



Integration of progressive hedging and dual decomposition in stochastic integer programs



Ge Guo^a, Gabriel Hackebeil^b, Sarah M. Ryan^{a,*}, Jean-Paul Watson^b, David L. Woodruff^c

^a Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA 50011, USA

^b Discrete Math and Complex Systems Department, Sandia National Laboratories, Albuquerque, NM 87185, USA

^c Graduate School of Management, University of California Davis, Davis, CA 95616, USA

ARTICLE INFO

Article history:

Received 2 January 2015
Received in revised form
24 March 2015
Accepted 26 March 2015
Available online 7 April 2015

Keywords:

Stochastic programming
Mixed-integer programming
Progressive hedging
Dual decomposition
Lower bounding

ABSTRACT

We present a method for integrating the Progressive Hedging (PH) algorithm and the Dual Decomposition (DD) algorithm of Carøe and Schultz for stochastic mixed-integer programs. Based on the correspondence between lower bounds obtained with PH and DD, a method to transform weights from PH to Lagrange multipliers in DD is found. Fast progress in early iterations of PH speeds up convergence of DD to an exact solution. We report computational results on server location and unit commitment instances.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Stochastic mixed-integer programs find a broad application in energy, facility location, production scheduling and other areas where a set of decisions must be taken before full information is revealed on some random events and some of the decisions are required to be integer [1]. The combination of uncertainty and discrete decisions leads to the difficulty in solving stochastic mixed-integer programs.

Until now much progress has been made in developing algorithms to solve these problems, extending from special instances [13,15,24] to more general stochastic mixed-integer programs [3,22]. Carøe and Schultz [2] developed a dual decomposition (DD) algorithm based on scenario decomposition and Lagrangian relaxation. Lubin et al. [16] demonstrated the potential for parallel speedup by addressing the bottleneck of parallelizing dual decomposition. Originally proposed by Rockafellar and Wets [20] for stochastic programs with only continuous variables, progressive hedging (PH) has been successfully applied by Listes and Dekker [14], Fan and Liu [6], Watson and Woodruff [26], and many others as a heuristic to solve stochastic mixed-integer programs. To assess the quality of the solutions generated by PH relative to

the optimal solution, Gade et al. [8] presented a lower bounding technique for the PH algorithm and showed that the best possible lower bound obtained from PH is as tight as the lower bound obtained using DD.

The PH algorithm can find high-quality solutions within a reasonable number of iterations, but is not guaranteed to converge to a globally optimal solution in the case of mixed-integer problems. The DD algorithm, on the other hand, will achieve convergence combined with branch and bound but may be slow. This paper combines advantages of both scenario decomposition methods. By transforming PH weights into Lagrangian multipliers as a starting point for DD, the convergence of DD can be sped up considerably.

The remainder of this paper is organized as follows. In Section 2 we describe the PH and DD algorithms, two scenario-based decomposition algorithms for stochastic mixed-integer programs. Our integration approach to transfer information from PH to DD is developed in Section 3. In Section 4, we document the implementation of our integration method and in Section 5, provide experimental results on a set of stochastic server location instances and two stochastic unit commitment instances.

2. Scenario decomposition algorithms for stochastic mixed integer programs

Decomposition methods for stochastic programs generally fall into two groups: stage-based methods and scenario-based methods [21]. The exemplary stage-based decomposition method is the

* Corresponding author. Tel.: +1 515 294 4347.

E-mail address: smryan@iastate.edu (S.M. Ryan).

L-shaped method, or Benders decomposition [25]. Paradigms of scenario-based decomposition include the PH algorithm [20] and the DD algorithm [2]. One advantage of scenario-based decomposition methods over the stage-based ones is their mitigation of the computational difficulty associated with large problem instances by decomposing the problem by scenario and solving the subproblems in parallel. In practical applications, PH can easily be implemented as a “wrapper” for existing software for large-scale implementation of the deterministic scenario problems. In this section, we will discuss these two scenario-based decomposition methods for stochastic mixed-integer programs in detail.

2.1. Two-stage stochastic mixed-integer program

We consider the following two-stage stochastic mixed-integer program:

$$z = \min\{cx + Q(x) : Ax = b, x \in X\}, \quad (1)$$

where $Q(x) = \mathbb{E}_\xi \phi(x, \xi)$ and $\phi(x, \xi) = \min\{q(\xi)y : Wy = h(\xi) - T(\xi)x, y \in Y\}$. Here $c^T \in \mathbb{R}^{n_1}$ and $b \in \mathbb{R}^{m_1}$ are known vectors, while $A \in \mathbb{R}^{m_1 \times n_1}$ and $W \in \mathbb{R}^{m_2 \times n_2}$ are known matrices. The vector ξ is a random variable defined on some probability space $(\mathcal{E}, \mathfrak{S}, P)$ and for each $\xi \in \mathcal{E}$, the vectors $q(\xi)^T \in \mathbb{R}^{n_2}$ and $h(\xi) \in \mathbb{R}^{m_2}$ and the matrix $T(\xi) \in \mathbb{R}^{m_2 \times n_1}$. The sets $X \subseteq \mathbb{R}_+^{n_1}$ and $Y \subseteq \mathbb{R}_+^{n_2}$ denote the mixed-integer requirements on the first-stage and second-stage variables. The decisions are two-stage in the sense that first-stage decisions x have to be taken without full information on some random events while second-stage decisions y are taken after full information is received on the realization of the random vector ξ . The notation \mathbb{E}_ξ denotes expectation with respect to the distribution of ξ .

To avoid complications when computing the integral behind \mathbb{E}_ξ we assume that we have only a finite number of realizations of ξ , known as scenarios $\xi^j, j = 1, \dots, r$, with corresponding probabilities p^j . Then problem (1) can be written as a large-scale deterministic mixed-integer linear program with a block-angular structure called the extensive form of the deterministic equivalent:

$$z = \min \left\{ cx + \sum_{j=1}^r p^j q^j y^j : (x, y^j) \in S^j, j = 1, \dots, r \right\}, \quad (2)$$

where $S^j = \{(x, y^j) : Ax = b, x \in X, Wy^j = h^j - T^j x, y^j \in Y\}$.

The block-angular structure of Eq. (2) enables the decomposition methods to split it into scenario subproblems by introducing copies of the first-stage variables. This idea leads to the so-called scenario formulation of the stochastic program:

$$z = \min \left\{ \sum_{j=1}^r p^j (cx^j + q^j y^j) : (x^j, y^j) \in S^j, \right. \\ \left. j = 1, \dots, r, x^1 = \dots = x^r \right\}. \quad (3)$$

The subproblems are coupled by the non-anticipativity constraints, $x^1 = \dots = x^r$, which force the first-stage decisions to be scenario-independent.

2.2. Dual decomposition

The dual decomposition (DD) algorithm of Carøe and Schultz relaxes the non-anticipativity constraints and uses branch and bound to restore non-anticipativity. DD obtains lower bounds on the optimal value of problem (3) by solving the Lagrangian dual obtained by relaxing the non-anticipativity constraints.

The non-anticipativity requirement of problem (3) can be expressed by several equivalent representations. Lulli and Sen [17]

as well as Lubin and Martin [16] introduce an additional variable x_\cdot and model non-anticipativity as

$$x_j - x_\cdot = 0, \quad j = 1, \dots, r, \quad (4)$$

while Carøe and Schultz represent non-anticipativity by

$$\sum_{j=1}^r H^j x^j = 0, \quad (5)$$

where the matrix $H^j \in \mathbb{R}^{n_1(r-1) \times n_1}$.

Using non-anticipativity representation (4), the Lagrangian relaxation of non-anticipativity constraints may be written as

$$P(\lambda) = \min \left\{ \sum_{j=1}^r [R_j(x^j, y^j, \mu^j) - \mu^j x_\cdot] : (x^j, y^j) \in S^j \right\}, \quad (6)$$

where $R_j(x^j, y^j, \mu^j) = p^j(cx^j + q^j y^j) + \mu^j x_\cdot$ for $j = 1, \dots, r$ and the parameter $(\mu^j)^T \in \mathbb{R}^{n_1}$. The Lagrangian (6) is separable into $P(\mu^1, \dots, \mu^r) = \sum_{j=1}^r P_j(\mu^j)$, where

$$P_j(\mu^j) = \min\{R_j(x^j, y^j, \mu^j) : (x^j, y^j) \in S^j\}, \quad (7)$$

with the condition $\sum_{j=1}^r \mu^j = 0$ required for boundedness of the Lagrangian. The Lagrangian dual is expressed as

$$c_{LD} = \max_{\mu^1, \dots, \mu^r} \left\{ P(\mu^1, \dots, \mu^r) : \sum_{j=1}^r \mu^j = 0 \right\}. \quad (8)$$

The non-anticipativity representation (5), on the other hand, leads to the Lagrangian relaxation in the form

$$D(\lambda) = \min \left\{ \sum_{j=1}^r L_j(x^j, y^j, \lambda) : (x^j, y^j) \in S^j \right\}, \quad (9)$$

where $L_j(x^j, y^j, \lambda) = p^j(cx^j + q^j y^j) + \lambda(H^j x^j)$ for $j = 1, \dots, r$, where the vector $\lambda = (\lambda^1, \dots, \lambda^{r-1})$ and the vector $(\lambda^j)^T \in \mathbb{R}^{n_1}$. The Lagrangian (9) is separable into $D(\lambda) = \sum_{j=1}^r D_j(\lambda)$, where

$$D_j(\lambda) = \min\{L_j(x^j, y^j, \lambda) : (x^j, y^j) \in S^j\}. \quad (10)$$

The Lagrangian dual problem then becomes the problem

$$z_{LD} = \max_{\lambda} D(\lambda). \quad (11)$$

The Lagrangian dual (11) is a convex non-smooth program and can be solved using subgradient methods.

Due to the integer requirements in Eq. (2), a duality gap may occur between the optimal value of the Lagrangian dual (11) and the optimal value of Eq. (2) as described in the proof of Proposition 2 in [2]. Lagrangian dual (11) provides lower bounds on the optimal value of Eq. (2) and the optimal solutions of the Lagrangian relaxation. In general, these first-stage solutions will not coincide unless the duality gap vanishes. The DD algorithm employs a branch and bound procedure that uses Lagrangian relaxation of non-anticipativity constraints as lower bounds [2].

Step 1 Initialization: Set $z^* = \infty$ and let \mathcal{P} consist of problem (2).

Step 2 Termination: If $\mathcal{P} = \emptyset$ and $z^* < \infty$, then x^* with $z^* = cx^* + Q(x^*)$ is optimal.

Step 3 Node selection: Select and delete a problem P from \mathcal{P} , solve its Lagrangian dual (11). If the associated optimal value $z_{LD}(P)$ equals infinity go to *Step 2*.

Step 4 Bounding: If $z_{LD}(P)$ is greater than z^* go to *Step 2*. Otherwise proceed as follows; if the first-stage solutions $x^j, j = 1, \dots, r$, of the subproblems are

- (1) identical, then set $z^* := \min\{z^*, cx^j + Q(x^j)\}$.
- (2) not identical, then compute a suggestion $\hat{x} = \text{Heu}(x^1, \dots, x^r)$ using some heuristic. If \hat{x} is feasible then let $z^* := \min\{z^*, c\hat{x} + Q(\hat{x})\}$. Go to *Step 5*.

Download English Version:

<https://daneshyari.com/en/article/1142093>

Download Persian Version:

<https://daneshyari.com/article/1142093>

[Daneshyari.com](https://daneshyari.com)