European Journal of Operational Research 230 (2013) 487–494

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A linear programming approach for linear programs with probabilistic constraints

Daniel Reich *

Ford Research & Advanced Engineering, Dearborn, MI 48124, USA
School of Business, Universidad Adolfo Ibáñez, Santiago, Chile

ARTICLE INFO

Article history:
Received 2 June 2012
Accepted 22 April 2013
Available online 2 May 2013

Keywords:
Linear programming
Integer programming
Stochastic programming
Chance constrained programming
Heuristics

ABSTRACT

We study a class of mixed-integer programs for solving linear programs with joint probabilistic constraints from random right-hand side vectors with finite distributions. We present *greedy* and *dual* heuristic algorithms that construct and solve a sequence of linear programs. We provide optimality gaps for our heuristic solutions via the linear programming relaxation of the extended mixed-integer formulation of Luedtke et al. (2010) [13] as well as via lower bounds produced by their cutting plane method. While we demonstrate through an extensive computational study the effectiveness and scalability of our heuristics, we also prove that the theoretical worst-case solution quality for these algorithms is arbitrarily far from optimal. Our computational study compares our heuristics against both the extended mixed-integer programming formulation and the cutting plane method of Luedtke et al. (2010) [13]. Our heuristics efficiently and consistently produce solutions with small optimality gaps, while for larger instances the extended formulation becomes intractable and the optimality gaps from the cutting plane method increase to over 5%.

1. Introduction

Consider a linear program with a *joint probabilistic* or *chance constraint*

$$\begin{aligned} \min_{x\in X}\quad & cx \\ \text{s.t.}\quad & \mathbb{P}(Ax \geq \tilde{b}) \geq 1-\varepsilon \end{aligned} \tag{1}$$

where $X \subseteq \mathbb{R}^d$ is a polyhedron, $c \in \mathbb{R}^d, A \in \mathbb{R}^{m\times d}, \tilde{b}$ is a random vector taking values in $\mathbb{R}^m$ and $\varepsilon \in (0,1)$ is the *reliability level*.

Chance constrained models have been utilized in several applications. In the context of finance (see [19]), the joint probabilistic constraint is commonly referred to as a Value-at-Risk constraint. In supply chain management [10], these models are used to consider random supply and demand. In distillation processes [7], chance constraints are used to analyze random water inflows. Optimal vaccination strategies for preventing epidemics [20] is yet another area where chance constrained models have been applied. For additional references, we refer the reader to [17].

Problems with joint probabilistic constraints (1) can be grouped into one of the following two categories:

1. The distribution of $\tilde{b}$ is discrete and finite.
2. The distribution of $\tilde{b}$ is continuous or infinite.

Case 1 problems can at least in theory be solved to optimality, by using binary variables to cast the problems as mixed-integer programs with "big-M constraints" [18,15]. However, in practice, this approach may have limited computational tractability in some settings.

For case 2, aside from a few select distributions, no closed-form exists for evaluating $\mathbb{P}(Ax \geqslant \tilde{b})$ for a given candidate solution *x*, which prevents us from solving these problems to optimality. In lieu of exact solution methods, recent attention has focused on gradient methods [6] and on approximation methods that utilize Monte Carlo sampling [2,15,8]. The latter yields case 1 problems [12,16], which can then be solved either through mixed-integer programming [11] or through heuristic algorithms.

In this paper, we build upon the work in Pagnoncelli et al. [16] to develop specialized heuristics for case 1 problems.

Luedtke et al. [13] proved that the case 1 problems are NP-hard and to solve them they developed both a cutting plane algorithm and an extended mixed-integer programming formulation, which is a specialization of work by Miller and Wolsey [14], where all integer variables are binary. Luedtke et al. [13] leverage a natural ordering in the right-hand side to overcome the weakness of the big-M formulation. This inherent ordering has been utilized before in case 2 problems to develop a branch-and-bound algorithm [3] and we will also leverage this ordering in developing our linear programming based heuristic algorithms.

Although we focus on case 1 problems, the algorithms we develop in this paper will have direct applicability to case 2 problems

* Address: Ford Research & Advanced Engineering, Dearborn, MI 48124, USA.
E-mail address: dreich8@ford.com

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.04.049

when used in conjunction with sampling approaches. We compare our algorithms with the cutting plane method and extended mixed-integer programming formulation of Luedtke et al. [13]. We show that while their extended formulation becomes intractable for larger problems and their cutting plane method produces increasing optimality gaps, our heuristics remain efficient and provide near-optimal solutions.

The remainder of this paper is organized as follows. Section 2 introduces the mixed-integer programming problem that we aim to solve and presents the extended formulation of Luedtke et al. [13]. (We refer the reader to Luedtke et al. [13] and Atamtürk et al. [1] for detail on the cutting plane method.) In Section 3, we present our *greedy* and *dual* heuristic. In Section 4, we prove that the worst-case solution quality for our heuristic algorithms is arbitrarily far from optimal. In Section 5, we present a computational study that compares our algorithms with the extended formulation and the cutting plane method. Section 6 summarizes our contributions and discusses future research directions.

2. Background

Consider case 1 of chance constrained problem (1), where the distribution of the right-hand side $\tilde{b}$ is discrete and has scenarios b^{ω} with corresponding probabilities p_{ω} for all $\omega \in \Omega$. For simplicity, without loss of generality, we assume that $b^{\omega} \geqslant 0$ for all $\omega \in \Omega$. By introducing $|\Omega|$ binary variables, we can restate this problem as a mixed-integer program with the following big-M formulation:

$$\text{(big-M)} \quad \min_{x \in X} \quad cx \tag{2}$$

$$\text{s.t.} \quad Ax + z_{\omega} b^{\omega} \geqslant b^{\omega} \quad \omega \in \Omega \tag{3}$$

$$\sum_{\omega \in \Omega} p_{\omega} z_{\omega} \leqslant \varepsilon \tag{4}$$

$$z \in \{0,1\}^{|\Omega|}, \tag{5}$$

where the big-M constant is b^{ω}, for each ω. If binary variable $z_{\omega} = 0$, then $Ax \geqslant b^{\omega}$ ($\geqslant 0$ by assumption). If $z_{\omega} = 1$, then we have $Ax \geqslant 0$, which is satisfied because $\varepsilon < 1$ implies that there will be at least one $\omega \in \Omega$ such that $z_{\omega} = 0$. The knapsack inequality (4) is equivalent to the probabilistic constraint

$$\sum_{\omega \in \Omega} p_{\omega}(1 - z_{\omega}) \geqslant 1 - \varepsilon.$$

2.1. Ordering the scenarios

Consider a single row in the big-M formulation (3):

$$A_i x + z_{\omega} b_i^{\omega} \geqslant b_i^{\omega} \quad \omega \in \Omega, \tag{6}$$

where A_i is the ith row of the constraint matrix A and b_i^{ω} is the ith row of the right-hand side scenario b^{ω}. Let $\omega(i,k)$ be the scenario with kth largest right-hand side $b_i^{\omega(i,k)}$ for row i. Then for every row, there exists an index l_i such that

$$\sum_{k=1}^{l_i - 1} p_{\omega(i,k)} \leqslant \varepsilon < \sum_{k=1}^{l_i} p_{\omega(i,k)}.$$

In other words, it would not be possible to remove all scenarios $\{\omega(i,1), \ldots, \omega(i,l_i)\}$ without exceeding ε; however, it would be possible to remove all scenarios $\{\omega(i,1), \ldots, \omega(i,l_i - 1)\}$. Therefore, any feasible solution x to case 1 of problem (1) must satisfy

$$A_i x \geqslant b_i^{\omega(i,l_i)} \text{ for all } i \in I.$$

2.2. The tight-M formulation

Using l_i and $w(i,k)$, we can replace the big-M formulation (2)–(5) with the following *tight-M* mixed-integer program:

$$\text{(tight-M)} \quad \min_{x \in X} \quad cx \tag{7}$$

$$\text{s.t.} \quad A_i x + z_{\omega(i,k)} \left(b_i^{\omega(i,k)} - b_i^{\omega(i,l_i)} \right) \geqslant b_i^{\omega(i,k)} \quad i \in I, \ 1 \leqslant k \leqslant l_i - 1 \tag{8}$$

$$\sum_{\omega \in \Omega} p_{\omega} z_{\omega} \leqslant \varepsilon \tag{9}$$

$$z \in \{0,1\}^{|\Omega|}, \tag{10}$$

where $b_i^{\omega(i,k)} - b_i^{\omega(i,l_i)}$ strengthens the formulation and $1 \leqslant k \leqslant l_i - 1$ avoids the redundant constraints that were identified in ordering the scenarios according to row. For more detail on the tight-M formulation (7)–(10) and on additional valid inequalities that can be used to strengthen it, we refer the reader to Luedtke et al. [13] and to work on *mixing sets* by Atamtürk et al. [1], Günlük and Pochet [5], Guan et al. [4], Miller and Wolsey [14] and Küçükyavuz [9].

2.3. The extended formulation

Luedtke et al. [13] make further use of the ordered scenarios by defining additional binary variables u_i^{ω}, for all $\omega \in \{\omega(1,i), \ldots, \omega(i,l_i)\}$, to arrive at the following extended mixed-integer programming formulation:

$$\text{(extended)} \quad \min_{x \in X} \quad cx \tag{11}$$

$$\text{s.t.} \quad A_i x + \sum_{k=1}^{l_i - 1} u_i^{\omega(i,k)} \left(b_i^{\omega(i,k)} - b_i^{\omega(i,k+1)} \right) \geqslant b_i^{\omega(i,1)} \ i \in I \tag{12}$$

$$u_i^{\omega(i,k)} - u_i^{\omega(i,k+1)} \geqslant 0 \quad i \in I, 1 \leqslant k \leqslant l_i - 1 \tag{13}$$

$$z_{\omega(i,k)} - u_i^{\omega(i,k)} \geqslant 0 \quad i \in I, 1 \leqslant k \leqslant l_i - 1 \tag{14}$$

$$\sum_{\omega \in \Omega} p_{\omega} z_{\omega} \leqslant \varepsilon \tag{15}$$

$$u_i^{\omega(i,l_i)} = 0 \quad i \in I \tag{16}$$

$$u_i^{\omega(i,k)} \in \{0,1\} \quad i \in I, 1 \leqslant k \leqslant l_i \tag{17}$$

$$z \in \{0,1\}^{|\Omega|}, \tag{18}$$

which they prove is a valid formulation for the tight-M problem (7)–(10). Constraint (13) orders the binary variables u and constraint (14) connects those binary variables to their corresponding scenarios. This allows us to require only a single constraint (12) for each row of A, which accounts for all scenarios corresponding to those individual rows. For further detail on the extended formulation, we refer the reader to Luedtke et al. [13].

3. The greedy and dual algorithms

In this section, we present *greedy* and *dual* heuristic algorithms for solving case 1 of chance constrained problem (1). As we demonstrate in Section 5, the mixed-integer programming formulations – even the extended one – have limited computational tractability. By leveraging the ordering detailed in Section 2, we develop effective and scalable algorithms for heuristically solving case 1 problems.

3.1. The greedy and dual algorithms

Consider the tight-M formulation (7)–(10). Our heuristic algorithms solve a sequence of linear programming problems similar to (7)–(10), while leveraging order to reduce the linear program problem size. For each row i of constraint matrix A, we need only include constraint (8) for the non-removed scenario $\omega(i,k)$ for which $b_i^{\omega(i,k)}$ is largest. The linear program we solve iteratively can be written as follows:

$$(\text{tight-LP}|z) \quad \min_{x \in X} \quad cx \tag{19}$$

$$\text{s.t.} \quad A_i x \geqslant \max_{\omega \in \Omega}(1 - z_\omega) b_i^\omega, \quad i \in I. \tag{20}$$

3.1.1. The greedy method for removing scenarios with equal or unequal probabilities

Initially $z = \vec{0}$. At each iteration, we solve "tight-LP|z" to obtain the optimal decision vector x^*. We then solve an additional " tight-LP| max{z,e^ω}" linear program, where e^ω is the unit vector with 1 in position ω and zeros elsewhere, for each active constraint $A_i x^* = \max_{\omega \in \Omega}(1 - z_\omega) b_i^\omega$. The scenario ω that yields the greatest ratio of objective value improvement to scenario probability is removed by setting $z_\omega = 1$. We continue to iterate until the constraint $\sum_{\omega \in \Omega} p_\omega z_\omega \leqslant \varepsilon$ in (9) does not have sufficient slack to set another $z_\omega = 1$. Pseudocode for the greedy method is given in Algorithm 1.

Algorithm 1. Greedy algorithm for solving min cx s.t. $\mathbb{P}(Ax \geqslant \tilde{b}) \geqslant 1 - \varepsilon$.

```
Initialize $r = \varepsilon$
Initialize $z_\omega = 0$ for every $\omega \in \Omega$
if no feasible solution exists for "tight − LP|$z = \vec{0}$" then
  Run Algorithm 3 to remove infeasibility and update $r$ and $z$
end if
while true do
  Solve "tight-LP|$z$" to obtain optimal solution $x^*$ with
  objective $v^*$
  for all $\omega \in \Omega$ do
    $v^\omega = v^*$
  end for
  for all $i \in I$ such that $A_i x^* = \max_{\omega \in \Omega}(1 - z_\omega) b_i^\omega$ do
    for all $\omega \in \text{argmax}_{s \in \Omega}(1 - z_s) b_i^s$ do
      if ($p_\omega < r$ and $z_\omega = = 0$ and $v^\omega = = v^*$) then
        Let $v^\omega$ = "tight-LP| max{$z$,$e^\omega$}"
      end if
    end for
  end for
  if $v^* > \min_{\omega \in \Omega} v^\omega$ then
    Let $\omega \in \text{argmax}_{s \in \Omega}(v^* - v^s)/p_s$
    Let $r = r - p_\omega$
    Let $z_\omega = 1$
  else
    Return $v^*$
  end if
end while
```

3.1.2. The dual method for removing scenarios with equal or unequal probabilities

Initially $z = \vec{0}$. At each iteration, we solve "tight-LP|z" to obtain the dual price vector π. We then compute, for all $\omega \in \Omega$ such that $z_\omega = 0$,

$$d_\omega = \sum_{i \in I} \pi_i \left[\max_{s \in \Omega}(1 - z_s) b_i^s - \max_{s \in \Omega \setminus \omega}(1 - z_s) b_i^s \right], \tag{21}$$

which is the sum of the dual prices weighted by the magnitude of the change in the right-hand side that would be induced by removing scenario ω. Note that for any $\omega \in \Omega$, only those constraints i for which $\omega \in \text{argmax}_{s \in \Omega : z_s = 0} b_i^s$ contribute to d_ω. The scenario ω that yields the greatest ratio of d_ω to scenario probability p_ω is removed by setting $z_\omega = 1$. We continue to iterate until the constraint $\sum_{\omega \in \Omega} p_\omega z_\omega \leqslant \varepsilon$ in (9) does not have sufficient slack to set another $z_\omega = 1$. Pseudocode for the dual method is given in Algorithm 2.

Algorithm 2. Dual algorithm for solving min cx s.t. $\mathbb{P}(Ax \geqslant \tilde{b}) \geqslant 1 - \varepsilon$.

```
Initialize $r = \varepsilon$
Initialize $z_\omega = 0$ for every $\omega \in \Omega$
if no feasible solution exists for "tight − LP|$z = \vec{0}$" then
  Run Algorithm 3 to remove infeasibility and update $r$ and $z$
end if
while true do
  Solve "tight-LP|$z$" to obtain optimal objective $v^*$ and dual
  prices $\pi$
  for all $\omega \in \Omega$ do
    if $z_\omega = = 0$ and $p_\omega \leqslant r$ then
      $d_\omega = \sum_{i \in I} \pi_i [\max_{s \in \Omega}(1 - z_s) b_i^s - \max_{s \in \Omega \setminus \omega}(1 - z_s) b_i^s]$
    else
      $d_\omega = 0$
    end if
  end for
  if $\max_{\omega \in \Omega} d_\omega > 0$ then
    Let $\omega \in \text{argmax}_{s \in \Omega} d_s/p_s$
    Let $r = r - p_\omega$
    Let $z_\omega = 1$
  else
    Return $v^*$
  end if
end while
```

Remark 1. If a feasible solution exists to "tight − LP|$z = \vec{0}$" (19), (20), then Algorithms 1 and 2 provide feasible solutions to case 1 of chance constrained problem (1).

Remark 1 assumes that a feasible solution exists to "tight − LP|$z = \vec{0}$", which cannot be expected in general. When no feasible solution exists, the dual and greedy algorithms require the following additional routine.

3.1.3. The Lagrange multiplier method for removing infeasibility from "tight − LP|$z = \vec{0}$" (19) and (20)

We select a $\lambda \in \mathbb{R}_+^m$ and $\delta \in (0,1)$ and solve the Lagrange multiplier linear program

$$(\text{Lagrange-LP}|\lambda, \delta) \quad \min_{x \in X} \quad \sum_{i \in I} \lambda_i (\max_{\omega \in \Omega} b_i^\omega - A_i x) \tag{22}$$

$$\text{s.t.} \quad A_i x \geqslant \max_{\omega \in \Omega \setminus \Omega_i} b_i^\omega \quad i \in I \tag{23}$$

where

$$b_i^\omega \geqslant b_i^s \text{ for all } \omega \in \Omega_i, \quad s \in \Omega \setminus \Omega_i \tag{24}$$

and

$$\sum_{\omega \in \Omega_i} p_\omega \leqslant \delta < p_s + \sum_{\omega \in \Omega_i} p_\omega \text{ for all } s \in \Omega \setminus \Omega_i. \tag{25}$$

In other words, Ω_i is the largest possible subset of highest right-hand side values with total probability less than δ. We start with a small $\delta \ll \varepsilon$. λ can be selected based on problem characteristics or set to a vector of ones. If the Lagrange-LP|λ,δ (22), (23) is infeasible, slowly increase δ by step size h up to a limit of ε. If ε is reached, the original chance constrained problem is infeasible. Alternatively, if an optimal solution x is achieved, we identify the scenarios that must be removed for that solution to be feasible for tight-LP|z (19), (20). If those scenarios can be removed while satisfying the knapsack constraint (9), we remove them and proceed with the greedy or dual methods. Otherwise, we rerun this method for removing infeasibility with a smaller step size h and/or another λ. While this method is not guaranteed to find a feasible solution, we show through our computational study its effectiveness in practice. Pseudocode for the Lagrange multiplier removal method is provided in Algorithm 3.

Algorithm 3. Remove infeasibility and update r and z.

```
User inputs a step size $h$
User inputs Lagrange multipliers $\lambda_i > 0$ for violating
  constraints $A_i x \geqslant \max_{\omega \in \Omega} b_i^{\omega}$ for all $i \in I$
Let $\delta = 0$
while true do
  if $\delta \geqslant \varepsilon$ then
    Return the chance constrained problem is infeasible
  end if
  Let $\delta = \delta + h$
  for all $i$ in $1, \ldots, m$ do
    Let $\Omega_i = \emptyset$ and $r = \delta$
    while $r > 0$ do
      for all $\omega \in \mathrm{argmax}_{s \in \Omega \setminus \Omega_i} b_i^s$ do
        $r = r - p_s$
        if $r > 0$ then
          $\Omega_i = \Omega_i \cap s$
        end if
      end for
    end while
  end for
  Solve Lagrange-LP|$\lambda$, $\delta$ (22) and (23) to obtain the optimal
  solution $x^*$
  if a feasible solution exists to Lagrange-LP|$\lambda$,$\delta$ then
    $\Omega' = \{\omega \in \Omega : Ax^* < b^{\omega}\}$
    $r = \varepsilon - \sum_{\omega \in \Omega'} p_{\omega}$
    Let $z_{\omega} = 1$ for every $\omega \in \Omega'$ and $z_{\omega} = 0$ for every $\omega \in \Omega \setminus \Omega'$
    if $r \geqslant 0$ then
      Return $r$ and $z$
    else
      Request a smaller $h$ and/or another $\lambda$ from user and
  rerun Algorithm 3
    end if
  end if
end while
```

3.1.4. Optimality gaps for the greedy and dual algorithms

We obtain optimality gaps, which are useful in assessing the quality of solutions obtained through our heuristic algorithms, via a linear programming relaxation of the extended formulation (11)–(18) and via lower bounds from the cutting plane method presented in Luedtke et al. [13]. As we will see in Section 5, while the linear programming relaxation of the extended formulation provides tight optimality gaps, it has limited scalability. Using lower bounds from the cutting plane method remains a practical alternative for establishing optimality gaps for our heuristic methods.

4. Worst-case solution quality for the greedy and dual methods

In the next section, we provide computational evidence for the effectiveness and efficiency of our greedy and dual heuristic algorithms. However, these heuristics can in fact produce solutions that are arbitrarily far from optimal.

Theorem 1. *Let v^* be the optimal solution to a case 1 chance constrained problem (1). Let v^1 and v^2 be feasible solutions obtained by Algorithms 1 and 2, respectively. Then $\nexists M < \infty$ such that*

$$\frac{\min\{v^1, v^2\} - v^*}{v^*} < M$$

for all case 1 chance constrained problems (1).

Proof. Assume $\exists M < \infty$ such that $(\min\{v^1, v^2\} - v^*)/v^* < M$ for all case 1 chance constrained problems (1). We construct a counterexample where $(\min\{v^1, v^2\} - v^*)/v^* > M$ as follows.

Consider the chance constrained problem

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad & \mathbb{P}(x_1 \geqslant \tilde{b}_1, x_2 \geqslant \tilde{b}_2) \geqslant \frac{1}{2}. \end{aligned} \tag{26}$$

Assume $c_1 > 0$ and $c_2 > 0$. Let $b^1 = (2, -1)$, $b^2 = (2, 0)$, $b^3 = (0, 1)$, and $b^4 = (0, 2)$ be 4 scenarios, each with probability 1/4. $\varepsilon = 1/2$ allows us to remove at most two of the four scenarios and – because they all have probability 1/4 – removing any two provides a feasible solution to chance constrained problem (26). The key to this counterexample is that Algorithms 1 and 2 remove the same two scenarios, neither of which is removed in the optimal solution to chance constrained problem (26).

The first tight-LP|$z = \vec{0}$ problem solved by Algorithms 1 and 2 is

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad & x_1 \geqslant 2 \\ & x_2 \geqslant 2. \end{aligned} \tag{27}$$

The optimal solution to (27) is $x_1 = 2$ and $x_2 = 2$ with objective value $2c_1 + 2c_2$.

Removing scenario b^4 is the only one out of the four scenarios that relaxes the constraint system in (27). Therefore, both Algorithms 1 and 2 remove b^4 and the resulting tight-LP|$z = (0, 0, 0, 1)$ linear program is

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad & x_1 \geqslant 2 \\ & x_2 \geqslant 1. \end{aligned} \tag{28}$$

The optimal solution to (28) is $x_1 = 2$ and $x_2 = 1$ with objective value $2c_1 + c_2$.

Removing scenario b_3 is the only one out of the three remaining scenarios that relaxes the constraint system in (28). Therefore, both Algorithms 1 and 2 remove b_3 and the resulting tight-LP|$z = (0, 0, 1, 1)$ linear program is

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad & x_1 \geqslant 2 \\ & x_2 \geqslant 0. \end{aligned} \tag{29}$$

The optimal solution to (29) is $x_1 = 2$ and $x_2 = 0$ with objective value $v^1 = v^2 = 2c_1$. Both Algorithms 1 and 2 terminate because no further removals are possible while satisfying the knapsack constraint (9).

Removing b_1 and b_2 from chance constrained problem (26) yields the linear program

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad & x_1 \geqslant 0 \\ & x_2 \geqslant 2. \end{aligned} \tag{30}$$

The optimal solution to (30) is $x_1 = 0$ and $x_2 = 2$ with objective value $2c_2$.

By assumption, $c_1 > 0$ and $c_2 > 0$. Regardless of their specific values, Algorithms 1 and 2 provide a feasible solution to the chance constrained problem (26) with an objective value of $2c_1$. However, we have just shown in (30) that another feasible solution for (26) exists with cost $2c_2$. Therefore, if we choose $c_2 = 1$ and $c_1 > 1$, we have that $v^* = 2$ and

$$\frac{\min\{v^1, v^2\} - v^*}{v^*} = \frac{2c_1 - 2}{2} = c_1 - 1.$$

Therefore, by choosing $c_1 > M + 1$ we contradict our assumption.

The counterexample in the preceding proof demonstrates the worst-case solution for these algorithms. Note though that we obtain optimality gaps equal to c_1, which is arbitrarily large, for the heuristic solutions via the extended linear programming relaxation. Therefore, we do have a warning against the heuristic solutions achieved in this worst-case counterexample. In the next section, we provide computational evidence that these heuristics are effective, efficient and scalable in practice by testing them on the complete set of instances from Luedtke et al. [13].

5. Computational results

We focused our computational study on a probabilistic version of the classical transportation problem. The deterministic transportation problem is to find the least costly transportation strategy x for which demand b_j is satisfied at each location $j \in J$ from the available supplies M_i from suppliers $i \in I$, where the cost of sending x_{ij} units from supplier i to customer j is c_{ij}. We consider random demand $\tilde{b}$ and formulate the problem as

$$\begin{aligned} \min_x \quad & \sum_{i \in I, j \in J} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j \in J} x_{ij} \leqslant M_i \\ & \forall i \in I \\ & \mathbb{P}\left\{ \sum_{i \in I} x_{ij} \geqslant \tilde{b}_j, j \in J \right\} \geqslant 1 - \varepsilon \\ & x_{ij} \geqslant 0 \\ & \forall i \in I, \quad j \in J. \end{aligned} \tag{31}$$

5.1. Instances

We tested our algorithms on the complete set of instances introduced in Luedtke et al. [13]. These instances are divided into two categories: one where the scenarios have equal probabilities and the other where the probabilities are unequal. All instances have 40 suppliers with either 100 or 200 customers, and between 1000 and 3000 scenarios. For each problem size, there are 5 instances, which we tested for ε = 0.05, 0.10, 0.15 and 0.20. The random demands were generated via independent normal distributions with randomly generated means and variances. For more information about the generation of these instances, we refer the reader to Luedtke et al. [13].

Table 1
Average running times in seconds over 5 instances for each solution method.

Solution method			Dual	Greedy	Cutting plane	Extended formulation	Root LP extended formulation
ε	m	n	Equal probability scenarios				
0.05	100	1000	0.2	1.7	0.7	0.5	0.2
	100	2000	0.3	3.5	3.1	3.3	0.8
	200	2000	0.7	12.9	3.1	4.2	5.4
	200	3000	1.1	24.2	6.2	18.5	13.2
0.10	100	1000	0.3	3.6	10.4	2.4	1.4
	100	2000	0.6	7.7	37.2	7.9	4.5
	200	2000	1.4	29.7	693.5	130.4	136.1
	200	3000	2.1	51.7	1719.7	564.6	488.7
0.15	100	1000	0.4	5.2	755.8	508.9	12.3
	100	2000	1.0	10.5	3602.4	1101.3	84.2
	200	2000	2.0	42.3	3600.6	3600.1	200.8
	200	3000	3.4	71.6	3604.9	3600.1	461.6
0.20	100	1000	0.6	7.3	3600.1	3600.2	42.8
	100	2000	1.2	14.9	3602.0	3600.1	133.2
	200	2000	2.7	61.6	3605.2	3620.6	535.0
	200	3000	4.2	108.7	3606.1	3602.4	1897.7
			Unequal probability scenarios				
0.05	100	1000	0.3	4.1	4.9	1.9	0.8
	100	2000	0.5	7.3	19.8	6.5	3.7
	200	2000	1.2	33.4	69.9	30.5	19.7
	200	3000	2.0	52.6	133.3	119.4	56.1
0.10	100	1000	0.4	6.8	46.4	11.5	5.5
	100	2000	0.9	12.8	1907.3	36.4	26.2
	200	2000	2.1	61.1	2812.6	472.7	183.9
	200	3000	3.4	92.2	3601.5	1511.6	566.5
0.15	100	1000	0.6	6.9	832.7	66.9	19.5
	100	2000	1.2	13.5	3600.6	586.3	88.7
	200	2000	2.9	59.4	3600.7	3396.9	194.6
	200	3000	4.7	97.1	3605.5	3600.1	501.4
0.20	100	1000	0.7	12.0	3600.1	808.9	36.0
	100	2000	1.6	22.9	3601.1	3358.5	199.6
	200	2000	3.7	103.7	3600.4	3600.7	1247.2
	200	3000	5.8	172.7	3601.0	3602.5	2031.6

5.2. Instances requiring Algorithm 3

In 10% of the instances, no feasible solution exists to "tight-LP|$z = \vec{0}$" (19) and (20). For those instances, the greedy and dual methods require Algorithm 3 to remove infeasibility. We use a vector of ones for the Lagrange multipliers λ and a step size $h = 1/1000$, which removes infeasibility in all instances requiring Algorithm 3. The running times and optimality gaps for these instances were similar to all other instances, so we present all results together, whether they require Algorithm 3 or not.

5.3. Implementation

We implemented Algorithms 1 (greedy), 2 (dual) and 3 (remove infeasibility), as well as the extended formulation and cutting plane method, in C. We did not implement Algorithm 1 (greedy) in parallel. All computations were performed on identical machines each having x86_64 architecture, an Intel Xeon @2.5 GHz processor, 16 GB RAM and running CentOS 4 Linux. The callable library for CPLEX 12.1 was used to solve all linear and integer programs. We required the same time limit of 1 h and memory limit of 2 GB that was required in Luedtke et al. [13]. However, our implementation of the cutting plane method most likely differs, as specific implementation details were not provided in Luedtke et al. [13].

5.4. Problem sizes

For simplicity in comparing problem sizes, let us consider the case of equi-probable scenarios with $m = 200$ customers, $n = 3000$ scenarios and $\varepsilon = 0.20$. The dual algorithm requires solving $n\varepsilon = 600$ linear programs each with $m = 200$ demand constraints. The greedy algorithm requires solving up to (less if not all demand constraints are active) $mn\varepsilon = 120{,}000$ linear programs each with 200 demand constraints, but these can be solved in $n\varepsilon = 600$ iterations where at each iteration up to (less if not all demand constraints are active) $m = 200$ linear programs are solved in parallel. The tight-M formulation requires $m\ n\ \varepsilon = 120{,}000$ demand constraints (8) and up to $n = 3000$ binary variables. The extended formulation requires $mn\varepsilon + n = 123{,}000$ binary variables, $2mn\varepsilon = 240{,}000$ ordering constraints (13) and (14) and $m = 200$ demand constraints (12).

5.5. Running time and scalability

Table 1 provides the average running times over 5 instances for the dual and greedy algorithms, the cutting plane method, the extended formulation, and the linear programming relaxations of the extended formulation. The latter linear programming relaxation is used to obtain optimality gaps for our heuristic algorithms, and also provides further insight into the scalability of the extended formulation. The findings for Table 1 can be summarized as follows:

- The average running time of the dual method is under 6 s for all problem sizes.
- The average running time of the greedy method increases by a factor of around 4 when the number of customers m is doubled, and scales roughly linearly in both the number of scenarios n and in ε.

Table 2
Maximum relative optimality gaps over 5 instances. Gaps for greedy and dual algorithms are equal for all scenarios.

Solution method (gap obtained using)			Extended formulation (CPLEX) (%)	Cutting plane (CPLEX) (%)	Greedy and dual (root LP ext. form.) (%)	Greedy and dual (cutting plane bound) (%)
ε	m	n	Equal probability scenarios			
0.05	100	1000	0.0	0.0	0.1	0.1
	100	2000	0.0	0.0	0.4	0.4
	200	2000	0.0	0.0	0.4	0.4
	200	3000	0.0	0.0	0.2	0.2
0.10	100	1000	0.0	0.0	0.3	0.3
	100	2000	0.0	0.0	0.7	0.7
	200	2000	0.0	0.0	0.3	0.3
	200	3000	0.0	0.2	0.4	0.4
0.15	100	1000	0.0	0.0	0.6	0.5
	100	2000	0.0	0.9	0.9	0.9
	200	2000	2.1	1.0	0.8	0.8
	200	3000	2.3	2.3	0.9	0.9
0.20	100	1000	0.7	1.1	1.6	1.4
	100	2000	0.8	3.4	1.6	1.7
	200	2000	4.8	4.4	2.0	2.0
	200	3000	∞	4.7	2.1	2.3
			Unequal probability scenarios			
0.05	100	1000	0.0	0.0	0.2	0.2
	100	2000	0.0	0.0	0.6	0.6
	200	2000	0.0	0.0	0.4	0.4
	200	3000	0.0	0.0	0.5	0.5
0.10	100	1000	0.0	0.0	0.6	0.7
	100	2000	0.0	0.2	1.0	1.0
	200	2000	0.0	0.3	0.5	0.5
	200	3000	0.0	1.0	0.6	0.8
0.15	100	1000	0.0	0.0	0.7	0.7
	100	2000	0.0	2.1	1.3	1.6
	200	2000	0.4	2.4	0.8	1.0
	200	3000	2.0	3.0	1.2	1.7
0.20	100	1000	0.0	1.1	1.4	1.3
	100	2000	0.7	4.2	1.8	2.3
	200	2000	4.6	4.5	1.8	2.0
	200	3000	∞	5.2	2.2	2.9

- The average running time of the cutting plane method requires the 1 h time limit for larger instances, but it can produce feasible solutions in all cases.
- The average running time of the extended formulation is increasing at an exponential rate with ε, and does not have a consistent scaling factor when doubling either the number of customers m or the number of scenarios n. The extended formulation cannot solve to optimality within the 1 h time limit, as ε, m and n increase. Moreover, it cannot produce even a single feasible solution within 1 h for the largest case.
- The average running time of the linear programming relaxation for the extended formulation increases to over 30 min for the largest instances.

5.6. Optimality gaps

Table 2 provides the maximum optimality gaps over 5 instances for the dual and greedy algorithms, using lower bounds produced by the cutting plane method and using the linear programming relaxation of the extended formulation. For the dual and greedy algorithms, we compute the gaps as the differences between the heuristic solutions and the bounds divided by the heuristic solutions, which is consistent with the measure used by CPLEX to compute the relative gaps we report for both the extended formulation and the cutting plane method. The findings for Table 2 can be summarized as follows:

- The extended formulation, while superior for smaller instances, cannot provide even a single feasible solution for 200 customers, 3000 scenarios and $\varepsilon = 0.20$ within the 1 h time limit.
- The optimality gaps for the cutting plane method increase to over 5% for the largest instances.
- The greedy and dual algorithms provide the exact same solutions for all instances.
- The greedy and dual algorithms provide a solution within 2.2% of optimal for all instances.
- The extended linear programming relaxation and the lower bounds from the cutting plane method provide similar optimality gaps for most instances.
- The bounds from the cutting plane method can be used in cases where the extended linear programming relaxation cannot be solved due to its limited scalability.

For the instances with $\varepsilon \leqslant 10\%$, for which the extended formulation solved to optimality, we were able to verify that the solutions obtained through the greedy and dual algorithms were indeed suboptimal, by the amount reflected in the optimality gaps.

6. Summary and conclusions

We have developed two linear programming based heuristic methods for solving linear programs with joint probabilistic constraints, where the constraint matrix is deterministic and the right-hand side vector is random. Our greedy and dual heuristics (Algorithms 1 and 2) account for infeasibility and for scenarios with non-equal probabilities.

We compare our greedy and dual algorithms against the cutting plane method and extended mixed-integer programming formulation proposed by [13] for all their transportation instances, both for the reliability levels $\varepsilon = 0.05$, 0.10 that they considered and for slightly higher ones of $\varepsilon = 0.15$, 0.20. Our computational study shows that while both the extended mixed-integer programming formulation and cutting plane method perform well for $\varepsilon = 0.05$, 0.10, even for $\varepsilon = 0.15$, the larger instances of Luedtke et al. [13] cannot be solved to optimality by their exact solution methods. Moreover, for $\varepsilon = 0.20$, their extended formulation is unable to identify even a single feasible solution within the 1 h time limit and the optimality gap from their cutting plane method exceeds 5%.

Our greedy and dual heuristics achieve optimality gaps of at most 2.2% for all instances for all reliability levels. These optimality gaps are obtained through lower bounds produced by the cutting plane method and linear programming relaxations of the extended mixed-integer programming formulation. While the greedy and dual algorithms provide the exact same solutions for all instances in our computational study, the dual heuristic provides remarkable speed – less than 6 s for any single instance.

In future work, the greedy and dual algorithms presented in this paper can be applied in conjunction with sampling approaches to chance constrained problems where the underlying distribution of the right-hand side vector is continuous. These algorithms can also be used to warmstart exact integer programming approaches in order to reduce their running time.

Acknowledgments

Special thanks to the reviewer whose insight and thorough feedback led to a significant improvement in the dual algorithm presented. We thank James Luedtke, Shabbir Ahmed and George Nemhauser for providing their transportation problem instances from *An Integer Programming Approach for Linear Programs with Probabilistic Constraints*. We thank Bernardo Pagnoncelli for suggesting that these instances would be ideal for our computational study. This research was funded by ANILLO Grant ACT-88 and Basal Project CMM, Universidad de Chile.

References

[1] A. Atamtürk, G.L. Nemhauser, M.W. Savelsbergh, The mixed vertex packing problem, Mathematical Programming 89 (2000) 35–53.
[2] M. Campi, S. Garatti, A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality, Journal of Optimization Theory and Applications (2010) 1–24.
[3] M.S. Cheon, S. Ahmed, F. Al-Khayyal, A branch-reduce-cut algorithm for the global optimization of probabilistically constrained linear programs, Mathematical Programming 108 (2006) 617–634.
[4] Y. Guan, S. Ahmed, G. Nemhauser, Sequential pairing of mixed integer inequalities, Discrete Optimization 4 (1) (2007) 21–39.
[5] O. Günlük, Y. Pochet, Mixing mixed-integer inequalities, Mathematical Programming 90 (2001) 429–457.
[6] R. Henrion, Gradient estimates for Gaussian distribution functions: application to probabilistically constrained optimization problems, Control and Optimization 4 (4) (2012) 655–668.
[7] R. Henrion, A. Möller, Optimization of a continuous distillation process under random inflow rate, Computers & Mathematics with Applications 45 (2003) 247–262.
[8] R. Henrion, W. Römisch, Metric regularity and quantitative stability in stochastic programs with probabilistic constraints, Mathematical Programming 84 (1999) 55–88.
[9] S. Küçükyavuz, On mixing sets arising in chance-constrained programming, Mathematical Programming 132 (2010) 31–56.
[10] M. Lejeune, A. Ruszczynski, An efficient trajectory method for probabilistic inventory production–distribution problems, Operations Research 55 (2007) 378–394.
[11] J. Luedtke, An integer programming and decomposition approach to general chance-constrained mathematical programs, Lecture Notes in Computer Science 6080 (2010) 271–284.
[12] J. Luedtke, S. Ahmed, A sample approximation approach for optimization with probabilistic constraints, SIAM Journal on Optimization 19 (2008) 674–699.
[13] J. Luedtke, S. Ahmed, G. Nemhauser, An integer programming approach for linear programs with probabilistic constraints, Mathematical Programming 122 (2010) 247–272.
[14] A.J. Miller, L.A. Wolsey, Tight formulations for some simple mixed integer programs and convex objective integer programs, Mathematical Programming 98 (2003) 73–88.
[15] B.K. Pagnoncelli, S. Ahmed, A. Shapiro, Sample average approximation method for chance constrained programming: theory and applications, Journal of Optimization Theory and Applications 142 (2) (2009) 399–416..
[16] B.K. Pagnoncelli, D. Reich, M.C. Campi, Risk-return trade-off with the scenario approach in practice: a case study in portfolio selection, Journal of Optimization Theory and Applications 155 (2) (2012) 707–722.

[17] A. Prékopa, Probabilistic programming, Handbooks in Operations Research and Management Science 10 (2003) 267–351.

[18] A. Ruszczyński, Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra, Mathematical Programming 93 (2) (2002) 195–215.

[19] A. Shapiro, A. Philpott, A Tutorial on Stochastic Programming, 2007. <www2.isye.gatech.edu/∼ashapiro/publications.html>.

[20] M.W. Tanner, L. Ntaimo, IIS branch-and-cut for joint chance-constrained stochastic programs and application to optimal vaccine allocation, European Journal of Operational Research 207 (1) (2010) 290–296.