Math. Program., Ser. A 106, 203–223 (2006)

Digital Object Identifier (DOI) 10.1007/s10107-005-0592-5

Suvrajeet Sen · Hanif D. Sherali

Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming

Received: September 20, 2002 / Accepted: February 14, 2005
Published online: July 14, 2005 –

Abstract. Decomposition has proved to be one of the more effective tools for the solution of large-scale problems, especially those arising in stochastic programming. A decomposition method with wide applicability is Benders' decomposition, which has been applied to both stochastic programming as well as integer programming problems. However, this method of decomposition relies on convexity of the value function of linear programming subproblems. This paper is devoted to a class of problems in which the second-stage subproblem(s) may impose integer restrictions on some variables. The value function of such integer subproblem(s) is not convex, and new approaches must be designed. In this paper, we discuss alternative decomposition methods in which the second-stage integer subproblems are solved using branch-and-cut methods. One of the main advantages of our decomposition scheme is that Stochastic Mixed-Integer Programming (SMIP) problems can be solved by dividing a large problem into smaller MIP subproblems that can be solved in parallel. This paper lays the foundation for such decomposition methods for two-stage stochastic mixed-integer programs.

Key words. Stochastic Programming – Decomposition – Branch-and-Cut – Mixed-Integer Programming

1. Introduction

This paper continues the line of work initiated in two earlier papers dealing with convexification of stochastic mixed-integer programming (SMIP) problems. One of the papers (Sherali and Fraticelli [2002]) drew upon results from the theory of reformulation-linearization techniques (Sherali and Adams [1990,1994,1999]) to develop a sequence of relaxations for 0–1 mixed-integer second-stage problems, and these were incorporated within a Benders' decomposition algorithm for a two-stage stochastic program having binary first-stage variables. The other motivating paper is that of Sen and Higle [2004] in which a disjunctive decomposition method involving set convexification (D^2-SC) was proposed for SMIP problems. As one might expect, there are definite connections between these approaches, although their development, as well as the algorithmic schemes, are different. The current paper is intended to enhance these approaches by adding a major piece of the "algorithmic puzzle".

In essence, this paper revolves around the design of branch-and-cut (BAC) algorithms for SMIP problems. BAC algorithms provide one of the more successful approaches for deterministic mixed-integer programming (MIP). However, their extension to SMIP problems is far from obvious. The main challenges arise from the need to combine

S. Sen: Department of System and Industrial Engineering, University of Arizona, PO Box 210020, Tucson, AZ 85721, USA. e-mail: sen@sie.arizona.edu

H.D. Sherali: Grado Department of ISE, Virginia Polytechnic Institute and State University Blacksburg, VA 24061, USA.

decomposition-based methods that have worked well in the Stochastic Programming arena (see Ruszczyński [1997]) with BAC methods that have worked well for MIP problems. Preliminary success with decomposition of SMIP problems has already been reported in Ntaimo and Sen [2003], where the authors report solving large SMIP problems, some of which have deterministic equivalents containing over a million binary variables, and over a hundred thousand constraints. As reported in the above paper, direct methods based on using branch-and-bound (BAB) and cutting planes for the deterministic equivalent problem are woefully inadequate to the task of solving these very large-scale MIP problems. In contrast, computational results using decomposition methods, such as D^2-SC, demonstrate that they are very effective for SMIP problems (Ntaimo and Sen [2003]).

We begin this paper by exploring a method that combines decomposition with a BAC method in a deterministic setting. This (deterministic) setting allows some simplifications that permit a few algorithmic liberties. The development not only provides the first taste of the algorithmic issues for decomposition-based BAC (D-BAC) algorithms, but also sets the stage for a BAC algorithm in the stochastic programming setting. The D-BAC algorithm will guide us towards Disjunctive Decomposition (D^2) algorithms to accommodate SMIP problems in which both first and second stages have integer variables. These extensions provide powerful additions to the algorithmic toolkit for SMIP.

2. Decomposing deterministic mixed-integer programs

This section is devoted mainly to a deterministic binary mixed-integer programming problem. This problem also provides the vehicle that will be used to set the stage for designing algorithms for the SMIP problem.

Consider the problem:

$$\operatorname{Min} c^\top x + g^\top y \tag{1.1}$$

$$Tx + Wy \geq r \tag{1.2}$$

$$x \in X \cap \mathcal{B},\ y \in Y \cap \mathcal{B}' \tag{1.3}$$

where $\mathcal{B}$ ($\mathcal{B}'$) denotes the set of binary (mixed 0–1) vectors, and where $X \subseteq \Re^{n_1}$ and Y are nonempty polyhedra. The set X is represented as $\{x \mid Ax \geq b, x \geq 0\}$, where the constraints $x_j \leq 1, \forall j$ are included in the linear inequalities $Ax \geq b$. Similarly, we assume that Y is bounded and represent it as follows:

$$Y = \{y \mid Dy \geq f, y \geq 0, y_j \text{ binary for } j \in J_2\}. \tag{1.4}$$

According to our notation, inequalities of the form $y_j \leq 1, \forall j \in J_2$ are also included in the constraints $Dy \geq f$, and the index set J_2 provides the subset of the second-stage variables that are restricted to be binary.

2.1. A decomposition-based cutting plane approach

This subsection is based on ideas presented in Sherali and Fraticelli [2002]. One of the key observations made by these authors is that (1) is equivalent to the following problem:

$$\begin{aligned} &\operatorname{Min} c^\top x + g^\top y && (2.1)\\ &\quad (x, y) \in \operatorname{conv}\{(x, y) \mid Tx + Wy \geq r, 0 \leq x \leq e, y \in Y \cap \mathcal{B}'\} && (2.2)\\ &\quad x \in X \cap \mathcal{B}, && (2.3) \end{aligned}$$

where e is a vector whose elements are all 1.

Note that (2) is clearly a relaxation of (1). Moreover, if $(\bar{x}, \bar{y})$ solves (2), where $\bar{y}$ is an extreme point of the linear program resulting from (2) with x fixed at $\bar{x}$, then in particular, $\bar{y} \in \mathcal{B}'$, since the restriction $x = \bar{x}$ is facial with respect to (2.2), and (2.2) has $y \in \mathcal{B}'$ for all its extreme points. Consequently, $(\bar{x}, \bar{y})$ must be feasible, and therefore, optimal for (1). Hence, we have that (1) and (2) are equivalent.

Based on this observation, Sherali and Fraticelli [2002] developed an approach akin to Benders' decomposition, wherein the subproblem solved at any iteration k, with x fixed at $x^k \in X \cap \mathcal{B}$, as given by:

$$\operatorname{Min}\{g^\top y \mid Wy \geq r - Tx^k, y \in Y \cap \mathcal{B}'\}, \tag{3}$$

was solved using a finitely convergent cutting plane algorithm. In particular, the cutting planes used in solving (3) were assured to be actually valid for (2.2) in the (x, y) space. Hence, while x was fixed at x^k when solving the foregoing subproblem, these cuts could be reused at any subsequent iteration $k + \tau$, $\tau \geq 1$, simply by fixing $x = x^{k+\tau}$ in these cuts.

Accordingly, suppose that this subproblem is solved by a cutting plane method in which the accumulated additional constraints generated thus far, which are valid for (2.2), are of the form

$$G_k x + H_k y \geq h_k. \tag{4}$$

Appending these cuts to the linear relaxation of (3) with x fixed at x^k, yields the following LP relaxation, which, say, ultimately solves the subproblem (3).

$$\begin{aligned} &\operatorname{Min}\ g^\top y && (5.1)\\ &\quad Wy \ \geq\ r - Tx^k && (5.2)\\ &\quad H_k y \ \geq\ h_k - G_k x^k && (5.3)\\ &\quad Dy \ \geq\ f && (5.4)\\ &\quad y \ \geq\ 0. && (5.5) \end{aligned}$$

In order to represent (5) in a compact form, we append the sub-matrices associated with the x variables in rows (5.2)–(5.4) to form the matrix T_k. Similarly, the sub-matrices associated with the y variables in (5.2)-(5.4) are taken to form a matrix W_k, and the right-hand-side vector of rows (5.2)–(5.4) is recorded as the vector r_k. Hence, LP (5) can be rewritten in a compact form as:

$$\text{Min}\, g^\top y \tag{6.1}$$
$$W_k y \;\geq\; r_k - T_k x^k \tag{6.2}$$
$$y \;\geq\; 0. \tag{6.3}$$

Letting θ_k denote a vector of optimal dual multipliers associated with (6.2), Sherali and Fraticelli [2002] derive the following Benders' cut that can be added to the master program, where η represents the second-stage value function.

$$\eta \geq \theta_k^\top (r_k - T_k x). \tag{7}$$

Then the master program has the following familiar form:

$$\underset{x \in X \cap \mathcal{B}}{\text{Min}} \; \{c^\top x + \eta \mid \eta \geq \theta_t^\top (r_t - T_t x), \quad \forall\, t = 1, \ldots, k\}.$$

It is important to reiterate that since the inequalities of (4) are valid in the space of both x and y variables, they can be used in all subsequent iterations. Hence, the subproblem solved during iteration $k+1$ can begin by using the inequalities obtained through the first k iterations, and additional globally valid inequalities may be generated during iteration $k+1$ and beyond. Finally, if the cuts (4) used for solving (3) have the property that they ultimately construct the necessary facets of the convex hull of the set described by (2.2), then finite convergence of the resulting algorithm is immediate. There are various families of valid inequalities that satisfy this property (see Sherali and Adams [1990,1994,1999], Lovász and Schrijver [Lovász and Schrijver (1991)], and Balas Ceria and Cornuéjols [1993]).

2.2. A decomposition-based BAC approach (D-BAC)

The results of this subsection extend the approach of the foregoing section to solving (1) for the case in which the second-stage mixed-binary optimization problem is solved using BAC. Here, the term decomposition is to be interpreted in the sense that the integer restrictions on the x and y variables are treated separately. That is, the variables may appear in both the master as well as subproblems simultaneously; however, the integer restrictions on both these variables (x and y) are not imposed simultaneously in either the master or subproblems.

Suppose that in a BAC approach for solving (3), where all cuts are valid for (2.2) in the (x, y) space, we obtain an optimal solution at some node, denoted *, of the branch-and-bound tree. Let us denote the index sets of fixed variables at node * by $J_{2*}^- = \{j \in J_2 \mid y_j \equiv 0\}$, and $J_{2*}^+ = \{j \in J_2 \mid y_j \equiv 1\}$. Then the LP problem for node * in a notation analogous to (6) is:

$$\text{Min}\;\; g^\top y \tag{8.1}$$
$$W_k y \;\geq\; r_k - T_k x^k \tag{8.2}$$
$$y \;\geq 0 \tag{8.3}$$
$$-y_j \geq 0,\, j \in J_{2*}^-,\;\; y_j \geq 1,\, j \in J_{2*}^+. \tag{8.4}$$

Proposition 1. *Let θ_k denote a vector of optimal dual multipliers associated with (8.2). Moreover, let ψ_{kj}^- and ψ_{kj}^+ denote optimal dual multipliers for the bounding constraints associated with J_{2*}^- and J_{2*}^+, respectively, in (8.4). Then the following inequality*

$$\eta \geq \theta_k^\top (r_k - T_k x) + \sum_{j \in J_{2*}^+} \psi_{kj}^+ y_j - \sum_{j \in J_{2*}^-} \psi_{kj}^- y_j \tag{9}$$

provides a lower bounding function on the value of $g^\top y$ over (x, y) feasible to (2.2).

Proof. Since $\theta_k \geq 0$, and $T_k x + W_k y \geq r_k$ is valid for (2.2), we have $0 \geq \theta_k^\top [r_k - T_k x - W_k y]$. Hence,

$$\eta \geq g^\top y + \theta_k^\top [r_k - T_k x - W_k y] = \theta_k^\top [r_k - T_k x] + [g^\top - \theta_k^\top W_k] y. \tag{10}$$

Furthermore, θ_k is dual feasible for (8), and $y \geq 0$. Hence,

$$[g^\top - \theta_k^\top W_k] y \geq \sum_{j \in J_{2*}^+} \psi_{kj}^+ y_j - \sum_{j \in J_{2*}^-} \psi_{kj}^- y_j.$$

Substituting this inequality in (10) yields the desired result. □

Although we have presented Proposition 1 in such a manner as to add only one cut, it is possible to add more cuts to obtain a stronger approximation. Note that by incorporating an artificial variable column (with an arbitarily high cost) in the subproblem, we can assume that every node q of the BAC tree is associated with a feasible LP, and that nodes are fathomed when their LP lower bounds exceed the best upper bound obtained. If θ_{kq}, ψ_{kjq}^+, ψ_{kjq}^- denote the dual multipliers with each fathomed node q, then one can add as many cuts as there are nodes in the BAC tree. That is, the following inequalities may be added to the master program.

$$\eta \geq \theta_{kq}^\top (r_k - T_k x) + \sum_{j \in J_{2q}^+} \psi_{kjq}^+ y_j - \sum_{j \in J_{2q}^-} \psi_{kjq}^- y_j, \quad \forall q. \tag{11}$$

The arguments supporting the validity of (9) also support the validity of (11).

Proposition 2. *Consider the following partial relaxed master program in which the binary restrictions on the y variables are enforced.*

$$\text{Min}\{c^\top x + \eta \mid (\eta, x, y) \text{ satisfy (11)}, y_j \text{ binary } \forall j \in J_2\}. \tag{12}$$

(Note that (11) requires the inequalities based on all nodes of the BAC tree to be included.) Then, for $x = x^k$ (fixed), the optimal value of η in (12) is the same as the value of the subproblem (3).

Proof. Note that all inequalities (11) indexed by q are valid lower bounding inequalities, and so, (12) (with x fixed at x^k) is a relaxation of (3). Moreover, the inequality for the node that yields the optimal value is included in the description of (12). Furthermore, for any inequality q in (11), when we fix $x = x^k$ along with $y_j = 1, \forall j \in J_{2q}^+$, and $y_j = 0$ $\forall j \in J_{2q}^-$, the right-hand-side of (11) yields (by duality) the value of the corresponding nodal subproblem analogous to (8) in the enumeration tree. Since any binary solution $(\hat{y}_j, j \in J_2)$ relates to precisely one terminal node q in this enumeration tree for which $\hat{y}_j = 1, \forall j \in J_{2q}^+$, and $\hat{y}_j = 0, \forall j \in J_{2q}^-$, the result follows. □

This result is a generalization of the corresponding result for the case in which the second-stage problem is a linear program (with all second-stage variables being continuous). In that case, fixing $x = x^k$ in the standard Benders' cut that was generated corresponding to this first-stage solution yields the same value of η as the objective value of the associated LP subproblem. Because this subproblem is an LP, one may interpret this occurrence as an application of Proposition 2 in which the only node that is needed is the "root node." More generally, however, when the subproblem is an MIP, as in the present case, all nodes of the BAC tree are necessary to recover the optimal value of the underlying subproblem as portended by the proposition.

Notwithstanding this result, one may wish to relax the requirement that the second-stage binary variables (in y) be restricted to be integer-valued within the master program, and yet be able to recover the corresponding subproblem value when x is fixed at x^k. This would result in a more workable and tractable, albeit weaker, master program. Indeed, this can be achieved by incorporating the following additional valid inequality within the master program whenever one solves the subproblem to optimality (as we have in this section). Let

$$I_k = \{i \mid x_i^k = 1\}, \quad Z_k = \{1, \dots, n_1\} - I_k.$$

Next, define the linear function

$$\delta_k(x) = |I_k| - [\sum_{i \in I_k} x_i - \sum_{i \in Z_k} x_i]. \tag{13}$$

It is easily seen that when $x = x^k$ (assumed binary), $\delta_k(x) = 0$; whereas, for all other binary $x \neq x^k$, at least one of the variables must switch "states." Hence, for $x \neq x^k$, we have

$$[\sum_{i \in I_k} x_i - \sum_{i \in Z_k} x_i] \leq |I_k| - 1, \text{ i.e., } \delta_k(x) \geq 1.$$

Now, suppose that a lower bound on the second-stage, denoted ℓ, is available. Furthermore, let $\eta(x^k)$ denote the optimal value of the subproblem, given x^k. Then the following inequality is valid and may be included in the master program.

$$\eta \geq \eta(x^k) - \delta_k(x)[\eta(x^k) - \ell]. \tag{14}$$

This is essentially the "optimality" cut of Laporte and Louveaux [1993]. To verify its validity, first observe that when $x = x^k$, the second term in (14) vanishes, and hence, the master program recovers the value of the corresponding subproblem. On the other hand, if $x \neq x^k$, then,

$$\delta_k(x)[\eta(x^k) - \ell] \geq [\eta(x^k) - \ell].$$

Hence, for all $x \neq x^k$, the right-hand-side of (14) obeys

$$\eta(x^k) - \delta_k(x)[\eta(x^k) - \ell] \leq \eta(x^k) - \eta(x^k) + \ell = \ell.$$

Hence, imposing (14) does not delete any viable first-stage solution. Moreover, since (14) itself asserts the desired inequality for the purpose of finite convergence of Benders' algorithm, namely, that for $x = x^k$, $\eta \geq \eta(x^k)$ in the master program, we need not explicitly impose either the integrality restrictions on any of the binary components of y (as in Proposition 2), or all the terminal node inequalities (11). Thus, while there is a price to be paid in solving the second-stage problem to optimality, including (14) in addition to (11) may provide stronger relaxations in the master program. Furthermore, this construct obtains finite convergence of Benders' decomposition as in Sherali and Fraticelli [2002].

Remark 1. While implementing (11), if we wish to dispense with the inclusion of y in the master program, then we can derive a projection that provides an inequality for which the coefficients of y vanish. To do so, one would have to select nonnegative multipliers on $y_j \geq 0$, $-y_j \geq -1$, $j \in J_2$, together with a multiplier on (11) in such a way that the aggregated coefficient vector for y is zero.

Remark 2. In the foregoing analysis, we have assumed that by adding appropriate artificial variables, each nodal subproblem in the tree is feasible. It is instructive to note that (13) provides the facility of generating feasibility cuts in the absence of the above assumption. Specifically, if some node yields an infeasible restriction, then in lieu of (11), one may add the constraint

$$\sum_{j \in J_{2q}^+} (1 - y_j) + \sum_{j \in J_{2q}^-} y_j \geq 1 - \delta_k(x). \tag{15}$$

To see that (15) is valid, note that whenever $x = x^k$, $\delta_k(x^k) = 0$, and then (15) asserts that at least one of the binary variables for y_j, $j \in J_2$, must assume a value different from the restrictions imposed at node q of the branch-and-bound tree. On the other hand, when $x \neq x^k$, we have $\delta_k(x) \geq 1$, and then (15) is simply redundant. Moreover, by incorporating (15) within (11) for all infeasible nodes q, the assertion of Proposition 2 continues to hold true.

2.3. *An illustration of the D-BAC algorithm*

Consider the following problem:

$$\begin{aligned} \text{Min } & -x_1 - 2y_1 + 4y_2 \\ & -4x_1 - 3y_1 + y_2 \geq -6 \\ & (x_1, y_1) \text{ binary}, \ y_2 \geq 0. \end{aligned}$$

We will refer to x_1 as the first-stage decision variable, and y will be designated a second-stage decision vector. It is easily seen that for binary values of y_1, and nonnegative y_2, a lower bound on the expression $-2y_1 + 4y_2$ is -2; that is $\eta \geq -2$. Hence, we may initialize the process with the following master program.

$$\begin{aligned} \text{Min } & -x_1 + \eta \\ & \eta \geq -2 \\ & x_1 \text{ binary}. \end{aligned}$$

The optimal value of the above problem is -3, and this value also provides a lower bound on the optimal value of the original problem. The optimal solution to this approximation is $x_1 = 1$, $\eta = -2$. Using $x_1 = 1$, we formulate the second-stage problem as follows.

$$\begin{aligned} \text{Min } & -2y_1 + 4y_2 \\ & -3y_1 + y_2 \geq -2 \\ & y_1 \text{ binary}, \;\; y_2 \geq 0. \end{aligned}$$

The LP relaxation at the root node provides a fractional solution $(2/3, 0)$. Suppose that we solve this problem using a branch-and-bound scheme. The nodes of the tree are analyzed below.

Node 1 $(y_1 = 1)$: We solve the following LP:

$$\begin{aligned} \text{Min } & -2y_1 + 4y_2 \\ & -3y_1 + y_2 \geq -2 \\ & -y_1 \geq -1 \\ & \;\; y_1 \geq 1 \\ & y_1, y_2 \geq 0. \end{aligned}$$

Relating this problem to (8), note that the first two rows form the matrix W_k of (8.2), and the third constraint above is a lower bound of the form in (8.4). Solving this LP yields $(y_1, y_2) = (1, 1)$ and dual multipliers $\theta = (4, 0)$, and $\psi_1^+ = 10$. Accordingly, (11) yields the following approximation of the second-stage value function.

$$\eta \geq 4(-6 + 4x_1) + 10y_1.$$

Node 2 $(y_1 = 0)$: The LP for this node is given by:

$$\begin{aligned} \text{Min } & -2y_1 + 4y_2 \\ & -3y_1 + y_2 \geq -2 \\ & -y_1 \geq -1 \\ & -y_1 \geq 0 \\ & y_1, y_2 \geq 0. \end{aligned}$$

Upon solving this LP, we have $(y_1, y_2) = (0, 0)$, and the dual multipliers are $\theta = (0, 0)$, and $\psi_1^- = 2$. Applying (11), we obtain

$$\eta \geq -2y_1.$$

The upper bound for the original problem at this iteration is -1 (obtained at node 2). Moreover, an inequality of type (14) may be obtained by letting x^k correspond to $x_1 = 1$, $\eta(x^k) = 0$, $\ell = -2$, and using (13) to define $\delta_k(x) = 1 - x_1$. The resulting inequality is

$$\eta \geq -2(1 - x_1). \tag{16}$$

Using inequalities of type (11) and (14) (and omitting the dominated inequality $\eta \geq -2$), the updated master program is as follows.

$$\begin{aligned} \text{Min } & -x_1 + \eta \\ & -2x_1 + \eta \geq -2 \\ & -16x_1 + \eta - 10y_1 \geq -24 \\ & \eta + 2y_1 \geq 0 \\ & x_1 \text{ binary}, \ 0 \leq y_1 \leq 1. \end{aligned}$$

Having updated the master program, we have now completed one iteration. At this point in the algorithm, the upper bound is -1, and the lower bound is -3.

Starting the next iteration, we solve the updated master problem. The solution to this problem yields $(x_1, \eta, y_1) = (0, -2, 1)$, and the updated lower bound is -2. Fixing $x_1 = 0$, the LP relaxation of the second-stage problem yields $(y_1, y_2) = (1, 0)$. The resulting upper bound is therefore -2, which equals the lower bound. Hence, the method stops with an optimal first-stage solution, which is $x_1 = 0$.

Note that because of our use of a decomposition scheme, no individual problem (master or subproblem) has more than one integer variable, although the original problem has two integer variables. Thus, similar to Benders' decomposition for continuous second-stage decisions, our D-BAC algorithm solves MIP problems of the above type by solving a sequence of easier mixed-integer programs.

3. Decomposition for stochastic mixed-integer programming

We now turn our attention to a study of SMIPs in which, as before, both the first and a subset of the second-stage variables are required to satisfy integer restrictions. We state the problem as follows.

$$\underset{x \in X \cap \mathcal{B}}{\text{Min}} \ c^\top x + E[f(x, \tilde{\omega})],$$

where X, and $\mathcal{B}$ are sets defined in (1), $\tilde{\omega}$ is a random variable defined on a probability space $(\Omega, \mathcal{A}, \mathcal{P})$, and for any realization ω of $\tilde{\omega}$,

$$\begin{aligned} f(x, \omega) = \text{Min } & g^\top y \\ Wy \ & \geq \ r(\omega) - T(\omega)x \\ & y \geq 0, \ y_j \text{ binary}, \ j \in J_2. \end{aligned}$$

Within the stochastic programming literature, a realization of $\tilde{\omega}$ is known as a "scenario". As such, the second-stage problem is often referred to as a "scenario subproblem." In our development, we assume that the right-hand-side vector $r(\tilde{\omega})$, and the technology matrix $T(\tilde{\omega})$ are governed by random variables, whereas, the other data elements are deterministic.

Caroe [1998] provided the first systematic application of integer programming algorithms to SMIP problems. The approach studied by Caroe views the SMIP problem as a large-scale deterministic equivalent MIP, for which disjunctive cuts are generated to solve this problem. It is not difficult to see that the dual block-angular structure of the MIP is retained by the addition of cuts, and hence, one may adopt a decomposition method to solve the LP relaxation of the deterministic equivalent. Subsequently, Ahmed, Tawarmalani, and Sahinidis [2004], Sen and Higle [2004], and Sherali and Fraticelli [2002] have used global optimization and integer programming methods for solving SMIP problems. Characterizations of stability of these problems are given in Schultz [1995]. For multi-stage SMIP problems, Caroe and Schultz [1999] discuss a decomposition method based on Lagrangian relaxation, and Lulli and Sen [2004] present a branch-and-price method for multi-stage SMIP problems. The latter paper also reports computational results for multi-stage batch sizing problems, with and without backlogging. For an overview of characterizations and algorithms for SMIP problems, we refer the reader to surveys by Klein Haneveld and van der Vlerk [1999], Schultz [2003], and Sen, Higle, and Ntaimo [2003].

In our approach, we will decompose the SMIP problem, and approximate the value function of MIP subproblems. In addition, our method is applicable to a more general class of problems than Caroe's. In particular, our approach is applicable to cases where the second-stage problem includes general integers (not simply 0–1), and the first-stage decisions are required to be extreme points of X (see Proposition 3, and Section 3.2). Of course, the class of problems stated above satisfy these requirements. The choice to restrict our discussion to 0–1 problems is motivated by the need for clarity and consistency with the previous section.

We begin this section by examining what it takes to make the transition from deterministic MIPs to stochastic MIPs. Simply put, it is a matter of scalability. The manner in which an algorithm accommodates the presence of a large number of alternative scenarios determines its effectiveness for solving SMIP problems. While analytical approaches simply rely on the fact that there are only finitely many scenarios in the SMIP problem, realistic models often result in a relatively large number of scenarios. Recognizing this, let us first discuss the merits of obtaining an optimal solution for each scenario subproblem for a given first-stage solution x. Since these subproblems are generally NP-hard, the decomposition method may get bogged down in attempts to solve subproblems, even while the particular first-stage decision x may not be near a reasonably small neighborhood of an optimal solution. In essence, our view is that the algorithm should allow partial (i.e. suboptimal) solves of the MIP subproblems, but as iterations proceed, the method should learn enough about the structure of the MIP subproblems, so that ultimately, the "partial solves" begin to yield optimal solutions. This can be accomplished via a sequential convexification process in which only a small subset of facets are generated during any iteration.

Scalability of an algorithm is not only determined by the computational effort per iteration, but also by its memory requirements reflected through the size of each optimization problem solved during the algorithmic process. This is one of the main motivations for the C^3 theorem in Sen and Higle [2004]. Because of the common cut coefficients, approximations of the second-stage can be generated without storing cuts separately for each scenario. In contrast, the cuts presented in §2.1 are intended to be recorded explicitly for each scenario, and although they may be reused for different first-stage decisions, there is the potential of dealing with a very memory-intensive algorithm. In making the transition from deterministic to stochastic mixed-integer programming problems, care must be taken to design algorithms that acknowledge the potential for a large number of scenarios.

We note that the BAC approach for subproblems (see §2.2) can help reduce the number of valid inequalities recorded for each scenario. However, the augmented cuts of §2.2 introduce other complications for SMIPs. For instance, a naive extension of the method of §2.2 to SMIP problems would lead us to include as many copies of second-stage variables as there are scenarios in the SMIP. Notwithstanding the fact that these are treated as continuous variables in the master problem, the size of the resulting MIP can quickly get out of hand for all but the most modest SMIP problems. We therefore adopt the D^2 approach whereby common cut coefficients allow us to curtail the explosive growth of cuts, without sacrificing asymptotic accuracy (Sen and Higle [2004]). The remainder of this section is devoted to the incorporation of BAC methods for the second-stage problem in a D^2 algorithm.

Consider a partial branch-and-bound tree generated during a "partial solve" of the second-stage problem. Let $Q(\omega)$ denote the set of terminal nodes of the tree that have been generated for the subproblem associated with scenario ω. As in §2.2, we will assume that all nodes of the branch-and-bound tree are associated with feasible LP relaxations, and that nodes are fathomed when the LP lower bound exceeds the best available upper bound. This may be accomplished by introducing artificial variables, if necessary. Our strategy revolves around using the dual problem associated with the LP relaxation (one for each node), and then stating a disjunction that will provide a valid inequality for the first-stage problem. When compared with the inequalities obtained previously in (9), these cuts involve only the first-stage variables (x). (The reader may also refer to Remark 1 for projecting (9) onto the space of first-stage variables x.)

In the following, we use k as the index of iterations, where at each iteration, the second-stage subproblems are solved to some degree of accuracy. For any node $q \in Q(\omega)$, let $z_{q\ell}(\omega)$ and $z_{qh}(\omega)$ denote vectors whose elements are used to define lower and upper bounds, respectively, on the second-stage (integer) variables. In some cases, an element $(z_{qh})_j$ may be $+\infty$, and in this case, the associated constraint may be ignored, implying that the associated dual multiplier is fixed at 0. In any event, the LP relaxation for node q may be written as:

$$\begin{aligned} \text{Min} \quad & g^\top y \\ & W_k y \ \geq \ r_k(\omega) - T_k(\omega) x^k \\ & y \ \geq 0 \\ & y \geq z_{q\ell}(\omega), \quad -y \geq -z_{qh}(\omega), \end{aligned}$$

and, the corresponding dual LP is:

$$\begin{aligned} \text{Max} \quad & \theta_q(\omega)^\top [r_k(\omega) - T_k(\omega) x^k] + \psi_{q\ell}(\omega)^\top z_{q\ell}(\omega) - \psi_{qh}(\omega)^\top z_{qh}(\omega) \\ & \theta_q(\omega)^\top W_k + \psi_{q\ell}(\omega)^\top - \psi_{qh}(\omega)^\top \ \leq \ g^\top \\ & \theta_q(\omega) \geq 0, \psi_{q\ell}(\omega) \geq 0, \psi_{qh}(\omega) \geq 0, \end{aligned}$$

where the vectors $\psi_{q\ell}(\omega)$, and $\psi_{qh}(\omega)$ are appropriately dimensioned vectors.

We now turn our attention to approximating the value function of the second-stage MIP. As noted in Blair and Jeroslow [1982], and subsequently by Blair [1995], IP and MIP value functions are complicated objects; however, the branch-and-bound tree, together with the LP relaxations at these nodes, provide important information that can be used to approximate MIP value functions. The main observation that we use in this development is that the branch-and-bound tree embodies a disjunction, and when coupled with piecewise linear value functions of the LP relaxations for each node, we obtain a disjunctive description of an approximation to the MIP value function. By using the disjunctive cut principle, we will then obtain linear inequalities (cuts) that can be used to build value function approximations. In order to do so, we assume that we have a lower bound ℓ such that $f(x, \tilde{\omega}) \geq \ell$ (almost surely) for all x. Without loss of generality, this bound may be assumed to be 0.

Consider a node $q \in Q(\omega)$ and let $(\theta_q^k(\omega), \psi_{q\ell}^k(\omega), \psi_{qh}^k(\omega))$ denote optimal dual multipliers for node q. Then a lower bounding function may be obtained by requiring that $x \in X$ and that the following disjunction holds true.

$$\begin{aligned}\eta \geq \theta_q^k(\omega)^\top[r_k(\omega) - T_k(\omega)x] + \psi_{q\ell}^k(\omega)^\top z_{q\ell}(\omega) \\ -\psi_{qh}^k(\omega)^\top z_{qh}(\omega) \text{ for at least one } q \in Q(\omega).\end{aligned} \tag{17}$$

Note that each inequality in (17) corresponds to a second-stage value function approximation that is valid only when the restrictions (on the y-variables) associated with node $q \in Q(\omega)$ hold true. Since any optimal solution of the second-stage must be associated with at least one of the nodes $q \in Q(\omega)$, the disjunction (17) is valid.

It is instructive to examine the relationship between the disjunction in (17) directly and the set of inequalities (11). In stating (17), we first undertake an aggregation process using the dual multipliers to obtain a conditionally valid inequality for each node, given the corresponding integer restrictions. This leads to disjunction (17) to which we can now apply the disjunctive cut principle (Balas [1979]) to generate a valid inequality. In contrast, (11) is formed by directly developing a valid inequality for (2.2) in the (x, y) space based on each node subproblem, and as suggested in Remark 1, an aggregation process could be used subsequently to project these inequalities onto the space of the x variables. Thus the aggregation precedes the formation of valid inequalities in (17), whereas, the reverse would be true if (11) were to be projected onto the space of the x variables.

Returning to the structure of the inequalities in (17), note that for each $q \in Q(\omega)$ we can associate an epigraph

$$E_q^k(\omega) = \{(\eta, x) \mid \eta \geq \bar{\nu}_q^k(\omega) - \bar{\gamma}_q^k(\omega)^\top x,\ Ax \geq b,\ x \geq 0,\ \eta \geq 0\},$$

where,

$$\bar{\nu}_q^k(\omega) = \theta_q^k(\omega)^\top r_k(\omega) + \psi_{q\ell}^k(\omega)^\top z_{q\ell}(\omega) - \psi_{qh}^k(\omega)^\top z_{qh}(\omega),$$

and

$$\bar{\gamma}_q^k(\omega) = T_k(\omega)^\top \theta_q^k(\omega).$$

In the above statement, we have restricted the epigraph associated with each inequality in (17) to the domain $x \in X$, and $\eta \geq 0$. The validity of (17) implies that the epigraph of the subproblem (MIP) value function for outcome $\omega \in \Omega$ is a subset of the following disjunctive set

$$\Pi_k(\omega) = \{(\eta, x) \in \cup_{q \in Q(\omega)} E_q^k(\omega)\}.$$

We will use a convexification of this set to derive lower bounding functions for use in the master program.

Starting with the work of Balas [1979], disjunctive programming has provided a basis for convexifying disjunctive sets of the form given above. Facets of the convex hull of $\Pi_k(\omega)$ may be represented in the form

$$\sigma_0^k(\omega)\eta + \sum_j \sigma_j^k(\omega)x_j \geq \zeta^k(\omega),$$

where the vector $(\sigma_0^k(\omega), \sigma_1^k(\omega), \ldots, \sigma_{n_1}^k(\omega), \zeta^k(\omega))$ is an extreme point of the following polyhedral set

$$\begin{aligned}
\Pi_k^\dagger(\omega) \;&=\; \{\sigma_0(\omega) \in \Re, \sigma(\omega) \in \Re^{n_1}, \zeta(\omega) \in \Re \mid \forall\, q \in Q(\omega),\\
&\qquad\quad \exists\, \tau_q(\omega) \geq 0, \quad \tau_{0q}(\omega) \in \Re_+ \text{ s.t.}\\
\sigma_0(\omega) \;&\geq\; \tau_{0q}(\omega), \quad \forall q \in Q(\omega)\\
\sum_{q \in Q(\omega)} \tau_{0q}(\omega) \;&=\; 1\\
\sigma_j(\omega) \;&\geq\; \tau_q(\omega)^\top A_j \;+\; \tau_{0q}(\omega)\bar{\gamma}_{qj}^k(\omega), \quad \forall q \in Q(\omega), j = 1, \ldots, n_1\\
\zeta(\omega) \;&\leq\; \tau_q(\omega)^\top b \;+\; \tau_{0q}(\omega)\bar{\nu}_q^k(\omega), \quad \forall q \in Q(\omega)\\
\tau_q(\omega) \;&\geq\; 0, \;\; \tau_{0q}(\omega) \geq 0, \quad \forall q \in Q(\omega)\}.
\end{aligned} \tag{18}$$

This polyhedron is derived by aggregating the first two (sets of) constraints defining each $E_q^k(\omega)$ by using nonnegative mutlipliers τ_{0q}, and τ_q respectively, and then applying the disjunctive cut principle, together with the normalizing constraint $\sum_{q \in Q(\omega)} \tau_{0q}(\omega) = 1$. There is a one-one correspondence between facets of the convex hull of $\Pi_k(\omega)$, and the extreme points of $\Pi_k^\dagger(\omega)$. We refer to the latter as the epi-reverse polar since it represents the reverse polar (see Balas [1979]) of the union of epigraphs.

Let η^k denote the lower bound on the expectation, and x^k the first-stage solution resulting from the master program at iteration k. Corresponding to the value η^k, assume that we also have outcomes $\eta^k(\omega)$, $\omega \in \Omega$, such that $\eta^k = \sum_{\omega \in \Omega} p(\omega)\eta^k(\omega)$. Subsequently, we will discuss how these quantities may be obtained (see (22)). Initially, we use $\eta^1 = \eta^1(\omega) = 0$, the assumed lower bound. Now, for each outcome $\omega \in \Omega$, we propose to identify a facet of the convex hull of $\Pi_k(\omega)$ by solving the following LP.

$$\text{Max}\{-\eta^k(\omega)\sigma_0(\omega) - \sum_j x_j^k \sigma_j(\omega) + \zeta(\omega) \mid (\sigma_0(\omega), \sigma(\omega), \zeta(\omega)) \in \Pi_k^\dagger(\omega)\}. \tag{19}$$

Denoting an optimal solution to (19) by $(\sigma_0^k(\omega), \sigma^k(\omega), \zeta^k(\omega))$, a disjunctive cut that provides a lower bound on the MIP subproblem value function can then be generated as

$$\sigma_0^k(\omega)\eta(\omega) + \sum_j \sigma_j^k(\omega)x_j \geq \zeta^k(\omega). \tag{20}$$

Note that the conditions in (18) imply that $\sigma_0(\omega) \geq \mathrm{Max}_q\, \tau_{0q}(\omega) > 0$. Hence, the epi-reverse polar only allows those facets (of the convex hull of $\Pi_k(\omega)$) that have a positive coefficient for the variable η.

The "optimality cut" to be included in the first-stage master problem at iteration k is therefore given by

$$\eta \geq E\left[\frac{\zeta^k(\tilde{\omega})}{\sigma_0^k(\tilde{\omega})}\right] - E\left[\frac{\sigma^k(\tilde{\omega})}{\sigma_0^k(\tilde{\omega})}\right]^\top x. \tag{21.k}$$

It is obvious that one can also devise a multi-cut method using (20) for each outcome ω (see Birge and Louveaux [1997]). Note that during the solution of the resulting master program at iteration $k+1$, there must exist at least one inequality from the list of cuts (21.1), ... , (21.k) that is tight at the optimal solution of this master program. If η^{k+1} denotes the optimal value of η resulting from this master program, then there exists an index $t \leq k$ such that

$$\eta^{k+1}(\omega) = \left[\frac{\zeta^t(\omega)}{\sigma_0^t(\omega)}\right] - \left[\frac{\sigma^t(\omega)}{\sigma_0^t(\omega)}\right]^\top x^{k+1}, \quad \forall \omega \in \Omega. \tag{22}$$

These quantities may be used in (19) for iteration $k+1$.

Remark 3. It is important to draw the distinction between the disjunctive cuts that have appeared in the MIP literature, and the cut proposed in (21.k). Disjunctive cuts in the MIP literature are intended to provide tight relaxations of the set of feasible integer points. The D^2 algorithm with set convexification (D^2-SC, Sen and Higle [2004], and Sen, Higle, and Ntaimo [2002]) provides a sequential approach for tightening second-stage linear relaxations. Such a sequential convexification allows us to carry information from one iteration to the next, thus avoiding the need to re-start the convexification process from scratch. Nevertheless, the goal of D^2-SC remains one in which linear relaxations of the subproblems are tightened using valid inequalities. In contrast, the cuts here provide a completely novel application of disjunctive programming. We have used disjunctive programming to approximate the value function of MIP problems. This viewpoint facilitates the design of decomposition algorithms for specially structured MIP problems, particularly SMIP problems. Because of its connections with the disjunctive decomposition (D^2) algorithm of Sen and Higle [2004], we refer to the current method as the D^2-BAC algorithm, and is summarized in Figure 1.

Proposition 3. *Let the first-stage master program approximation solved at iteration k be*

$$\mathrm{Min}\{c^\top x + \eta \mid \eta \geq 0, x \in X \cap \mathcal{B}, (\eta, x) \text{ satisfies } (21.1), \ldots, (21.k-1)\}.$$

Moreover, assume that the second-stage subproblem is a mixed-integer (binary) linear program whose partial solutions are obtained using a branch-and-bound method in

- 0. Initialize. Let $\epsilon > 0$ be given; $k = 1$, and initialize an upper bound $V_1 = \infty$. Let $F^1(x) = 0$, where $F^k(x)$ denotes the approximation of the expected recourse function at iteration k.
- 1. Solve the Master Problem and check the stopping criterion.
  Let $x^k \in \text{argmin } \{c^\top x + F^k(x) \mid x \in X \cap \mathcal{B}\}$, and let v_k denote the optimal value of the master problem. If $V_k - v_k \leq \epsilon$, stop. Otherwise, proceed to Step 2.
- 2. Update the approximation.
  For each $\omega \in \Omega$ partially "solve" one MIP subproblem using BAB and derive an inequality of the form (20), and after processing all ω, derive (21.k). If $y^k(\omega)$ satisfies the integrality restrictions for all $\omega \in \Omega$, then set $V_{k+1} = \text{Min}\{c^\top x^k + E[f(x^k, \tilde{\omega})], V_k\}$. Include the inequality (21.k) into the master program approximation, and denote the updated approximation of the expected recourse function by F^{k+1}. Increment k by 1, and repeat from Step 1.

Fig. 1. A Basic D^2-BAC Algorithm

which all LP relaxations are feasible, and nodes are fathomed only when the lower bound (on the second-stage) exceeds the best available upper bound (for the second-stage). Suppose that there exists an iteration K such that for $k \geq K$, the branch-and-bound method (for each second-stage subproblem) provides an optimal second-stage solution for all $\omega \in \Omega$, thus yielding an upper bound on the two-stage problem. Then the resulting D^2-BAC algorithm provides an optimal first-stage solution in a finite number of iterations.

Proof. We consider iteration indices $k \geq K$, so that for such k, the second-stage objective values provide MIP values for each outcome, and an upper bound on the original problem becomes available. Given the solution (η^k, x^k) of the master program at iteration k, we choose some tight optimality cut to define $\eta^k(\omega)$ as in (22). Also note that $\eta^k = E[\eta^k(\tilde{\omega})] = \sum_{\omega\in\Omega} p(\omega)\eta^k(\omega)$.

For iteration k, and outcome ω, let $\eta^k_+(\omega)$ denote the MIP optimal value for the subproblem associated with outcome ω. Following the logic stated in (17), we have

$$\eta^k_+(\omega) = \underset{q\in Q(\omega)}{\text{Min}} \{\bar{\nu}^k_q(\omega) - \bar{\gamma}^k_q(\omega)^\top x^k\}.$$

Since (19) generates a facet of the convex hull of $\Pi_k(\omega)$, and $x^k \in \mathcal{B}$ is an extreme point of X, it follows that $(\eta^k_+(\omega), x^k)$ satisfies (20) as an equality. Moreover, $\sigma^k_0(\omega) > 0$ implies that

$$\eta^k_+(\omega) = \frac{\zeta^k(\omega)}{\sigma^k_0(\omega)} - \left[\frac{\sigma^k(\omega)}{\sigma^k_0(\omega)}\right]^\top x^k. \tag{23}$$

Since there are only finitely many settings of the second-stage binary variables, there can only be finitely many disjunctions that can be stated as (17), and for each of these disjunctions, there can only be finitely many extreme points generated via (19). Hence, in the worst case, it takes finitely many iterations, say $S-1$, to generate all the extreme points of the epi-reverse polar. In the worst case then, the master program in iteration S provides a point (η^S, x^S) such that

$$\eta^S(\omega)\sigma_0(\omega) + \sigma(\omega)^\top x^S \geq \zeta(\omega), \quad \forall(\sigma_0(\omega), \sigma(\omega), \zeta(\omega)) \in \Pi_t^\dagger(\omega), \forall t \leq S-1 \quad (24)$$

for all ω. Hence (23, 24) imply that

$$-\eta^S(\omega) - \left[\frac{\sigma^S(\omega)}{\sigma_0^S(\omega)}\right]^\top x^S + \frac{\zeta^S(\omega)}{\sigma_0^S(\omega)} \leq 0 = -\eta_+^S(\omega) - \left[\frac{\sigma^S(\omega)}{\sigma_0^S(\omega)}\right]^\top x^S + \frac{\zeta^S(\omega)}{\sigma_0^S(\omega)}.$$

Consequently, at the end of iteration S, we have $\eta^S(\omega) \geq \eta_+^S(\omega)$ for all outcomes ω. Therefore,

$$v_S = c^\top x^S + \eta^S \geq c^\top x^S + E[\eta_+^S(\tilde{\omega})] \geq V^*,$$

where V^* is the optimal value of SMIP. Hence, the method terminates in finitely many steps. □

Note that the main property of the binary first-stage solutions that we have used above is that such points are extreme points of X, which helps us conclude that the closure of the convex hull of the epigraph of the value function agrees with the collection of cuts (21.k) at extreme points of X. Hence, the D^2-BAC method is applicable to a larger class of problems called extreme point mathematical programs that require feasible solutions to be extreme points of a polyhedral set (Sen and Sherali [1985]).

3.2. *Illustrations of the D^2-BAC algorithm*

In this subsection, we illustrate the workings of the D^2-BAC through some examples. The first example is the same as that in §2.3. While this instance happens to be deterministic, it is interesting to compare the master program approximations resulting from the D^2-BAC method with those of the D-BAC method. The next example will illustrate the application of D^2-BAC to an SMIP instance.

3.2.1. A deterministic instance For the sake of convenience, the instance from §2.3 is restated below.

$$\begin{aligned} \text{Min } & -x_1 - 2y_1 + 4y_2 \\ & -4x_1 - 3y_1 + y_2 \geq -6 \\ & (x_1, y_1) \text{ binary}, \ \ y_2 \geq 0. \end{aligned}$$

The D^2-BAC method requires a lower bound on the objective value of the second-stage. By noting that the value of $-2y_1 + 4y_2$ on the feasible set must be at least -2, we conclude that $\eta \geq -2$. In order to be consistent with our development that requires $\ell = 0$, we make the translation that $\nu = \eta + 2$, so that $\nu \geq 0$. Hence, the first master program (see also §2.3) is given by

$$\begin{aligned} -2 + \text{Min}\, & -x_1 + \nu \\ & \nu \geq 0 \\ & x_1 \text{ binary.} \end{aligned}$$

The above approximation yields the first iterate $x_1 = 1$, $\nu = 0$. Hence, the lower bound is -3. Using the given value of x_1, we proceed to the second-stage problem:

$$\begin{aligned} \text{Min}\ & -2y_1 + 4y_2 \\ & -3y_1 + y_2 \geq -2 \\ & y_1 \text{ binary},\ \ y_2 \geq 0. \end{aligned}$$

As before, the solution to the LP relaxation is fractional, and we solve this problem using a branch-and-bound method that results in the same two nodes as in §2.3. During the branch-and-bound process, the value of the second-stage problem is $\eta = 0$, and so, an upper bound on the original problem is -1. From the dual solutions obtained for the LP relaxations at each node, we arrive at the following disjunction.

$$\begin{aligned} & \{\eta \geq 4(-6 + 4x_1) + 10, -x_1 \geq -1, x_1 \geq 0, \eta \geq -2\}, \\ \text{or} \quad & \{\eta \geq 0, -x_1 \geq -1, x_1 \geq 0, \eta \geq -2\}. \end{aligned}$$

Once again, translating $\nu = \eta + 2$, this disjunction can be stated as

$$\begin{aligned} & \{-16x_1 + \nu \geq -12, -x_1 \geq -1, x_1 \geq 0, \nu \geq 0\}, \\ \text{or} \quad & \{\nu \geq 2, -x_1 \geq -1, x_1 \geq 0, \nu \geq 0\}. \end{aligned}$$

Solving the separation problem (19) we obtain multipliers $\tau_{01} = 1/3$, $\tau_{02} = 2/3$, $\tau_1 = 0$, $\tau_2 = 16/3$, and the cut coefficients are $\sigma_0 = 2/3$, $\sigma_1 = -16/3$, and $\zeta = -4$. The resulting disjunctive cut yields the facet

$$-8x_1 + \nu \geq -6.$$

Adding this inequality results in an updated master program

$$\begin{aligned} -2 + \text{Min}\ & -x_1 + \nu \\ & -8x_1 + \nu \geq -6 \\ & \nu \geq 0 \\ & x_1 \text{ binary.} \end{aligned}$$

This completes one iteration. Observe that the inequality $-8x_1 + \nu \geq -6$ is implied by the constraints in the final master program listed in §2.3. To see this, note that by using a multiplier 1 for the inequality $-2x_1 + \eta \geq -2$ and a multiplier 6 for the inequality $-x \geq -1$, we obtain an aggregated inequality $-8x_1 + \eta \geq -8$. Substituting $\eta = \nu - 2$ we obtain the same inequality as the facet obtained in the above master program.

We begin the next iteration by solving the master program obtained at the end of the first iteration. The optimal solution to this problem is $(x_1, \nu) = (0, 0)$. Thus, the lower bound is -2. Solving the second-stage problem with $x_1 = 0$ provides a solution yielding an upper bound of -2 with $\eta = -2$. Since the upper and lower bounds are both -2, we declare $x_1 = 0$ as an optimal solution.

The reader may also find it interesting to use the D^2-BAC method to solve the following modified instance involving general integer variables in the second-stage. (One may also use the D-BAC method for this problem.)

$$\begin{aligned} \text{Min } & -x_1 - 2y_1 + 4y_2 \\ & -4x_1 - 3y_1 + y_2 \geq -7 \\ & x_1 \text{ binary}, \ y_1 \text{ integer}, \quad y_1, y_2 \geq 0. \end{aligned}$$

In the interest of brevity, we only provide the sequence of master programs that are generated by the method. The initial master program is the same as the one given for the previous instance, with the understanding that $\nu = \eta + 6$, where $\ell = -6$ is a lower bound on the second-stage value. The second master program is

$$\begin{aligned} -6+ \text{ Min } & -x_1 + \nu \\ & -(8/3)x_1 + \nu \geq 4/3 \\ & \nu \geq 0 \\ & x_1 \text{ binary}, \end{aligned}$$

and the third one is

$$\begin{aligned} -6+ \text{ Min } & -x_1 + \nu \\ & -(8/3)x_1 + \nu \geq 4/3 \\ & \nu \geq 2 \\ & \nu \geq 0 \\ & x_1 \text{ binary}. \end{aligned}$$

It so happens that the solution x_1 provided by the second master program is $x_1 = 0$, and the branch-and-bound process for the subproblem yields an upper bound of -4 on the original problem. The optimal solution for the third master program is $(x_1, \nu) = (0, 2)$, and since $\nu = \eta + 6$, we conclude that the lower bound on the original problem is -4. Since the upper and lower bounds are equal at this point, we conclude that the first-stage solution $x_1 = 0$ is optimal.

3.2.2. A stochastic programming instance This instance is a simple extension of the instance discussed above. Consider the following SMIP:

$$\begin{aligned} \text{Min } & -x_1 + E[f(x_1, \tilde{\omega})] \\ & x_1 \text{ binary}, \end{aligned}$$

where $\tilde{\omega}$ is a discrete random variable assuming two equally likely values $\{\omega_1 = -6, \omega_2 = -2\}$, and where

$$\begin{aligned} f(x_1, \omega) = \text{ Min } & -2y_1 + 4y_2 \\ & -3y_1 + y_2 \geq \omega + 4x_1 \\ & y_1 \text{ binary}, \ y_2 \geq 0. \end{aligned}$$

Note that the first scenario results in the same subproblems as in the deterministic instance. As before, the value of $-2y_1 + 4y_2$ on the feasible set for both scenarios must be at least -2, and hence, we conclude that $\eta \geq -2$. Therefore, as before, the first master program is

$$\begin{aligned} -2+ \text{ Min } & -x_1 + \nu \\ & \nu \geq 0 \\ & x_1 \text{ binary.} \end{aligned}$$

The above approximation yields the first iterate $x_1 = 1$, $\nu = 0$, resulting in a lower bound of -3. We now proceed to solve the second-stage scenario subproblems. As in the deterministic instance, the solution to the LP relaxation for $\omega_1 = -6$ yields a fractional solution, and we solve this problem using a BAB method. Following the calculations of the previous subsection, the same two nodes result in the BAB tree, and as before, the inequality corresponding to (20) is $-8x_1 + \nu_1 \geq -6$. Next, we solve the subproblem associated with $\omega_2 = -2$. It turns out that the solution to the LP relaxation satisfies the integer restrictions, and we obtain the following Benders' cut associated with scenario ω_2: $-16x_1 + \nu_2 \geq -6$. Using the probabilities associated with ω_1 and ω_2 as weights for each inequality, the resulting cut (21) appended to the master program is $-12x_1 + \nu \geq -6$. The updated master program is then given by

$$\begin{aligned} -2+ \text{ Min } & -x_1 + \nu \\ & -12x_1 + \nu \geq -6 \\ & \nu \geq 0 \\ & x_1 \text{ binary.} \end{aligned}$$

This completes one iteration of the algorithm.

We begin the next iteration by solving the master program obtained at the end of the first iteration. The optimal solution to this problem is $(x_1, \nu) = (0, 0)$. Thus the lower bound is -2. We now go on to solve the second-stage scenario problems with $x_1 = 0$. For scenario ω_1 the LP relaxation yields an integral solution with an objective value of $f(0, \omega_1) = -2$. Using the standard Benders' inequality for this scenario, we obtain a cut $\nu(\omega_1) \geq 0$.

For scenario ω_2 the solution to the LP relaxation is fractional, and we solve this problem using a BAB method, which results in two terminal nodes. During the BAB process, the best value of the second-stage problem is 0. Thus an upper bound on the original problem is equal to -1. From the dual solutions obtained for the LP relaxation at each node, we arrive at the following disjunction:

$$\begin{aligned} & \{\eta \geq 4(-2 + 4x_1) + 10, -x_1 \geq -1, x_1 \geq 0, \eta \geq -2\} \\ \text{or} \quad & \{\eta \geq 0, -x_1 \geq -1, x_1 \geq 0, \eta \geq -2\}. \end{aligned}$$

Once again, translating $\nu = \eta + 2$, this disjunction can be stated as

$$\begin{aligned} & \{-16x_1 + \nu \geq 4, -x_1 \geq -1, x_1 \geq 0, \nu \geq 0\} \\ \text{or} \quad & \{\nu \geq 2, -x_1 \geq -1, x_1 \geq 0, \nu \geq 0\}. \end{aligned}$$

Solving (19), we obtain multipliers $\tau_{01} = 1/3, \tau_{02} = 2/3, \tau_1 = 0, \tau_2 = 0$ and coefficients $\sigma_0 = 2/3, \sigma_1 = 0$, and $\zeta = 4/3$. Thus the resulting disjunctive cut for scenario ω_2 is $\nu(\omega_2) \geq 2$. Combining the two cuts for the two scenarios as in (21) yields the aggregated facet $\nu \geq 1$. Adding this inequality results in an updated master program and completes the second iteration.

We begin the third iteration by solving the master program obtained at the end of the second iteration. The optimal solution to this problem is $(x_1, \nu) = (0, 1)$. Thus the lower bound is -1. Currently the upper bound is -1. Since the upper and lower bounds are both -1, we declare $x_1 = 0$ as an optimal solution. It is interesting to observe that while the deterministic and stochastic instances discussed above are very similar, the latter required more iterations.

4. Conclusions

In this paper, we have developed extensions of decomposition-based cutting plane algorithms to allow the use of branch-and-cut methods for stochastic mixed-integer programming (SMIP) problems. These approaches allow us to solve the original MIP using a sequence of smaller MIPs. Given the complexity of these problems, such reductions are valuable.

We have investigated two alternative procedures: the D-BAC algorithm, and the D^2-BAC method. The former uses second-stage variables as continuous decisions in the first-stage, while the latter creates master problems using only first-stage decisions. As pointed out in Remark 1, the D-BAC approach may be modified to generate projected inequalities that use only the first-stage decisions. An alternative way to see the connections between the developments in sections 2 and 3 is by noting that for binary deterministic problems of the form considered in Section 2, (9) yields the following conditionally valid inequality at node q where $y_j = 1, \forall j \in J^+_{2q}$, and $y_j = 0, \forall j \in J^-_{2q}$:

$$\eta \geq \theta^\top_{kq}(r_k - T_k x) + \sum_{j \in J^+_{2q}} \psi^+_{kjq}.$$

Denoting Q as the index set for the fathomed end-nodes of the corresponding BAB tree, and observing that any y satisfying the binary restrictions must correspond to exactly one of these end-nodes, we can state a valid disjunction as follows:

$$\cup_{q \in Q}\{\eta \geq \theta^\top_{kq}(r_k - T_k x) + \sum_{j \in J^+_{2q}} \psi^+_{kjq}, x \in X, \eta \geq \ell\}.$$

Then, the disjunctive cut principle can be applied as in Section 3, and cuts can be derived in the space of (η, x) variables.

The D-BAC method of Section 2 is convenient in the deterministic setting, or for SMIP problems having only a few scenarios in the second-stage. However, if the number of outcomes in the second-stage is very large (as in large-scale SMIPs), the D^2-BAC method of Section 3 (or equivalently, the above modification of D-BAC) allows a more direct facility for creating a master program that uses only the first-stage variables.

Acknowledgements. This research has been supported by grants DMI-9978780, CISE-9975050, and DMI-0094462 from the National Science Foundation. We also thank the referees for their comments which helped improve upon an earlier version of the paper. We would like to thank Lewis Ntaimo for several interesting observations, and his assistance with some of the examples.

References

Ahmed, S., Tawarmalani, M., Sahinidis, N.V.: A finite branch and bound algorithm for two-stage stochastic integer programs. Math. Prog. **100**, 355–377 (2004)

Balas, E.: Disjunctive programming. Annals of Discrete Mathematics **5**, 3–51 (1979)

Balas, E., Ceria, S., Cornuéjols, G.: A lift-and-project cutting plane algorithm for mixed 0-1 programs. Math. Prog. **58**, 295–324 (1993)

Blair, C.: A closed-form representation of mixed-integer program value functions. Math. Prog. **71**, 127–136 (1995)

Blair, C., Jeroslow, R.: The value function of an integer program. Math. Prog. **23**, 237–273 (1982)

Birge, J.R., Louveaux, F.: Introduction to Stochastic Programming. Springer, Berlin, Germany, 1997

Caroe, C.C.: Decomposition in Stochastic Integer Programming. Ph.D. thesis, Institute of Mathematical Sciences, Dept. of Operations Research, University of Copenhagen, Denmark, 1998

Caroe, C.C., Schultz, R.: Dual decomposition in stochastic integer programming. Operations Research Letters **24**, 37–45 (1999)

Jeroslow, R.: A cutting plane game for facial disjunctive programs. SIAM Journal on Control and Optimization **18**, 264–281 (1980)

Klein Haneveld, W.K., van der Vlerk, M.H.: Stochastic integer programming: general models and algorithms. Annals of Operations Research **85**, 39–57 (1999)

Laporte, G., Louveaux, F.V.: The integer L-shaped methods for stochastic integer programs with complete recourse. Operations Research Letters **13**, 133–142 (1993)

Lovász, L., Schrijver, A.: Cones of matrices and set functions and 0-1 optimization. SIAM J. on Optimization **1**, 166–190 (1991)

Lulli, G., Sen, S.: A branch and price algorithm for multi-stage stochastic integer programs with applications to stochastic lot sizing problems. Management Science **50**, 786–796 (2004)

Ntaimo, L., Sen, S.: The million variable "march" for stochastic combinatorial optimization. To appear in Journal of Global Optimization 2003

Ruszczyński, A.: Decomposition methods in stochastic programming. Mathematical Programming B (Liebling and de Werra (eds.)), **79**, 333–353 (1997)

Schultz, R.: On the structure and stability in stochastic programs with random technology matrix and complete integer recourse. Mathematical Programming **70**, 73–89 (1995)

Schultz, R.: Stochastic programming with integer variables. Mathematical Programming Series B **97**, 285–309 (2003)

Sen, S.: Decomposition algorithms for stochastic mixed-integer programming models. To appear in Handbook of Discrete Optimization. (K. Aardal, G. Nemhauser, R. Weismental, eds.), Elsevier Publishers, 2003

Sen, S., Higle, J.L.: The C^3 theorem and a D^2 algorithm for large scale stochastic optimization: set convexification. To appear in Mathematical Programming, 2004

Sen, S., Higle, J.L., Ntaimo, L.: A summary and illustration of disjunctive decomposition with set convexification. Stochastic Integer Programming and Network Interdiction Models. (D.L. Woodruff ed.), Kluwer Academic Press, Dordrecht, The Netherlands, 2002, pp. 105–125

Sen, S., Sherali, H.D.: A branch and bound algorithm for extreme point mathematical programming. Discrete Applied Mathematics **11**, 265–280 (1985)

Sherali, H.D., Adams, W.P.: A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM Journal on Discrete Mathematics **3**, 411–430 (1990)

Sherali, H.D., Adams, W.P.: A hierarchy of relaxations and convex hull characterizations for mixed integer zero-one programming problems. Discrete Applied Mathematics **52**, 83–106 (1994)

Sherali, H.D., Adams, W.P.: A Reformulation-Lizearization Technique for Solving Discrete and Continuous Nonconvex problems. Kluwer Academic Publishers, Boston, MA, 1999

Sherali, H.D., Fraticelli, B.M.P: A modification of Benders' decomposition algorithm for discrete subproblems: an approach for stochastic programs with integer recourse. Journal of Global Optimization **22**, 319–342 (2002)