Solving Two-stage Robust Optimization Problems Using a Column-and-Constraint Generation Method

Bo Zeng and Long Zhao

Department of Industrial and Management Systems Engineering

University of South Florida,

Email: bzeng@usf.edu, longzhao@mail.usf.edu

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.

Key words:
two-stage robust optimization, cutting plane algorithm, location-and-transportation

1 Introduction

Robust optimization (RO) [2, 3, 4, 11, 7, 8] 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 [5], 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, 15, 12], portfolio optimization [16], and power system scheduling problems [19, 14, 10].

However, two-stage RO models are very difficult to compute. As shown in [5], even a simple two-stage RO problem could be NP-hard and become intractable. 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 presented in [9]. With this assumption, two-stage RO formulations can be generally reduced to (single-stage) RO problems. The second type of algorithms seeks to derive exact solutions using sophisticated procedures.

只是相似，还不是等于Benders分解。

These algorithms gradually construct the value function of the first-stage decisions using dual solutions of the second-stage decision problems [18, 19, 10, 14, 12]. They are actually very similar to Benders' decomposition method, in that constraints that approximate the value function are iteratively generated from a subproblem and then supplied to a master problem. We call them *Benders-dual cutting plane algorithms (BD)*.

Zhao and Zeng [19] developed a different cutting plane strategy to solve a power system scheduling problem with an uncertain wind power 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 of the uncertainty set, which is very different from the philosophy behind Benders-dual procedures. For this reason, it was denoted as a *primal cut* algorithm in [19], but actually it is a column-and-constraint generation procedure. By observing in a preliminary computational study that this algorithm is very effective in solving two-stage robust power system scheduling problems, we believed that it could be refined into a general solution 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 Section 2, we give a general formulation of a two-stage RO problem and briefly describe existing Benders-dual cutting plane procedures. In Section 3, we describe the column-and-constraint generation procedure with some discussion. In Section 4, we apply this algorithm to solve a two-stage robust location-transportation problem and demonstrate its efficiency with respect to Benders-dual cutting plane procedure. We conclude the paper with a discussion on future research directions in Section 5.

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 regular optimization formulation can be constructed [16]. However, to the best of our knowledge, except for the work in [19], no algorithm has been reported that uses these variables and constraints within a cutting plane procedure, either with a discrete or continuous uncertainty set, to solve two-stage RO problems. This is the first presentation of this cutting plane algorithm with the column-and-constraint generation strategy in a general setup and the first systematic comparison of its computational 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 second-stage decision variables, respectively. Unless mentioned explicitly, no restrictions are imposed on them so they can take either discrete or continuous values. The uncertainty set $\mathcal{U}$ is a bounded set that could be a discrete set or a polyhedron. The general form of two-stage RO formulation is

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{cy} + \max_{u \in \mathcal{U}} \min_{\mathbf{x} \in \mathcal{F}(\mathbf{y},u)} \mathbf{bx} \\ \text{s.t.} \quad & \mathbf{Ay} \geq \mathbf{d}, \ \mathbf{y} \in \mathbf{S_y} \end{aligned} \tag{1}$$

where $\mathcal{F}(\mathbf{y},u) = \{\mathbf{x} \in \mathbf{S_x} : \mathbf{Gx} \geq \mathbf{h} - \mathbf{Ey} - \mathbf{M}u\}$ with $\mathbf{S_y} \subseteq \mathbf{R}_+^n$ and $\mathbf{S_x} \subseteq \mathbf{R}_+^m$. As a two-stage RO of this form is difficult to solve [5], some studies solve it approximately by assuming that the value of $\mathbf{x}$ is a simple function of u [9]. However, when the second-stage decision problem is not of any simple structure or it could be infeasible for some u, such an approximation strategy may not be applicable. A few cutting plane based methods have been developed and implemented to derive the exact solution when $\mathbf{S_x} = \mathbf{R}_+^m$ [18, 19, 10, 14]. 即完全补偿。
Because they are designed in the line of Benders' decomposition [6, 13] 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.

$$\mathbf{SP_1}: \ \mathcal{Q}(\mathbf{y}) = \max_{u,\pi} \ \pi^{\mathbf{T}}(\mathbf{h} - \mathbf{Ey} - \mathbf{M}u)$$

$$\text{s.t.} \quad \mathbf{G}^{\mathbf{T}}\pi \leq \mathbf{b}^{\mathbf{T}} \tag{2}$$
$$u \in \mathcal{U}, \ \pi \geq \mathbf{0}$$

Note that the resulting problem in (2) is a bilinear optimization problem. To solve this challenging NP-hard problem, several solution strategies have been developed, either in a heuristic fashion [10] or for instances with specially-structured $\mathcal{U}$ [18, 19, 14, 12].

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

$$\eta \geq \pi_k^{*\mathbf{T}}(\mathbf{h} - \mathbf{E}\mathbf{y} - \mathbf{M}u_k^*) \tag{3}$$

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

只要不确定集U是有限集，那么就一定在有限步内能得到最优解。
p是有限集的极点个数，或者离散集的基数。

$$
\begin{aligned}
\mathbf{MP_1}: \ \min_{\mathbf{y},\eta} \quad & \mathbf{c}\mathbf{y} + \eta \\
\text{s.t.} \quad & \mathbf{A}\mathbf{y} \geq \mathbf{d} \\
& \eta \geq \pi_l^*(\mathbf{h} - \mathbf{E}\mathbf{y} - \mathbf{M}u_l^*), \ \forall l \leq k \\
& \mathbf{y} \in \mathbf{S_y}, \eta \in \mathbf{R},
\end{aligned} \tag{4}
$$

which can compute an optimal solution $(\mathbf{y}_{k+1}^*, \eta_{k+1}^*)$. Note that $\mathbf{c}\mathbf{y}_k^* + \mathcal{Q}(\mathbf{y}_k^*)$ provides an upper bound and $\mathbf{c}\mathbf{y}_{k+1}^* + \eta_{k+1}^*$ 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. Note that when $\mathcal{U}$ is a polyhedron, because u_{k+1}^* and π_{k+1}^* are extreme points of their respective feasible sets, only a finite number of cuts in the form of (3) will be generated. A detailed algorithm description can be found in Appendix A-1.

Because the above observation also holds when $\mathcal{U}$ is a finite discrete set, 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 $\{\pi : \mathbf{G}^{\mathbf{T}}\pi \leq \mathbf{b}^{\mathbf{T}}, \pi \geq \mathbf{0}\}$. Then, the Benders-dual cutting plane algorithm will generate an optimal solution to* (1) *in $O(pq)$ iterations.*

Compared with classical Benders' decomposition procedures [6, 13], the generated cut in (3) can be treated as a *optimality cut*. In the cases where the the relatively complete recourse assumption does not hold, Terry [17] and Jiang et al. [14] 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) can be reformulated as the following:

所谓的补偿变量，其实就是第二阶段的决策变量。

$$\min_{\mathbf{y}} \quad \mathbf{c}\mathbf{y} + \eta \tag{5}$$
$$\text{s.t.} \quad \mathbf{A}\mathbf{y} \geq \mathbf{d} \tag{6}$$
$$\eta \geq \mathbf{b}\mathbf{x}^l, \ l = 1, \ldots, r \tag{7}$$

数。本来是从目标函数中拆出来的。

$$\mathbf{E}\mathbf{y} + \mathbf{G}\mathbf{x}^l \geq \mathbf{h} - \mathbf{M}u_l, \ l = 1, \ldots, r \tag{8}$$

但这个为什么是向量就不清楚了。

如果不确定集很大或者是一个多面体，那么，枚举所有的不确定场景来得到对等式是不可行的。但是，根据（7），我们可以部分枚举，不断增加非平凡场景。这样得到的是松弛问题，其解是原问题的下限。

$$\mathbf{y} \in \mathbf{S_y},\ \mathbf{x}^l \in \mathbf{S_x},\ l = 1, \dots, r. \tag{9}$$

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) into the existing partial enumeration on the fly.

不断产生补偿变量和约束（7-8）

3.1 Algorithm Description

Similar to the Benders-dual method, this column-and-constraint generation procedure is implemented in a master-subproblem framework. The master problem is the augmenting of partial enumeration that iteratively provides tighter relaxations as well stronger lower bounds. We also assume that an oracle can solve the following max min problem, which is the subproblem in the procedure and computes the most significant scenario for the subset expansion in the next iteration.

假定已经有谕示算法可以解子问题，要么得到最优解，要么识别不可行的场景。

$$\begin{aligned} \mathbf{SP_2}: \quad & \mathcal{Q}(\mathbf{y}) = \max_{u \in \mathcal{U}} \min_{\mathbf{x}} \mathbf{bx} \\ \text{s.t.} \quad & \mathbf{Gx} \geq \mathbf{h} - \mathbf{Ey} - \mathbf{M}u \\ & \mathbf{x} \in \mathbf{S_x}. \end{aligned} \tag{10}$$

This oracle 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. Next, we describe the algorithm.

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.

$$\begin{aligned} \mathbf{MP_2}: \min_{\mathbf{y},\eta} \quad & \mathbf{cy} + \eta \\ \text{s.t.} \quad & \mathbf{Ay} \geq \mathbf{d} \\ & \eta \geq \mathbf{bx}^l,\ \forall l \in \mathbf{O} \\ & \mathbf{Ey} + \mathbf{Gx}^l \geq \mathbf{h} - \mathbf{M}u_l^*,\ \forall l \leq k \\ & \mathbf{y} \in \mathbf{S_y},\ \eta \in \mathbf{R},\ \mathbf{x}^l \in \mathbf{S_x}\ \forall l \leq k \end{aligned} \tag{11}$$

   Derive an optimal solution $(\mathbf{y}_{k+1}^*, \eta_{k+1}^*, \mathbf{x}^{1*}, \dots, \mathbf{x}^{k*})$ and update $LB = \mathbf{cy}_{k+1}^* + \eta_{k+1}^*$.
3. Call the oracle to solve subproblem $\mathbf{SP_2}$ in (10) and update $UB = \min\{UB, \mathbf{cy}_{k+1}^* + \mathcal{Q}(\mathbf{y}_{k+1}^*)\}$.
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

$$\eta \geq \mathbf{bx}^{k+1} \tag{12}$$
$$\mathbf{Ey} + \mathbf{Gx}^{k+1} \geq \mathbf{h} - \mathbf{M}u_{k+1}^* \tag{13}$$

   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.

注意，x（K+1)是变量，新加入的。而u(k+1)是常量。

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

这条约束对成本没有影响，但是对y有约束作用。

$$\mathbf{E}\mathbf{y} + \mathbf{G}\mathbf{x}^{k+1} \geq \mathbf{h} - \mathbf{M}u^*_{k+1} \tag{14}$$

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. Also, from the above description, this algorithm dynamically constructs a formulation ($\mathbf{MP_2}$) with recourse decisions for each identified scenario whose structure is very similar to that of a two-stage stochastic programming model.

对每一个辨识出来的场景都建立了补偿变量，所以与随机优化很相似！

Next, if the second-stage decision problem is LP and the relatively complete recourse assumption holds, we show that this algorithm terminates in a finite number of iterations.

相对完全补偿：对每一个第一阶段决策，第二阶段都能补偿，所以称为相对完全。

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.*

Proof. Because the proof for the case where $\mathcal{U}$ is a discrete set is similar and simpler, we provide proof for the case where $\mathcal{U}$ is a polyhedron.

We first show that the two-stage RO in (1) is equivalent to a mixed integer program that is built on all the extreme points of $\mathcal{U}$. Because of the relatively complete recourse assumption and the strong duality property, $\mathbf{SP_2}$ can be converted into $\mathbf{SP_1}$. Note that $\mathbf{SP_1}$ is a bi-linear program over two disjoint polyhedrons, it always has an optimal solution combining extreme points of these two polyhedrons, in spite of the value of $\mathbf{y}$. Let $\mathbf{U} = \{u_1, u_2, \ldots, u_p\}$ be the collection of extreme points of $\mathcal{U}$. Hence, two-stage RO in (1) is equivalent to

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{c}\mathbf{y} + \max_{u \in \mathbf{U}} \min_{\mathbf{x} \in F(\mathbf{y}, u)} \mathbf{b}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{y} \geq \mathbf{d}, \ \mathbf{y} \in \mathbf{S_y}. \end{aligned} \tag{15}$$

As a result, by enumerating values in the finite set $\mathbf{U}$, it reduces to the following mixed integer program:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \mathbf{c}\mathbf{y} + \eta \\ \text{s.t.} \quad & \mathbf{A}\mathbf{y} \geq \mathbf{d} \\ & \eta \geq \mathbf{b}\mathbf{x}^l, \ l = 1, \ldots, p \\ & \mathbf{E}\mathbf{y} + \mathbf{G}\mathbf{x}^l \geq \mathbf{h} - \mathbf{M}u_l, \ l = 1, \ldots, p \\ & \mathbf{y} \in \mathbf{S_y}, \ \mathbf{x}^l \in \mathbf{S_x}, \ l = 1, \ldots, p. \end{aligned} \tag{16}$$

Clearly, the master problem $\mathbf{MP_2}$ is a relaxation of (16), providing a lower bound.

Note that solving $\mathbf{SP_2}$ will expand $\mathbf{MP_2}$ by including one more u_l and its corresponding recourse variables and constraints. Next, we show that any repeated u^* in the procedure implies the optimality, i.e. $LB = UB$. Assume that in k^{th} iteration, $(\mathbf{y}^*, \eta^*)$ and $(u^*, \mathbf{x}^*)$ are optimal solutions of $\mathbf{MP_2}$ and $\mathbf{SP_2}$, respectively, and u^* appears in a previous iteration. One one hand, from Step 3 of the algorithm, we have $UB \leq \mathbf{c}\mathbf{y}^* + \mathbf{b}\mathbf{x}^*$. On the other hand, because u^* has been identified in a previous iteration, $\mathbf{MP_2}$ in this iteration is identical to that in $(k-1)^{\text{th}}$ iteration. Hence, $(\mathbf{y}^*, \eta^*)$ is the optimal solution of $\mathbf{MP_2}$ in the $(k-1)^{\text{th}}$ iteration. From Step 2 of the algorithm, we have $LB \geq \mathbf{c}\mathbf{y}^* + \eta^* \geq \mathbf{c}\mathbf{y}^* + \mathbf{b}\mathbf{x}^*$, where the last inequality follows from the fact that u^* is already identified and the related constraints are added to $\mathbf{MP_2}$ before or in the $(k-1)^{\text{th}}$ iteration. Consequently, we have $LB = UB$.

Then, the conclusion follows immediately from the fact that $\mathbf{U}$, i.e. the set of all extreme points of the polyhedral uncertainty set, is finite. $\square$

Comparing the Benders-dual algorithm and the aforementioned C&CG algorithm, we note some significant differences in the following aspects:

(*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*) *Numbers and structures of generated constraints.* Both algorithms generate a constraint, providing the lower bound to the worst case cost in each iteration. In the C&CG algorithm, such a constraint is defined by primal recourse variables along with a set of constraints to restrict them. In the Benders-dual algorithm, such a constraint is constructed by the first-stage primal decision variables with optimal values of the second-stage dual variables. So, the C&CG algorithm generates a set of constraints, while the Benders-dual algorithm creates only a single constraint in each iteration.

(*iii*) *Feasibility cut.* The C&CG algorithm provides a general approach to deal with the feasibility issue of the second-stage decision problem, while current approaches for the Benders-dual algorithm are problem-specific.

(*iv*) *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 on a power system scheduling problem [19] and on a location-transportation problem presented in Section 4 confirms this point.

(*v*) *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].

(*vi*) *Strength of the cut.* Under the relatively complete recourse assumption, we show in the following proposition that the optimal value of $\mathbf{MP_1}$ is an underestimation of that of $\mathbf{MP_2}$ after including corresponding cutting planes.

Proposition 3. *For the same set of scenarios $u_1^*, u_2^*, ..., u_k^*$ that are considered in both of the master problems, the objective function of* $\mathbf{MP_1}$ *is an underestimation of that of* $\mathbf{MP_2}$.

Proof. We claim that for any fixed first-stage decision $\mathbf{y}^*$, the optimal value of $\mathbf{MP_1}$, i.e., $\mathbf{c^T y}^* + \max\{\pi_l^*(\mathbf{h} - \mathbf{E y}^* - \mathbf{M}u_l^*)\}_{l=1}^k$, is smaller than or equal to that of $\mathbf{MP_2}$, i.e., $\mathbf{c^T y}^* + \max\{\min_{\mathbf{x}^l \geq 0} \ \mathbf{b^T x}^l : \mathbf{G x}^l \geq \mathbf{h} - \mathbf{M}u_l^* - \mathbf{E y}^*\}_{l=1}^k$. To this end, it is sufficient to show that for a fixed $l \in \{1, 2, ..., k\}$, we have $\min\{\mathbf{b^T x}^l : \mathbf{G x}^l \geq \mathbf{h} - \mathbf{M}u_l^* - \mathbf{E y}^*, \mathbf{x}^l \geq 0\} \geq \pi_l^*(\mathbf{h} - \mathbf{E y}^* - \mathbf{M}u_l^*)$ which, by the strong duality of an LP, is equivalent to $\max\{\pi_l^T(\mathbf{h} - \mathbf{E y}^* - \mathbf{M}u_l^*) : \mathbf{G^T}\pi_l \leq \mathbf{b^T},\ \pi_l \geq \mathbf{0}\} \geq \pi_l^*(\mathbf{h} - \mathbf{E y}^* - \mathbf{M}u_l^*)$. The last inequality holds, because π_l^* is a fixed feasible solution (in Benders-dual algorithm) to the maximization problem on the left-hand side. The conclusion follows immediately. $\square$

CCG的下限更紧。

3.2 Handling General Polyhedral Uncertainty Sets

Capturing complicated uncertain data by a general uncertainty polyhedron in the two-stage RO framework would be advantageous. Clearly, effective solution procedures are required to solve subproblems, i.e., $\mathbf{SP_1}$ or $\mathbf{SP_2}$ in both the Benders-dual and the C&CG algorithms. Several solution methods are developed for both relatively simple cardinality uncertainty sets and structured polyhedral uncertainty sets, including an outer approximation algorithm [10] and mixed integer linear reformulations [18, 19, 14, 12]. The first is a heuristic procedure that provides a local solution to $\mathbf{SP_1}$ with a general polyhedral uncertainty set. The latter group uses the special structure of the uncertainty set to convert the bilinear program into an equivalent mixed integer linear program. The conversion procedure relies on an observation that if $\mathcal{U}$ is a polyhedron of a clear structure, optimal u^*s can be identified or characterized. 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) condition 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. Note that the second-stage decision problem always has an optimal solution, and KKT conditions are both sufficient and necessary for a solution to be optimal. Hence, $\mathbf{SP_2}$ is equivalent to the following:

$$\begin{aligned}
\max \quad & 0 && (17)\\
\text{s.t.} \quad & \mathbf{Gx} \geq \mathbf{h} - \mathbf{Ey} - \mathbf{M}u && (18)\\
& \mathbf{G^T}\pi \leq \mathbf{b^T} && (19)\\
& \pi_i(\mathbf{Gx} - \mathbf{h} + \mathbf{Ey} + \mathbf{M}u)_i = 0, \ \forall i && (20)\\
& x_j(\mathbf{b^T} - \mathbf{G^T}\pi)_j = 0, \ \forall j \quad ? && (21)\\
& u \in \mathcal{U}, \mathbf{x} \in \mathbf{S_x}, \pi \geq \mathbf{0} && (22)
\end{aligned}$$

Constraints in (20) and (21) 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 (21). Then, it can be reformulated as

用大M将互补松弛条件线性化。

$$x_j \leq Mv_j, \ (\mathbf{b^T} - \mathbf{G^T}\pi)_j \leq M(1 - v_j), \ v_j \in \{0, 1\}. \tag{23}$$

By using this linearization strategy and with the fact that $\mathcal{U}$ is a polyhedra set, $\mathbf{SP_2}$ is converted into a $0-1$ mixed integer program whose solution can be computed by an existing solver. We recognize that the big-M value could bring heavy computational expenses in solving $\mathbf{SP_2}$. When the two-stage RO has a clear structure, and a tight bound on big-M can be analytically obtained, e.g., the study in [12] on the robust location-transportation problem, a better algorithm performance can be achieved.

4 Case Study: 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,

$z_i \in \mathbf{R}_+$ be the capacity variable, and $x_{ij} \in \mathbf{R}_+$ be the transportation variable. Then, the nominal formulation of this location-transportation problem is as follows:

$$\min_{\mathbf{y},\mathbf{z},\mathbf{x}} \sum_i f_i y_i + \sum_i a_i z_i + \sum_i \sum_j c_{ij} x_{ij} \tag{24}$$

$$\text{s.t. } z_i \leq K_i y_i, \forall i \tag{25}$$

$$\sum_j x_{ij} \leq z_i, \forall i \tag{26}$$

$$\sum_i x_{ij} \geq d_j, \forall j \tag{27}$$

$$y_i \in \{0,1\},\ z_i \geq 0\ \forall i,\quad x_{ij} \geq 0\ \forall i,j \tag{28}$$

The objective function in (24) is to minimize the overall cost, including the fixed cost, capacity cost, and transportation cost. Constraints in (25) and (26) require that capacity can be installed only at a site with a built facility, and the supply cannot exceed the capacity. Constraints in (27) 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 [9, 14, 1]:

$$\mathbf{D} = \{\mathbf{d} : d_j = \underline{d}_j + g_j \tilde{d}_j,\ g_j \in [0,1],\ \sum_j g_j \leq \Gamma,\ j = 1,\ldots,n\} \tag{29}$$

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, the decision process can be decomposed into two stages that are implemented before and after the realization of demands. So, in the first stage, facility and capacity are determined and established; in the second stage, transportation is determined to meet customer demands. To minimize the total cost in the worst situation, a two-stage robust counterpart of the nominal formulation can be obtained as follows. Similar to the nominal model, we assume $\sum_i K_i \geq \max\{\sum_j d_j : \mathbf{d} \in \mathbf{D}\}$ to ensure the existence of feasible solutions.

$$\begin{aligned}
\min_{(\mathbf{y},\mathbf{z})\in\mathbf{S_y}} \quad & \sum_i f_i y_i + \sum_i a_i z_i + \max_{\mathbf{d}\in\mathbf{D}} \min_{\mathbf{x}\in\mathbf{S_x}} \sum_i \sum_j c_{ij} x_{ij} \\
\text{s.t.} \quad & \mathbf{S_y} = \{(\mathbf{y},\mathbf{z}) \in \{0,1\}^m \times \mathbf{R}_+^m : (25)\} \\
& \mathbf{S_x} = \{\mathbf{x} \in \mathbf{R}_+^{m\times n} : (26-27)\}
\end{aligned}$$

The detailed formulations of master and sub-problems are omitted here but provided in Appendix A-2. In the following, we investigate the dynamic behaviors of the C&CG and Benders-dual methods on a small scale. This demonstrates the different convergence rates of the algorithms. Then, we perform a systematic study on a set of random instances to observe their general performance.

4.1 Results of a Simple Case Study

An illustrative problem is given with three potential facilities, three customers, and a general polyhedral uncertainty set. The deterministic formulation is presented as follows:

$$\begin{aligned}
\min \ & 400y_0 + 414y_1 + 326y_2 + 18z_0 + 25z_1 + 20z_2 + 22x_{00} + 33x_{01} + 24x_{02} + 33x_{10} \\
& + 23x_{11} + 30x_{12} + 20x_{20} + 25x_{21} + 27x_{22}
\end{aligned}$$

Table 1: Algorithm Performance Comparison

Iteration	C&CG LB	C&CG UB	BD LB	BD UB
1	14296	35238	14296	35238
2	33680	33680	30532	34556
3			31335.4	34556
4			31520.9	34465.3
5			32219.8	34465.3
6			33126.9	33680
7			33598.1	33680
8			33680	33680

$$
\begin{aligned}
\text{s.t. } & z_i \leq 800y_i, \forall i = 0,1,2 \\
& \sum_j x_{ij} \leq z_i, \forall i = 0,1,2 \\
& \sum_i x_{ij} \geq d_j, \forall j = 0,1,2 \\
& y_i \in \{0,1\},\ z_i \geq 0\ \forall i = 0,1,2,\quad x_{ij} \geq 0\ \forall i = 0,1,2; j = 0,1,2.
\end{aligned}
$$

The uncertainty set is defined as follows:

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

The robust counterpart and the detailed solution procedures are omitted here. The upper and lower bounds of the two algorithms are presented in Table 1, which clearly shows the superiority of the C&CG method.

4.2 Computation Results

In all of the following experiments, CPLEX 12.1 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}. We implemented both the C&CG and Benders-dual algorithms in C++ and perform computation experiments on a desktop Dell OPTIPLEX 760 (Intel Core 2 Duo CPU, 3.0GHz, 3.25GB of RAM) in a Windows 7 environment.

To provide a basis for an apples-to-apples comparison, our numerical study was performed on instances that were randomly generated in a fashion used in [12] with an uncertainty set defined in (29). 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 $10\%, 20\%, \ldots, 100\%$ of m. So, overall, we had two sets of 100 testing problems. We also used the method presented in [12] to set values for M' to linearize the subproblems in Appendix A-2.2.

All the original computation results are presented in Tables A-1-A-4 in the appendix. We summarize those results in Table 2 and Table 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 algorithm to that of the C&CG method.

The results of the Benders-dual algorithm generally agree with those presented in [12]. 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 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 its subproblem in every iteration, 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 largely 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. The latter point is also supported by the fact that the reduction in the computational time is more significant than the reduction in the number of iterations.

We also observe that, unlike computation time, the number of iterations in the C&CG algorithm is insensitive to problem sizes. A similar result is also found in solving robust power system scheduling problems [19]. 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.

5 Conclusion

In this paper, we presented a cutting plane algorithm, the C&CG algorithm, in a general setting to solve two-stage robust optimization problems. Different from existing Benders-dual cutting plane methods, it is a column-and-constraint generation method that can solve general problems, does not restrict variable types, and provides a unified approach to deal with feasibility and optimality issues. In particular, this algorithm can derive an optimal solution with much lower computational expense. From our systematical study on a two-stage robust location-transportation problem, as well as a preliminary study on power system scheduling problems in [19], we observed that it performs an order of magnitude faster than the Benders-dual cutting plane algorithm.

As we mentioned, the master problem of the C&CG algorithm has a structure similar to a two-stage stochastic program. Given that the master problem could be a large-scale mixed integer program, one possible direction is how to take advantage of existing research on two-stage stochastic programming to further improve the computational performance of the C&CG algorithm in solving two-stage robust optimization. Another direction is to design advanced algorithms for the more challenging two-stage robust optimization problem with a mixed integer recourse problem, for which preliminary research has been done by extending this column-and-constraint generation method in a nested fashion to develop an algorithm [20].

References

[1] A. Atamturk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

[2] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

[3] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–14, 1999.

[4] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, 2000.

[5] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.

[6] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962.

[7] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71, 2003.

[8] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[9] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[10] D. Bertsimas, E. Litvinov, X.A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *To appear in IEEE Transactions on Power Systems*, 2011.

[11] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM journal of optimization*, 9:33–52, 1998.

[12] V. Gabrel, M. Lacroix, C. Murat, and N. Remli. Robust location transportation problems under uncertain demands. *To appear in Discrete Applied Mathematics*, 2011.

[13] A.M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[14] R. Jiang, M. Zhang, G. Li, and Y. Guan. Benders decomposition for the two-stage security constrained robust unit commitment problem. Technical report, available in *optimization-online*, University of Florida, 2011.

[15] F. Ordonez and J. Zhao. Robust capacity expansion of network flows. *Networks*, 50 (2):136–145, 2007.

[16] A. Takeda, S. Taguchi, and RH Tutuncu. Adjustable robust optimization models for a nonlinear two-period system. *Journal of Optimization Theory and Applications*, 136 (2):275–295, 2008.

[17] T.L. Terry. *Robust Linear Optimization with Recourse: Solution Methods and Other Properties*. PhD thesis, University of Michigan, 2009.

[18] A. Thiele, T. Terry, and M. Epelman. Robust linear optimization with recourse. Technical report, available in *optimization-online*, 2010.

[19] L. Zhao and B. Zeng. Robust unit commitment problem with demand response and wind energy. Technical report, available in *optimization-online*, University of South Florida, 2010.

[20] L. Zhao and B. Zeng. An exact algorithm for two-stage robust optimization with mixed integer recourse problems. Technical report, available in *optimization-online*, University of South Florida, 2012.

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.)	16.77	18.05	21.21	17.62	18.47	19.09	19.80	21.69	22.07	19.15	19.39
C&CG (CPU sec.)	1.17	1.89	2.28	1.99	1.33	1.61	1.41	1.68	1.08	0.34	1.48
Ratio	**14.31**	**9.57**	**9.31**	**8.86**	**13.89**	**11.83**	**14.07**	**12.92**	**20.51**	**57.08**	**17.24**
BD (# iter.)	62.70	58.70	58.50	47.30	46.20	43.30	43.20	44.70	43.80	40.60	48.90
C&CG (# iter.)	4.20	5.70	6.50	5.40	5.00	5.60	4.60	5.40	4.10	2.00	4.85
Ratio	**14.93**	**10.30**	**9.00**	**8.76**	**9.24**	**7.73**	**9.39**	**8.28**	**10.68**	**20.30**	**10.86**

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.)	689.32	1581.90	1725.21	1391.96	837.04	703.60	657.46	548.40	396.61	205.31	873.68
C&CG (CPU sec.)	22.57	26.52	76.55	70.19	39.76	50.76	12.03	11.59	6.59	1.15	31.77
Ratio	**30.54**	**59.65**	**22.54**	**19.83**	**21.05**	**13.86**	**54.67**	**47.31**	**60.18**	**178.71**	**50.84**
BD (# iter.)	198.50	153.50	137.70	122.70	143.00	133.20	138.70	131.40	139.20	133.30	143.12
C&CG (# iter.)	7.00	5.30	5.40	5.10	5.20	5.80	4.50	5.00	4.60	2.00	4.99
Ratio	**28.36**	**28.96**	**25.50**	**24.06**	**27.50**	**22.97**	**30.82**	**26.28**	**30.26**	**66.65**	**31.14**

Appendix

A-1 Benders-dual Method

Benders-dual Cutting Plane Algorithm

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

$$\begin{aligned} \mathbf{MP_1}: \min_{\mathbf{y},\eta} \quad & \mathbf{cy} + \eta \\ \text{s.t.} \quad & \mathbf{Ay} \geq \mathbf{d} \\ & \eta \geq \pi_l^*(\mathbf{h} - \mathbf{Ey} - \mathbf{M}u_l^*), \ \forall l \leq k \\ & \mathbf{y} \in \mathbf{S_y}, \eta \in \mathbf{R}. \end{aligned}$$

   Derive an optimal solution $(\mathbf{y}_{k+1}^*, \eta_{k+1}^*)$ and update $LB = \mathbf{cy}_{k+1}^* + \eta_{k+1}^*$.

3. Call the oracle to solve $\mathbf{SP_1}$ in (2) in Section 2, i.e. $\mathcal{Q}(\mathbf{y}_{k+1}^*)$, and derive an optimal solution (u_{k+1}^*, π_{k+1}^*). Update $UB = \min\{UB, \mathbf{cy}_{k+1}^* + \mathcal{Q}(\mathbf{y}_{k+1}^*)\}$.
4. If $UB - LB \leq \epsilon$, return $\mathbf{y}_{k+1}^*$ and terminate. Otherwise, update $k = k+1$ and go to Step 2 with addition of the following constraint to $\mathbf{MP_1}$.

$$\eta \geq \pi_{k+1}^{*\mathbf{T}}(\mathbf{h} - \mathbf{Ey} - \mathbf{M}u_{k+1}^*)$$

□

A-2 Algorithm Specifications

A-2.1 Solving Subproblems based on KKT-condition

For a general polyhedron case, given fixed $(\mathbf{y}^*, \mathbf{z}^*, \mathbf{d}^*)$, the optimal $\mathbf{x}$ for the innermost minimization problem is any of the feasible solutions to the following system by KKT-condition.

$$\begin{aligned} & \sum_j x_{ij} \leq z_i^*, \forall i \\ & \sum_i x_{ij} \geq d_j^*, \forall j \\ & \lambda_j - \pi_i \leq c_{ij}, \forall i, j \\ & x_{ij}(c_{ij} - \lambda_j + \pi_i) = 0, \forall i, j \\ & \lambda_j(\sum_j x_{ij} - d_j^*) = 0, \forall j \\ & \pi_i(z_i^* - \sum_j x_{ij}) = 0, \forall i \end{aligned}$$

Therefore, the subproblem is equivalent to the following nonlinear programming problem with complementary constraints:

$$\begin{aligned} & \max 0 \\ & st. \ \sum_j x_{ij} \leq z_i^*, \forall i \\ & \qquad \sum_i x_{ij} \geq d_j, \forall j \end{aligned}$$

$$
\begin{aligned}
&\lambda_j - \pi_i \le c_{ij}, \forall i,j \\
&x_{ij}(c_{ij} - \lambda_j + \pi_i) = 0, \forall i,j \\
&\lambda_j(\sum_j x_{ij} - d_j) = 0, \forall j \\
&\pi_i(z_i^* - \sum_j x_{ij}) = 0, \forall i \\
&\mathbf{d} \in \mathbf{D};\ \pi, x, \lambda \ge 0
\end{aligned}
$$

By defining $\overline{d}_t = \underline{d}_t + \tilde{d}_t$, the following upper bounds of variables and constraints can be obtained:

$$
\begin{aligned}
&x_{ij} \le \min\{\overline{d}_j, z_i^*\} = M_{x_{ij}}, \forall i,j \\
&\pi_i \le \max_j(c_{ij}) = M_{\pi_i}, \forall i \\
&\lambda_j \le \max_i(c_{ij}) = M_{\lambda_j}, \forall j \\
&c_{ij} - \lambda_j + \pi_i \le c_{ij} + M_{\pi_i}, \forall i,j \\
&\sum_j x_{ij} - d_j \le \tilde{d}_j, \forall j \\
&z_i^* - \sum_j x_{ij} \le z_i^*, \forall i
\end{aligned}
$$

Finally, the subproblem can be linearized as an MIP by introducing the binary variables $\alpha_{ij}, \beta_i, \gamma_j, \forall i,j$:

$$
\begin{aligned}
\max\ & 0 \\
st.\ & \sum_j x_{ij} \le z_i^*, \forall i \\
& \sum_i x_{ij} \ge d_j, \forall j \\
& \lambda_j - \pi_i \le c_{ij}, \forall i,j \\
& x_{ij} \le M_{x_{ij}} \alpha_{ij}, \forall i,j \\
& c_{ij} - \lambda_j + \pi_i \le (c_{ij} + M_{\pi_i})(1 - \alpha_{ij}), \forall i,j \\
& \lambda_j \le M_{\lambda_j} \beta_j, \forall j \\
& \sum_i x_{ij} - d_j \le \tilde{d}_j(1 - \beta_j), \forall j \\
& \pi_i \le \gamma_i M_{\pi_i}, \forall i \\
& z_i^* - \sum_j x_{ij} \le (1 - \gamma_i) z_i^*, \forall i \\
& \mathbf{d} \in \mathbf{D}; \pi, \mathbf{x}, \lambda \ge 0; \alpha, \beta, \gamma \text{ binary}
\end{aligned}
$$

A-2.2 Solving Subproblems based on Strong Duality

For the special uncertainty set (29) in Section 4 and for the fixed $(\mathbf{y}^*, \mathbf{z}^*)$, we obtain the dual problem of the second-stage decision problem and combine it with the maximization part. The combined problem is:

$$
\max_{\mathbf{d},\lambda,\pi} \sum_j \underline{d}_j \lambda_j + \sum_j \tilde{d}_j \lambda_j g_j - \sum_i \pi_i z_i^*
$$

$$\text{st. } \lambda_j - \pi_i \le c_{ij}, \forall i, j$$
$$\sum_j g_j \le \Gamma$$
$$0 \le g_j \le 1, \ \forall j$$
$$\mathbf{d} \in \mathbf{D}; \ \lambda_j \ge 0, \ \forall j, \ \pi_i \ge 0, \ \forall i$$

It is easy to see that we always have $g_j \in \{0, 1\}$ for all j in one optimal solution. So, we can apply the standard linearization technique to obtained the following mixed integer linear program. Note that w_j is introduced to replace $\lambda_j g_j$ and M' is a big number. The same strategy is also used in [12].

$$\max_{\mathbf{d},\lambda,\pi,\mathbf{w}} \ \sum_j \underline{d}_j \lambda_j + \sum_j \tilde{d}_j w_j - \sum_i z_i^* \pi_i$$
$$\text{st. } \lambda_j - \pi_i \le c_{ij}, \forall i, j$$
$$\sum_j g_j \le \Gamma$$
$$w_j \le \lambda_j, \forall j$$
$$w_j \le M' g_j, \forall j$$
$$w_j \ge \lambda_j - M'(1 - g_j), \forall j$$
$$\mathbf{d} \in \mathbf{D}; \ \lambda_j, \ w_j \ge 0, \ g_j \in \{0, 1\} \ \forall j, \ \pi_i \ge 0, \ \forall i$$

A-2.3 Master Problems

To address the feasibility issue, constraint

$$\sum_i z_i \ge \max\{\sum_j d_j : \mathbf{d} \in \mathbf{D}\}$$

is added to both initial master problems in our implementation.

Assume that an optimal solution, $(\mathbf{d}^k, \lambda^k, \pi^k, \mathbf{w}^k)$, is derived for the subproblem in the k^{th} iteration. The master problem of the C&C G method in the k^{th} iteration will be of the following form, where $\mathbf{x}^k$ are the created recourse variables in iteration k.

$$\min_{\mathbf{y},\mathbf{z},\mathbf{x},\eta} \ \sum_i f_i y_i + \sum_i a_i z_i + \eta$$
$$\text{s.t. } \sum_i z_i \ge \max\{\sum_j d_j : \mathbf{d} \in \mathbf{D}\}$$
$$z_i \le K_i y_i, \forall i$$
$$\eta \ge \sum_i \sum_j c_{ij} x_{ij}^l \ \forall i, j, \text{and } 1 \le l \le k$$
$$\sum_i x_{ij}^l \ge d_j^l, \forall j, \text{and } 1 \le l \le k$$
$$\sum_j x_{ij}^l \le z_i, \forall i, \text{and } 1 \le l \le k$$
$$z_i \ge 0, y_i \in \{0, 1\}, \forall i, \ \eta \in \mathbf{R}, \ x_{ji}^l \ge 0, \forall i, j, \text{and } 1 \le l \le k$$

For the Benders-dual algorithm, we have the following master problem in the k'^{th} iteration, where k' is the counter of Benders iterations:

$$\min_{\mathbf{y},\mathbf{z}} \sum_i f_i y_i + \sum_i a_i z_i + \eta$$

$$
\begin{aligned}
\text{s.t.} \sum_i z_i &\geq \max\{\sum_j d_j : \mathbf{d} \in \mathbf{D}\} \\
z_i &\leq K_i y_i, \forall i \\
\eta &\geq \sum_j \lambda_j^l d_j^l - \sum_i \pi_i^l z_i, 1 \leq l \leq k' \\
z_i &\geq 0, y_i \in \{0,1\}, \forall i, \ \eta \in \mathbf{R}
\end{aligned}
$$

A-3 Computational Results for 30×30 Instances

Table A-1: Computation Time (sec.) of C&CG Algorithm

ID\Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	0.57	0.96	1.54	2.21	1.26	1.49	1.23	1.33	0.49	0.34
2	0.64	1.17	1.73	0.88	1.16	1.66	0.98	2.24	0.90	0.31
3	1.38	3.32	2.98	2.02	1.50	1.71	2.99	3.21	2.67	0.36
4	1.20	2.03	2.21	2.36	1.09	2.11	3.00	3.05	1.25	0.32
5	0.61	1.04	0.79	1.08	1.80	2.22	0.82	1.73	0.67	0.35
6	2.16	5.13	7.48	7.34	2.36	1.10	0.98	0.71	1.01	0.35
7	0.64	0.83	0.86	1.04	1.18	1.89	0.57	0.57	0.82	0.33
8	0.88	2.44	2.34	1.63	0.86	0.71	0.46	0.67	0.63	0.32
9	2.05	1.19	2.07	0.90	0.64	0.68	0.40	0.83	0.68	0.33
10	1.59	0.77	0.79	0.43	1.45	2.57	2.66	2.45	1.65	0.36
Avg.	1.17	1.89	2.28	1.99	1.33	1.61	1.41	1.68	1.08	0.34

Table A-2: Computation Time (sec.) of Benders-dual Algorithm

ID\Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	11.51	14.53	17.57	15.97	13.21	11.47	12.41	13.61	9.58	7.78
2	19.23	13.80	18.82	12.49	16.68	17.33	14.77	21.70	17.69	13.45
3	38.77	41.20	51.69	36.19	47.52	57.77	56.16	68.43	76.89	66.09
4	11.70	13.97	14.49	13.51	13.30	13.19	15.67	12.57	8.63	7.49
5	19.82	17.71	17.87	16.63	15.64	14.04	15.97	16.45	15.44	12.94
6	12.39	20.87	27.84	21.97	23.02	20.58	20.66	17.07	17.19	19.43
7	11.06	16.87	16.87	14.51	17.22	17.75	18.34	26.37	22.39	21.51
8	10.59	14.04	16.31	13.45	10.14	7.17	7.51	5.50	6.37	7.01
9	11.83	11.30	12.95	8.94	4.28	4.12	2.47	1.85	2.42	3.05
10	20.84	16.24	17.69	22.54	23.68	27.47	34.01	33.39	44.15	32.72
Avg.	16.77	18.05	21.21	17.62	18.47	19.09	19.80	21.69	22.07	19.15

Table A-3: Number of Iterations in C&CG Algorithm

ID\Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	3	4	6	6	5	6	5	5	2	2
2	3	5	7	4	5	6	4	8	4	2
3	5	9	9	6	5	6	8	8	7	2
4	5	6	6	7	4	5	6	7	5	2
5	3	5	4	5	7	7	4	6	3	2
6	6	9	13	12	7	4	4	3	4	2
7	3	4	4	4	5	7	3	3	4	2
8	2	7	7	5	3	3	2	3	3	2
9	6	4	5	3	3	3	2	4	3	2
10	6	4	4	2	6	9	8	7	6	2
Avg.	4.2	5.7	6.5	5.4	5	5.6	4.6	5.4	4.1	2

Table A-4: Number of Iterations in Benders-dual Algorithm

ID\Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	50	56	56	40	39	33	33	33	25	22
2	77	50	60	38	44	40	35	52	43	36
3	116	107	113	81	98	96	84	95	108	94
4	49	40	39	37	34	30	33	26	22	22
5	74	67	61	52	38	36	40	42	38	33
6	49	69	62	49	55	53	54	44	41	49
7	48	57	58	47	50	48	51	66	54	52
8	40	48	51	35	29	22	23	17	19	23
9	42	30	26	25	14	14	9	7	9	13
10	82	63	59	69	61	61	70	65	79	62
Avg.	62.7	58.7	58.5	47.3	46.2	43.3	43.2	44.7	43.8	40.6

A-4 Computational Results for 70×70 Instances

Table A-5: Computation Time (sec.) of C&CG Algorithm

ID\Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	22.67	54.57	53.98	14.69	12.61	13.52	12.14	7.58	6.08	1.14
2	33.00	80.38	62.90	19.43	48.44	331.60	9.71	9.56	4.19	1.19
3	38.74	28.05	15.82	52.30	189.82	43.56	18.73	7.91	4.92	1.16
4	70.97	16.08	9.97	4.85	8.55	10.61	4.51	9.12	10.39	1.22
5	16.56	5.88	21.79	60.19	21.57	7.21	4.61	5.71	4.30	1.10
6	3.07	6.90	12.17	23.42	29.33	29.98	13.43	13.83	6.02	1.11
7	1.27	3.92	11.08	12.75	14.80	10.26	6.92	6.34	3.93	1.05
8	6.47	26.26	8.70	4.62	22.83	16.70	21.13	24.35	20.73	1.17
9	12.70	29.34	563.15	496.86	28.87	23.54	18.46	27.40	1.88	1.22
10	20.24	13.81	5.97	12.84	20.74	20.56	10.61	4.14	3.48	1.12
Avg.	22.57	26.52	76.55	70.19	39.76	50.76	12.03	11.59	6.59	1.15

Table A-6: Computation Time (sec.) of Benders-dual Algorithm

ID\Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1.0	486.1	660.8	800.7	218.9	235.7	230.7	141.3	133.7	103.9	68.3
2.0	1083.7	8798.8	7863.6	2429.9	1124.3	1246.4	593.0	129.1	104.2	119.5
3.0	1960.4	2751.9	1768.3	1644.5	1095.6	1063.1	1239.5	1163.4	1263.8	449.0
4.0	709.9	385.2	286.3	278.5	290.0	371.4	311.3	381.5	467.5	444.1
5.0	458.7	230.0	404.8	400.7	910.2	505.5	440.8	625.0	189.7	208.2
6.0	89.3	87.6	133.0	593.2	1014.7	892.3	980.2	749.1	523.0	72.4
7.0	345.4	364.3	462.0	403.9	697.5	619.2	656.5	236.0	139.6	151.1
8.0	194.2	490.0	358.6	263.3	924.4	922.2	1017.5	1116.8	282.3	169.8
9.0	1007.1	1592.0	4767.6	7235.0	1712.4	802.4	539.9	505.6	301.7	279.1
10.0	558.5	458.4	407.2	451.7	365.7	382.6	654.5	443.8	590.5	91.7
Avg.	689.3	1581.9	1725.2	1392.0	837.0	703.6	657.5	548.4	396.6	205.3

Table A-7: Number of Iterations in C&CG Algorithm

ID\Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	5	4	6	4	3	4	4	4	5	2
2	8	5	5	3	4	12	6	5	4	2
3	9	7	4	5	7	7	4	4	4	2
4	15	7	6	4	6	6	4	5	7	2
5	7	5	4	6	5	4	4	4	4	2
6	3	4	5	6	6	6	4	6	5	2
7	2	3	6	5	5	4	4	4	4	2
8	5	6	4	3	5	4	6	6	7	2
9	7	7	11	10	6	6	5	8	2	2
10	9	5	3	5	5	5	4	4	4	2
Avg.	7	5.3	5.4	5.1	5.2	5.8	4.5	5	4.6	2

Table A-8: Number of Iterations in Benders-dual Algorithm

ID\Γ	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	97	77	81	59	51	58	46	44	42	34
2	254	147	156	92	107	121	130	108	121	127
3	321	353	230	158	196	197	214	224	257	246
4	247	139	90	89	96	109	81	85	109	108
5	178	86	88	72	146	113	131	138	149	154
6	66	45	65	127	168	134	182	149	135	116
7	180	150	167	118	160	138	149	157	147	141
8	150	151	125	72	166	146	167	172	163	154
9	276	222	263	326	254	226	154	136	125	122
10	216	165	112	114	86	90	133	101	144	131
Avg.	198.5	153.5	137.7	122.7	143	133.2	138.7	131.4	139.2	133.3