal{Q}(\mathbf{y}^*_{k+1}) < +\infty$, create variables $\mathbf{x}^{k+1}$ and add the following constraints

$$\eta \geq \mathbf{b}^T\mathbf{x}^{k+1} \tag{12}$$
$$\boldsymbol{Ey} + \boldsymbol{Gx}^{k+1} \geq \mathbf{h} - \mathbf{M}u^*_{k+1} \tag{13}$$

   to $\mathbf{MP_2}$ where u^*_{k+1} is the optimal scenario solving $\mathcal{Q}(\mathbf{y}^*_{k+1})$. Update $k = k + 1$, $\mathbf{O} = \mathbf{O} \cup \{k + 1\}$ and go to Step 2.

(b) if $\mathcal{Q}(\mathbf{y}^*_{k+1}) = +\infty$, create variables $\mathbf{x}^{k+1}$ and add the following constraints

$$\boldsymbol{Ey} + \boldsymbol{Gx}^{k+1} \geq \mathbf{h} - \mathbf{M}u^*_{k+1} \tag{14}$$

to $\mathbf{MP_2}$ where u^*_{k+1} is the identified scenario for which $\mathcal{Q}(\mathbf{y}^*_{k+1}) = +\infty$. Update $k = k + 1$ and go to Step 2. □

Note that constraints (12)–(13) generated in Step 4(a) serve as *optimality cuts* and constraints (14) generated in Step 4(b) serve as *feasibility cuts*. In fact, because constraint (12) with $\mathbf{x}^{k+1}$ for an infeasible scenario is also valid, we can simply generate both (12) and (13) for any identified scenario. Therefore, it yields a unified approach to deal with optimality and feasibility. Next, if the second-stage problem is LP and the relatively complete recourse assumption holds, this algorithm terminates in a finite number of iterations (see A2 in the Electronic Companion [11] for the proof).

Proposition 2. *Let p be the number of extreme points of $\mathcal{U}$ if it is a polyhedron or the cardinality of $\mathcal{U}$ if it is a finite discrete set. Then, the C&CG algorithm will converge to the optimal value of (1) in $O(p)$ iterations.*

We note some significant differences between Benders-dual method and the above algorithm:

(i) *Decision variables in the master problem.* The C&CG algorithm increases the dimensionality of the solution space by introducing a set of new variables in each iteration, while the Benders-dual algorithm keeps working with the same set of variables.
(ii) *Feasibility cut.* The C&CG algorithm provides a general approach to deal with the feasibility issue, while current approaches for the Benders-dual algorithm are problem-specific.
(iii) *Computational complexities.* Compared with the Benders-dual algorithm, the C&CG algorithm solves the master program with a larger number of variables and constraints. However, under the relatively complete recourse assumption, according to Propositions 1 and 2, the number of iterations in the C&CG algorithm is reduced by the order of $O(q)$ if the second-stage decision problem is an LP. Actually, as the number of extreme points is exponential with respect to numbers of variables and constraints (in the second stage), such a reduction is very significant. The computational study presented in [21] and in Section 4 confirms this point.
(iv) *Solution capability.* Different from the Benders-dual algorithm, which requires the second-stage problem to be an LP problem, the C&CG algorithm is indifferent to the variable types in the second stage. We recently extended this algorithm in a nested fashion to deal with two-stage RO with a mixed integer recourse problem [20].
(v) *Strength of the cut.* Under the relatively complete recourse assumption, the following proposition (see A3 in the Electronic Companion [11] for the proof) shows that the optimal value of $\mathbf{MP_1}$ is an underestimation of that of $\mathbf{MP_2}$.

Proposition 3. *For the same set of scenarios $u^*_1, u^*_2, \ldots, u^*_k$ that are considered in both of the master problems, the objective function of* $\mathbf{MP_1}$ *is an underestimation of that of* $\mathbf{MP_2}$.

Next, we present a method to deal with general polyhedral uncertainty sets. Several solution methods are developed for both relatively simple cardinality uncertainty sets and structured polyhedral uncertainty sets, including an outer approximation algorithm [8] and mixed integer linear reformulations [19,21,15,13]. The first is a heuristic procedure to solve $\mathbf{SP_1}$ with a general polyhedral uncertainty set. The latter group uses the special structure of the uncertainty set to convert the bilinear program $\mathbf{SP_1}$ into an equivalent mixed integer linear program. Nevertheless, it remains a challenging problem to exactly solve two-stage RO with a general polyhedral uncertainty set. To address this issue, we make use of the classical Karush–Kuhn–Tucker (KKT) conditions to handle a general polyhedral uncertainty set, provided that the relatively complete recourse assumption holds.

Consider $\mathbf{SP_2}$. Let π be the vector of dual variables to the second-stage decision problem. Using KKT conditions, $\mathbf{SP_2}$ is equivalent to the following:

$$\max \mathbf{b}^T\mathbf{x} \tag{15}$$

$$\text{s.t. } \mathbf{Gx} \geq \mathbf{h} - \mathbf{Ey} - \mathbf{M}u \tag{16}$$

$$\mathbf{G}^T\pi \leq \mathbf{b} \tag{17}$$

$$(\mathbf{Gx} - \mathbf{h} + \mathbf{Ey} + \mathbf{M}u)_i\pi_i = 0, \quad \forall i \tag{18}$$

$$(\mathbf{b} - \mathbf{G}^T\pi)_j x_j = 0, \quad \forall j \tag{19}$$

$$u \in \mathcal{U}, \qquad \mathbf{x} \in \mathbf{S_x}, \qquad \pi \geq \mathbf{0}. \tag{20}$$

Constraints in (18) and (19) are complementary slackness conditions, where i and j are appropriate indices for variables or constraints. By making use of the *big-M* method, they can be linearized by introducing binary variables. For example, we introduce a binary variable v_j for a constraint in (19). Then, it can be reformulated as

$$x_j \leq Mv_j, \qquad (\mathbf{b} - \mathbf{G}^T\pi)_j \leq M(1 - v_j), \quad v_j \in \{0, 1\}. \tag{21}$$

So, $\mathbf{SP_2}$ can be converted into a 0–1 mixed integer program and computed by an existing solver. We recognize that if a tight bound on big-M can be analytically obtained, e.g., the study in [13] on the robust location-transportation problem, a better performance can be achieved.

4. Case study: robust location-transportation problem

4.1. Two-stage robust location-transportation problem

Consider the following location-transportation problem. To supply a commodity to customers, it will be first stored at m potential facilities and then be transported to n customers. The fixed cost of the building facilities at site i is f_i and the unit capacity cost is a_i for $i = 1, \ldots, m$. The demand is d_j for $j = 1, \ldots, n$, and the unit transportation cost between i and j is c_{ij} for $i - j$ pair. The maximal allowable capacity of the facility at site i is K_i and $\sum_i K_i \geq \sum_j d_j$ ensures feasibility. Let $y_i \in \{0, 1\}$ be the facility location variable, $z_i \in \mathbb{R}_+$ be the capacity variable, and $x_{ij} \in \mathbb{R}_+$ be the transportation variable. Then, the nominal formulation of this location-transportation problem is as follows:

$$\min_{\mathbf{y},\mathbf{z},\mathbf{x}} \sum_i f_i y_i + \sum_i a_i z_i + \sum_i \sum_j c_{ij} x_{ij} \tag{22}$$

$$\text{s.t. } z_i \leq K_i y_i, \quad \forall i \tag{23}$$

$$\sum_j x_{ij} \leq z_i, \quad \forall i \tag{24}$$

$$\sum_i x_{ij} \geq d_j, \quad \forall j \tag{25}$$

$$y_i \in \{0, 1\}, \qquad z_i \geq 0 \quad \forall i, \quad x_{ij} \geq 0 \, \forall i, j. \tag{26}$$

The objective function in (22) is to minimize the overall cost, including the fixed cost, capacity cost, and transportation cost. Constraints in (23) and (24) require that capacity can be installed only at a site with a built facility, and the supply cannot exceed the capacity. Constraints in (25) guarantee that the demand is satisfied.

In practice, the demand is unknown before any facility is built and capacity is installed. A popular way to capture that uncertainty is as follows [7,15,1]:

$$\mathbf{D} = \left\{ \mathbf{d} : d_j = \underline{d}_j + g_j\tilde{d}_j,\ g_j \in [0, 1], \right.$$
$$\left. \sum_j g_j \le \Gamma,\ j = 1, \ldots, n \right\} \tag{27}$$

where $\underline{d}_j$ is the basic demand, $\tilde{d}_j$ is the maximal deviation, and Γ, a predefined integer value, is introduced to define the constraint of *budget uncertainty* to control the conservative level. Note that more complicated constraints, which may lead to a general polyhedron, could be used by the decision maker to describe more general uncertainty sets. With the uncertainty set on the demand, to minimize the total cost in the worst situation, a two-stage robust counterpart of the nominal formulation can be obtained as follows.

$$\min_{(\mathbf{y},\mathbf{z})\in \mathbf{S_y}} \sum_i f_i y_i + \sum_i a_i z_i + \max_{\mathbf{d}\in \mathbf{D}} \min_{\mathbf{x}\in \mathbf{S_x}} \sum_i \sum_j c_{ij} x_{ij}$$
$$\text{s.t. } \mathbf{S_y} = \{(\mathbf{y}, \mathbf{z}) \in \{0, 1\}^m \times \mathbb{R}^m_+ : (23)\}$$
$$\mathbf{S_x} = \{\mathbf{x} \in \mathbb{R}^{m\times n}_+ : (24)\text{–}(25)\}$$

where facilities and capacities are determined and established in the first stage and transportation will be determined in the second stage to meet customer demands. Similar to the nominal model, we assume $\sum_i K_i \ge \max\{\sum_j d_j : \mathbf{d} \in \mathbf{D}\}$ to ensure the existence of feasible solutions.

4.2. Experimental results and discussion

Next, we employed both C&CG and Benders-dual methods to study this two-stage robust problem. The detailed formulations of master and sub-problems are omitted here but provided in A4 in the Electronic Companion [11]. In all of our experiments, CPLEX 12.4 was used as the solver to the master problem and the oracle to the linearized subproblem. For both the master problem and subproblems, the optimality tolerance was set to 10^{-4}. Both the C&CG and Benders-dual algorithms were implemented in C++ on a desktop Dell OPTIPLEX 760 (Intel Core 2 Duo CPU, 3.0 GHz, 3.25 GB of RAM) in a Windows 7 environment.

We first study dynamic behaviors of the C&CG and Benders-dual methods on a small scale. An illustrative problem is given with three potential facilities, three customers, and a general polyhedral uncertainty set. The deterministic formulation is presented as follows:

$$\begin{aligned} \min\ & 400y_0 + 414y_1 + 326y_2 + 18z_0 + 25z_1 + 20z_2 \\ & + 22x_{00} + 33x_{01} + 24x_{02} + 33x_{10} + 23x_{11} + 30x_{12} \\ & + 20x_{20} + 25x_{21} + 27x_{22} \\ \text{s.t. } & z_i \le 800y_i, \quad \forall i = 0, 1, 2; \\ & \sum_j x_{ij} \le z_i, \quad \forall i = 0, 1, 2; \\ & \sum_i x_{ij} \ge d_j, \quad \forall j = 0, 1, 2 \\ & y_i \in \{0, 1\}; \qquad z_i \ge 0 \quad \forall i = 0, 1, 2; \\ & x_{ij} \ge 0 \quad \forall i = 0, 1, 2;\ j = 0, 1, 2. \end{aligned}$$

The uncertainty set is defined as follows:

$$\begin{aligned} \mathbf{D} = \{\mathbf{d} : \ & d_0 = 206 + 40g_0, d_1 = 274 + 40g_1, d_2 = 220 + 40g_2, \\ & 0 \le g_0 \le 1, 0 \le g_1 \le 1, 0 \le g_2 \le 1, \\ & g_0 + g_1 + g_2 \le 1.8, g_0 + g_1 \le 1.2\}. \end{aligned}$$

The upper and lower bounds of the two algorithms are presented in Table 1, which clearly shows the superiority of the C&CG method.

Table 1
Algorithm performance comparison.

Iteration	C&CG LB	C&CG UB	BD LB	BD UB
1	14 296	35 238	14 296	35 238
2	33 680	33 680	30 532	34 556
3			31 335.4	34 556
4			31 520.9	34 465.3
5			32 219.8	34 465.3
6			33 126.9	33 680
7			33 598.1	33 680
8			33 680	33 680

We also performed a systematic study on a large set of random instances to observe their general performance. To provide a basis for an apples-to-apples comparison, instances were randomly generated in a fashion used in [13] with an uncertainty set defined in (27). The demand $\underline{d}_j$ was obtained from [10, 500], the deviation $\tilde{d}_j$ was $\alpha\underline{d}_j$ with $\alpha \in [0.1, 0.5]$, the maximal allowable capacity K_i was drawn from [200, 700] with the feasibility guarantee, the fixed cost was generated from [100, 1000], the unit capacity cost was selected from [10, 100], and the transportation cost was in interval [1, 1000]. With the aforementioned setup, 20 instances were randomly generated, with 10 for the case $m \times n = 30 \times 30$ and 10 for the case $m \times n = 70 \times 70$. Also, to investigate the impact of Γ, we set its value to 10%, 20%, ..., 100% of m. So, overall, we had two sets of 100 testing problems. We used reformulation SP_1 in Section 2 to solve second-stage problems, and provided the detailed formulations in A4.2 in the Electronic Companion [11]. We also used the method presented in [13] to set values for M' to linearize subproblems.

We summarize numerical results for those 2×100 instances in Tables 2 and 3, where the average performance over every 10 instances under different Γ is displayed. In those tables, *Ratio* represents the ratio of the performance of the Benders-dual (BD) algorithm to that of the C&CG method. And an average ratio is the average of corresponding ratios rather than the ratio of average performances.

The results of the Benders-dual algorithm generally agree with those presented in [13]. The computational time for $\Gamma \in [20\%, 80\%]$ is typically more than that of other cases. This is different from the results presented in [1], where the computational times are negatively correlated with Γ. One explanation is that the problem is solved approximately in [1], while exact solutions are derived by the Benders-dual algorithm. For the C&CG algorithm, we first observe that it performs an order of magnitude faster than the Benders-dual algorithm in all experiments. Such an improvement is more significant when the problem size is large. Besides the reduction in the computational time, it generally can complete within a small number of iterations, very different from the Benders-dual method that may need hundreds of iterations. We believe that the performance improvement can be explained by two reasons. First, the C&CG algorithm strictly identifies another significant scenario by solving its subproblem, which drastically increases the convergence rate. To the contrary, the Benders-dual method uses many iterations to obtain the value function for a particular first-stage decision under the same scenario. Second, the C&CG algorithm produces a (large-scale) mixed integer program as its master problem, which keeps the network structure of the nominal model. So, the solver can make full use of that structure in the computation, while the generated cutting planes by the Benders-dual method prevent it from identifying and using that structure. It probably explains an observation from Tables 2–3 that there is little difference between the average computation times for these two master problems of different scales.

We also observe that, unlike computation time, the number of iterations in the C&CG algorithm is insensitive to problem sizes.

Table 2
Performance of Benders-dual and C&CG algorithms on 30 × 30 instances.

Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	Avg.
BD (CPU sec.)	22.71	24.27	25.14	23.76	22.98	24.55	25.29	25.61	25.07	22.49	24.19
C&CG (CPU sec.)	1.35	2.59	3.12	2.54	1.85	2.51	1.90	2.17	1.39	0.38	1.98
Ratio	**16.82**	**9.37**	**8.06**	**9.35**	**12.42**	**9.78**	**13.31**	**11.80**	**18.04**	**59.18**	**16.81**
BD (# iter.)	65.4	59.4	56.8	50	47.6	45.6	45.8	43.7	43.3	42.1	49.97
C&CG (# iter.)	4.2	5.8	6.5	5.3	5.1	5.7	4.6	5.4	4	2	4.86
Ratio	**15.57**	**10.24**	**8.74**	**9.43**	**9.33**	**8.00**	**9.96**	**8.09**	**10.83**	**21.05**	**11.12**
BD Master (sec./iter.)	0.14	0.13	0.12	0.11	0.11	0.11	0.11	0.11	0.11	0.10	0.12
C&CG master (sec./iter.)	0.12	0.15	0.16	0.14	0.12	0.14	0.12	0.13	0.11	0.07	0.13
Ratio	**1.17**	**0.87**	**0.75**	**0.79**	**0.92**	**0.79**	**0.92**	**0.85**	**1.00**	**1.43**	**0.95**

Table 3
Performance of Benders-dual and C&CG algorithms on 70 × 70 instances.

Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%	Avg.
BD (CPU sec.)	776.42	1580.71	1367.34	1300.44	1002.96	935.42	672.68	735.81	619.7	466.68	945.82
C&CG (CPU sec.)	26.16	21.27	72.3	65.22	37.88	54.62	16.72	17.64	9.66	1.55	32.3
Ratio	**29.68**	**74.32**	**18.91**	**19.94**	**26.48**	**17.13**	**40.23**	**41.71**	**64.15**	**301.08**	**63.36**
BD (# iter.)	203.9	152.1	117.5	127.1	137.4	143.6	126.3	134.2	136.6	132.4	141.11
C&CG (# iter.)	6.8	5	4.9	5	5.2	5.9	4.5	5.1	4.9	2	4.93
Ratio	**29.99**	**30.42**	**23.98**	**25.42**	**26.42**	**24.34**	**28.07**	**26.31**	**27.88**	**66.20**	**30.90**
BD Master (sec./iter.)	1.13	0.79	0.57	0.56	0.46	0.41	0.34	0.35	0.33	0.3	0.52
C&CG Master (sec./iter.)	1.45	0.58	0.57	0.58	0.55	0.72	0.47	0.5	0.51	0.12	0.61
Ratio	**0.78**	**1.36**	**1.00**	**0.97**	**0.84**	**0.57**	**0.72**	**0.70**	**0.65**	**2.50**	**1.01**

A similar result is also found in solving robust power system scheduling problems [21]. Those results indicate that the number of significant scenarios defining the worst case cost is relatively stable and small, regardless of the problem size. So, a method to quickly identify the significant scenarios, along with an efficient algorithm for the resulting master problem, can greatly improve the solution capability on two-stage RO problems.

References

[1] A. Atamturk, M. Zhang, Two-stage robust network flow and design under demand uncertainty, Operations Research 55 (4) (2007) 662–673.
[2] J.F. Benders, Partitioning procedures for solving mixed-variables programming problems, Numerische Mathematik 4 (1) (1962) 238–252.
[3] A. Ben-Tal, A. Goryashko, E. Guslitzer, A. Nemirovski, Adjustable robust solutions of uncertain linear programs, Mathematical Programming 99 (2) (2004) 351–376.
[4] A. Ben-Tal, A. Nemirovski, Robust convex optimization, Mathematics of Operations Research 23 (4) (1998) 769–805.
[5] A. Ben-Tal, A. Nemirovski, Robust solutions of uncertain linear programs, Operations Research Letters 25 (1) (1999) 1–14.
[6] A. Ben-Tal, A. Nemirovski, Robust solutions of linear programming problems contaminated with uncertain data, Mathematical Programming 88 (3) (2000) 411–424.
[7] D. Bertsimas, D.B. Brown, C. Caramanis, Theory and applications of robust optimization, SIAM Review 53 (3) (2011) 464–501.
[8] D. Bertsimas, E. Litvinov, X.A. Sun, Jinye Zhao, Tongxin Zheng, Adaptive robust optimization for the security constrained unit commitment problem, IEEE Transactions on Power Systems 28 (1) (2013) 52–63.
[9] D. Bertsimas, M. Sim, Robust discrete optimization and network flows, Mathematical Programming 98 (1) (2003) 49–71.
[10] D. Bertsimas, M. Sim, The price of robustness, Operations Research 52 (1) (2004) 35–53.
[11] Electronic companion—solving two-stage robust optimization problems using a column-and-constraint generation method. http://imse.eng.usf.edu/faculty/bzeng/MOChA_group/Index.htm.
[12] L. El Ghaoui, F. Oustry, H. Lebret, Robust solutions to uncertain semidefinite programs, SIAM Journal on Optimization 9 (1998) 33–52.
[13] V. Gabrel, M. Lacroix, C. Murat, N. Remli, Robust location transportation problems under uncertain demands, Discrete Applied Mathematics (2013) in press. Available online.
[14] A.M. Geoffrion, Generalized benders decomposition, Journal of Optimization Theory and Applications 10 (4) (1972) 237–260.
[15] R. Jiang, M. Zhang, G. Li, Y. Guan, Benders decomposition for the two-stage security constrained robust unit commitment problem, Technical Report, University of Florida, 2011. Available in Optimization-Online.
[16] F. Ordonez, J. Zhao, Robust capacity expansion of network flows, Networks 50 (2) (2007) 136–145.
[17] A. Takeda, S. Taguchi, R.H. Tutuncu, Adjustable robust optimization models for a nonlinear two-period system, Journal of Optimization Theory and Applications 136 (2) (2008) 275–295.
[18] T.L. Terry, Robust linear optimization with recourse: solution methods and other properties. Ph.D. Thesis, University of Michigan, 2009.
[19] A. Thiele, T. Terry, M. Epelman, Robust linear optimization with recourse, Technical Report, 2010. Available in Optimization-Online.
[20] L. Zhao, B. Zeng, An exact algorithm for two-stage robust optimization with mixed integer recourse problems, Technical Report, University of South Florida, 2012. Available in Optimization-Online.
[21] Long Zhao, Bo Zeng, Robust unit commitment problem with demand response and wind energy. in: Proceedings of Power and Energy Society General Meeting, 2012 IEEE, 2012, pp. 1–8.