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