PROCESS DESIGN AND CONTROL

A Disjunctive Cutting-Plane-Based Branch-and-Cut Algorithm for 0−1 Mixed-Integer Convex Nonlinear Programs

Yushan Zhu*

Department of Chemical Engineering, Tsinghua University, Beijing 10084, People's Republic of China

Takahito Kuno

Institute of Information Sciences and Electronics, University of Tsukuba, Ibaraki 305-8573, Japan

In this paper, a disjunctive cutting-plane-based branch-and-cut algorithm is developed to solve the 0−1 mixed-integer convex nonlinear programming (MINLP) problems. In a branch-and-bound framework, the 0−1 MINLP problem is approximated with a 0−1 mixed-integer linear program at each node, and then the lift-and-project technology is used to generate valid cuts to accelerate the branching process. The cut is produced by solving a linear program that is transformed from a projection problem, in terms of the disjunction on a free binary variable, and its dual solutions are applied to lift the cut to become valid throughout the enumeration tree. A strengthening process is derived to improve the coefficients of the cut by imposing integrality on the left free binary variables. Finally, the computational results on four test problems indicate that the added cutting planes can reduce the branching process greatly and show that the proposed algorithm is very promising for large-scale 0−1 MINLP problems, because a linear program is always computationally less expensive than a nonlinear program.

1. Introduction

Mixed-integer nonlinear programming (MINLP) has had a crucial role over the last two decades for chemical process design via a superstructure approach that always involves discrete and continuous variables,[1] and now MINLP has almost become a new "paradigm" for chemical process integration. The formerly much-investigated MINLP problems involve continuous variables and binary variables, and the main characteristics defining their mathematical structures are linearity of the binary variables and convexity of the nonlinear functions that involve only continuous variables.[2] This separable and convex mathematical programming structure has many variants, and its application arises in many areas, especially the process synthesis problems in chemical engineering.[3,4] In this paper, a more general formulation of the 0−1 MINLP problem is considered:

$$
(\mathrm{P0})\quad \left|\begin{array}{ll} \min\limits_{x,y} & f(x,y) \\ \text{s.t.} & g(x,y) \le 0 \\ & Ax + Gy \le b \\ & x \in \mathscr{R}^n,\ y \in \{0,1\}^q \end{array}\right.
$$

where the functions $f:\mathscr{R}^{n+q} \rightarrow \mathscr{R}$ and $g:\mathscr{R}^{n+q} \rightarrow \mathscr{R}^m$ are assumed to be convex and, at one time, continuously differentiable. Note that all linear constraints have been incorporated into the polyhedral set described by $Ax + Gy \le b$. The solution to this general 0−1 MINLP problem was largely motivated by the recent progress from process design with control,[5] process simulation of discontinuous dynamic systems,[6] model predictive control of hybrid systems,[7] and process integration in terms of global optimization,[4,8−10] where a reliable and fast solver for large-scale 0−1 MINLP problems is necessary. The 0−1 MINLP problems formulated by Duran and Grossmann[12] can be solved using two well-known techniques, i.e., the general Benders decomposition method[11] and the outer approximation method.[2] The latter method was further extended by Fletcher and Leyffer[12] to solve problem P0. Generally, those methods solve the MINLP problem by computing a nonlinear program (NLP) primal problem at some particular integer combination and a mixed-integer linear program (MILP) master problem alternatively, to get the upper bound and lower bound of the original MINLP problem, respectively. At each round, the general Benders decomposition method just generates one cut to update the constraint set of the MILP master problem, then it yields a very loose lower bound. For the outer approximation method, because all constraints will be incorporated into the former constraint set of the MILP master problem, the number of the cutting planes then grows linearly with the number of nonlinear constraints, which is a great burden for the solution to the MILP master problem, although it produces a tight lower bound.

Although the branch-and-bound method is most popular for solving integer programming or MILP problems,[13] it is now understood that the branch-and-bound method can benefit considerably from strong relaxations. In particular, the successful employment of the branch-and-cut method for 0−1 integer programming[14,15] and 0−1 MILP[16,17] has spurred great interest in its application for 0−1 MINLP, because of the significant

* To whom correspondence should be addressed. E-mail: yszhu@tsinghua.edu.cn.

10.1021/ie0402719 CCC: $33.50
Published on Web 11/19/2005

progress of the interior point algorithm for convex programming problems. To decrease the computational efforts to solving NLPs further, Tawarmalani and Sahinidis[18,19] have pioneered the development of polyhedral relaxations for convex and nonconvex MINLPs under the branch-and-cut framework. The problem formulation of P0 can be rearranged to have a linear objective function by introducing an additional variable:

$$(\mathrm{P1})\quad \left| \begin{array}{ll} \min\limits_{x,y} & x_{n+1} \\ \text{s.t.} & f(x,y) - x_{n+1} \leq 0 \\ & g(x,y) \leq 0 \\ & Ax + Gy \leq b \\ & x \in \mathcal{R}^n, y \in \{0,1\}^q \end{array} \right.$$

A linear objective function is necessary for a cutting-plane type of algorithm, because, for general convex functions, it is possible that the optimum of the relaxation problem takes in the strict interior of the feasible set; therefore, we cannot add a valid inequality in this situation. However, for the aforementioned formulation, it is easy to see that the valid inequalities can be added, because the optimum of the relaxation problem always reaches at an extreme point of its feasible set. Stubbs and Mehrotra[20] generalized the lift-and-project method for 0−1 integer linear programming proposed by Sherali and Adams[14] and Lovasz and Schrijver[15] and extended their method into a branch-and-cut algorithm for the 0−1 mixed-integer convex programming. The lifting idea for 0−1 linear programming can be traced back to the classic work by Balas,[16] but not formally published until 1998, for disjunctive programming. It is interesting to note that the generalized idea developed by Stubbs and Mehrotra[20] can be taken as an natural extension of the Theorem 2.1 written in ref 21 by Balas. The lift-and-project cut presented by Stubbs and Mehrotra[20] is obtained by solving a convex projection problem, so it is computationally expensive. In this paper, it is shown that a valid lift-and-project cut can be constructed by solving a linear programming problem. Note that the cut generation linear program (LP) proposed by Balas and co-workers[16,17,22] can be considered to be the dual problem of our projection problem with a different norm objective function. However, their dual problem is always unbounded without normalization. Moreover, the cut generation LP in this paper can be easily solved using a dual simplex method, because a dual feasible basis is immediately available from our formulation. The disjunctive cut constructed at the current node can be further lifted to be valid throughout the enumeration tree by virtue of the dual solutions to the aforementioned LP problem. A strengthening procedure is also presented when the difference between the two dual multipliers for the disjunctive representation of the binary variables exists. The computational results suggest that the disjunctive cut developed in this paper is effective to accelerate the branching process and makes the branch-and-cut algorithm become very promising for large-scale 0−1 MINLP problems.

2. Branch-and-Cut Procedure for 0−1 Mixed-Integer Nonlinear Programming

The general 0−1 MINLP problem considered in this paper can be formulated as

$$(\mathrm{P})\quad \left| \begin{array}{ll} \min\limits_{x,y} & dx \\ \text{s.t.} & Ax + Gy \leq b \\ & g_i(x,y) \leq 0, i = 1, ..., l \\ & x \in \mathcal{R}^n, y \in \{0,1\}^q \end{array} \right.$$

where the constant vectors and matrixes are defined as $d \in \mathcal{R}^n$, $A \in \mathcal{R}^{m\times n}$, $G \in \mathcal{R}^{m\times q}$, $b \in \mathcal{R}^m$. Let the feasible region of the standard continuous relaxation of problem (P) be defined as

$$C = \left\{ (x,y) \in \mathcal{R}^{n+q} \middle| \begin{array}{l} Ax + Gy \leq b \\ g_i(x,y) \leq 0, i = 1, ..., l \\ 0 \leq y \leq 1 \end{array} \right\}$$

Hence, the feasible set of the 0−1 MINLP (P) can be formulated as

$$C^0 = \{(x,y) \in C{:}y_j \in \{0, 1\}, j = 1, ..., q\}$$

The following assumptions are made:

(A1) The continuous variables are bounded, i.e., $|x_i| \leq L$, $i = 1, ..., n$, which can be taken as a part of the constraints in problem (P), and L is a large positive number.

(A2) For any $(x,y) \in C$, $g_i(x,y) \geq -L$, $i = 1, ..., l$.

(A3) A constraint qualification holds at the solution of every resulting NLP problem that is obtained from problem (P) by fixing some or all of the binary variables.

When the aforementioned assumptions are valid, we know that problem (P) is well-defined in the sense that if it is feasible and does not have an unbounded optimal value, then it has an optimal solution. At a generic step of the branch-and-cut algorithm,[17,20] let $(\bar{x},\bar{y})$ be a solution to the current NLP relaxation of (P). If any of the components of the binary variables $\bar{y}$ are not in {0,1}, then we can add a valid inequality into the current feasible set, such that this inequality is violated by $(\bar{x},\bar{y})$. At the same time, it should be noted that some of the binary variables are fixed at either upper or lower bound in an enumeration tree. We denote the family of inequalities to describe the current feasible set and the newly incorporated inequalities. We denote by $F_0,F_1 \subseteq \{1, ..., q\}$ the sets of binary variables that have been fixed at 0 and 1, respectively. Let

$$K(c,F_0,F_1) = \left\{ (x,y) \in c \middle| \begin{array}{ll} y_j = 0 & (\text{for } j \in F_0) \\ y_j = 1 & (\text{for } j \in F_1) \end{array} \right\}$$

and let NLP(c,F_0,F_1) denote the nonlinear program

$$\min_{x,y} dx$$

$$\text{s.t. } (x,y) \in K(c,F_0,F_1)$$

This NLP is assumed to be either bounded from below with a finite minimum or infeasible at the current node, by virtue of the above-stated assumptions. The active nodes of the enumeration tree are represented by a list S with ordered pairs (F_0,F_1). Let UBD represent the current upper bound, i.e., the value of the best-known solution to the MINLP problem (P).

The branch-and-cut procedure for convex 0−1 MINLP can be written as follows, for an input of d, n, q, A, G, b, and f_i (for $i = 1, ..., l$):

(1) Initialization. Set $S = \{(F_0 = \phi, F_1 = \phi)\}$, and let c consist of the nonlinear programming relaxation of (P) and UBD $= \infty$.

(2) Node Selection. If $S = \phi$, stop. Otherwise, choose an ordered pair $(F_0,F_1) \in S$ and remove it from S.

(3) Lower Bounding Step. Solve the nonlinear program NLP(c,F_0,F_1). If the problem is infeasible, go to Step 2. Otherwise, let $(\bar{x},\bar{y})$ denote its optimal solution. If $d\bar{x} \geq$ UBD, go to Step 2.

If $\bar{y}_j \in \{0, 1\}$, $j = 1,...,q$, let $(x^*, y^*) = (\bar{x},\bar{y})$, UBD $= d\bar{x}$, and go to Step 2.

(4) Branching versus cutting decision. Should cutting planes be generated? If yes, go to Step 5; otherwise, go to Step 6.

(5) Cut generation. Generate a cutting plane for which $\alpha x + \beta y \le \gamma$ is valid for the (P) but is violated by $(\bar{x},\bar{y})$. Add the cuts into c and go to Step 3.

(6) Branching step. Pick an index $j \in \{1, ..., q\}$ such that $0 < \bar{y}_j < 1$. Generate the subproblems that correspond to $(F_0 \cup \{j\}, F_1)$ and $(F_0, F_1 \cup \{j\})$, add them into the node set S. Go to Step 2.

When the algorithm terminates, if UBD $< \infty$, (x^*,y^*) is an optimal solution to (P); otherwise (P) is infeasible. All steps in this procedure are defined completely, except for Step 5. The cut generation step introduced in the underlying section is divided into two stages. First, a valid inequality is generated at the current node on the basis of a linear approximation of the MINLP problem. Second, the resulting cut is lifted to become valid throughout the enumeration tree.

3. Separating Inequality

For 0−1 mixed-integer problems, the lift-and-project cut is obtained by a procedure that lifts the constraint set of the original problem into a higher dimensional space by introducing new variables, adds to it some equations implied by the 0−1 conditions, and then projects it back onto the original space by eliminating the new variables.[16,17,20] The newly incorporated inequality is still satisfied by every feasible solution to the MINLP, but it cuts off some parts of the original space that violates the 0−1 conditions. The outcome of the process of lifting−strengthening−projecting is a constraint set tighter than the original one. For the MILP problem, the lift-and-project cut can be generated directly from the original linearly constrained set; however, the straightforward extension is not convenient for MINLP problems, because it concerns the nonlinear constraints. Stubbs and Mehrotra[20] proposed an excellent way to generate a lift-and-project cut by solving a convex minimum distance problem after reformulating the projection problem using the perspective functions[23] or reference to the right scalar multiplication in function operations.[24] In this paper, it is shown that the valid lift-and-project cut can be obtained by solving a linear programming (LP) rather than convex programming for the 0−1 MINLP problems.

3.1. Linear Approximation of the NLP Relaxation. The continuous relaxation of the 0−1 MINLP at some node in an enumeration tree can be described by

$$(\text{NLP})\quad \left|\begin{array}{ll} \min\limits_{x,y} & dx \\ \text{s.t.} & \tilde{A}x + \tilde{G}y \le \tilde{b} \\ & g_i(x,y) \le 0,\ i = 1, ..., l \\ & y_j = 0,\ j \in F_0 \\ & y_j = 1,\ j \in F_1 \\ & (x,y) \in \mathcal{R}^{n+q} \end{array}\right.$$

where the reformulated linear constraint set consists of the original linear constraint set and the upper and lower bound constraints for binary variables, so we have $\tilde{A} \in \mathcal{R}^{(m+2q)\times n}$, $\tilde{G} \in \mathcal{R}^{(m+2q)\times q}$, $\tilde{b} \in \mathcal{R}^{m+2q}$. Assume that the above NLP continuous problem is feasible and has a finite minimum at $(\bar{x},\bar{y})$, because, otherwise, the node is done. A linear approximation problem at $(\bar{x},\bar{y})$ for the above NLP problem can be obtained by

$$(\text{LP})\quad \left|\begin{array}{l} \min\limits_{x,y} dx \\ s.t.\ \tilde{A}x + \tilde{G}y \le \tilde{b}, \\ \quad g_i(\bar{x},\bar{y}) + \nabla g_i(\bar{x},\bar{y})\begin{pmatrix} x-\bar{x} \\ y-\bar{y}\end{pmatrix} \le 0, \quad i = 1,...,l, \\ \quad y_j = 0, \quad j \in F_0, \\ \quad y_j = 1, \quad j \in F_1, \\ \quad (x,y) \in \Re^{n+q}, \end{array}\right.$$

where the original convex and differentiable functions are replaced by their first-order Taylor approximation at $(\bar{x},\bar{y})$. Accordingly, a MILP problem corresponding to the MINLP problem at the current node can be described byThe following

$$(\text{MILP})\quad \left|\begin{array}{l} \min\limits_{x,y} dx \\ s.t.\ \tilde{A}x + \tilde{G}y \le \tilde{b}, \\ \quad g_i(\bar{x},\bar{y}) + \nabla g_i(\bar{x},\bar{y})\begin{pmatrix} x-\bar{x} \\ y-\bar{y}\end{pmatrix} \le 0, \quad i = 1,...,l, \\ \quad y_j = 0, \quad j \in F_0, \\ \quad y_j = 1, \quad j \in F_1, \\ \quad x \in \Re^n,\ y \in \{0,1\}^q, \end{array}\right.$$

theorem defines the relationship between the NLP and its LP approximation at $(\bar{x},\bar{y})$ and helps to realize that $(\bar{x},\bar{y})$ is an extreme point of the relaxed feasible set of the aforementioned MILP, in the sense that it can be cut away by introducing a valid disjunctive cut to the aforementioned MINLP feasible set.

Theorem 3.1. *Assume that the above NLP achieves its optimal solution at $(\bar{x},\bar{y})$. Then, $(\bar{x},\bar{y})$ is also an optimal solution to the aforementioned LP.*

Proof. *See Theorem 4.2.15 on page 153 of ref 25.*

The geometrical explanation of the aforementioned linear approximation is presented in Figure 1, and it is obvious that the mixed-integer set is expanded after the linear approximation. If we introduce the subdifferential concept (for reference, see ref 24 or 23) of the convex function into our problem, then the assumption that the nonlinear function is continuously differential is not needed. For a nonlinear inequality constraint $g_i(x,y) \le 0$ defining C, let

$$\partial g_i(\bar{x},\bar{y}) \equiv \left\{ \xi_i \middle| g_i(\bar{x},\bar{y}) + \xi_i^{\mathrm{T}}\begin{pmatrix} x-\bar{x} \\ y-\bar{y}\end{pmatrix} \le g_i(x,y) \text{ (for all } (x,y) \in C) \right\}$$

The linear approximation set of C then can be constructed by choosing any element from this subdifferential set. The geometrical explanation is meant using the tangent cone at the point $(\bar{x}, \bar{y})$ to approximate the convex set, and the statement claimed in Theorem 3.1 still holds, according to the general Kuhn−Tucker conditions for convex programming.[24] By virtue of Theorem 3.1, it is interesting to note that the Gomory mixed-integer cut[26] can be used to cut away the point $(\bar{x}, \bar{y})$, because the optimal simplex tableau can be easily recovered when an optimal solution is available. The lifting procedure for the Gomory mixed-integer cut for MILP is provided by Balas et al.[27] However, the efficiency of the Gomory mixed-integer cut is not comparable to the disjunctive cut, which is elaborately generated by solving cut generation linear programming that has been constructed in the consequent sections.

3.2. Lift-and-Project Cut Generation. For problem (P), it is very attractive to construct the lift-and-project cut in terms of the approximated LP instead of the NLP, but the cut still

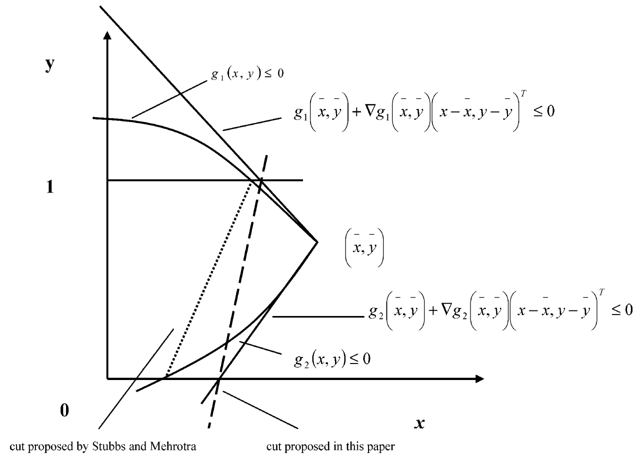

Figure 1. Linear approximation of the mixed-integer nonlinear programming (MINLP) problem, where the mixed-integer convex set is described by a continuous variable and a binary variable.

can cut away the fraction point $(\bar{x}, \bar{y})$. Such cut can be derived by imposing the 0−1 integral condition on a binary variable y_j while $0 < \bar{y}_j < 1$. In Figure 1, the short dashed line represents the cut generated directly using the convex hull of the mixed-integer convex set presented by Stubbs and Mehrotra,[20] and the long dashed line represents the cut to be generated in this paper, based on the linear approximation. The direct cut generation needs to solve a convex program, which is computationally expensive, although it yields better quality than that constructed in this paper, if measured by some norm distance between the cut hyper-plane and the current fraction solution $(\bar{x},\bar{y})$. For cut generation, the node sets F_0 and F_1 can be expanded to include additional binary variables whose optimal solutions are taken at 0 or 1 for the NLP problem at the current node. We then redefine these two sets as $\bar{F}_0 = \{i{:}\ \bar{y}_i = 0\}$ and $\bar{F}_1 = \{i{:}\ \bar{y}_i = 1\}$. It is not difficult to verify that the aforementioned NLP and LP problems have the same optimal solutions if we change the original node sets to be the expanded sets. Let the feasible region of the above LP be defined as

$$K = K(c,\bar{F}_0,\bar{F}_1) = \left\{ (x,y) \in \mathscr{R}^{n+q} \;\middle|\; \begin{array}{ll} \bar{A}x + \bar{G}y \le \bar{b} & \\ y_i = 0 & (i \in \bar{F}_0) \\ y_i = 1 & (i \in \bar{F}_1) \end{array} \right\}$$

where $\bar{A} \in \mathscr{R}^{(m+2q+l)\times n}$, $\bar{G} \in \mathscr{R}^{(m+2q+l)\times q}$, $\bar{b} \in \mathscr{R}^{m+2q+l}$, i.e., the newly reformulated linear constraint set consists of the linear approximation set as well as the original one, as

$$\bar{A}x + \bar{G}y \le \bar{b} \equiv \begin{cases} Ax + Gy \le b, \\ \nabla g^x(\bar{x},\bar{y})x + \nabla g^y(\bar{x},\bar{y})y \le \nabla g(\bar{x},\bar{y})\begin{pmatrix}\bar{x}\\ \bar{y}\end{pmatrix} - g(\bar{x},\bar{y}), \\ -y_i \le 0 \quad i = 1,...,q, \\ y_i \le 1 \quad\ \ i = 1,...,q, \end{cases}$$

Note that the node sets have already been changed to the expanded sets in the aforementioned formulation. If we impose the integrality condition on a binary variable y_j for which $0 < \bar{y}_j < 1$, the lift-and-project cut can be obtained by choosing a valid inequality for

$$P_j(K) = conv(K \cap \{(x,y) \in \mathscr{R}^{n+q}{:}y_j \in \{0,1\}\})$$

The convex hull of this union set can be further described by its disjunctive form as

$$P_j(K) = \mathrm{conv}(\{K \cap \{(x,y) \in \mathscr{R}^{n+q}{:}y_j \le 0\}\} \cup \{K \cap \{(x,y) \in \mathscr{R}^{n+q}{:}-y_j \le -1\}\})$$

Note that the disjunction $(y_j \le 0) \vee (-y_j \le -1)$ instead of $(y_j = 0) \vee (y_j = 1)$ is used to describe $P_j(K)$, although both are equivalent to each other. A key idea of disjunctive programming is that it is equivalent to an LP in a higher-dimensional space, such as that described in Theorem 2.1 in ref 21. In this paper, it is shown that the projection problem to generate the cut can be achieved using this reformulation. First, we introduce the following notations to simplify the description of the constraint set of the aforementioned LP, using a linear transformation. Let $F = \{1, ..., q\}\backslash(\bar{F}_0 \cup \bar{F}_1)$ denote the set of free variables at node $(\bar{F}_0,\bar{F}_1)$, and the vector corresponding to those free variables can be defined as $y^F = y\backslash\{y_i{:}\ i \in (\bar{F}_0 \cup \bar{F}_1)\}$. The columns of matrix $\bar{G}$ corresponding to the fixed binary variables can be removed from the constraint set by defining $\bar{G}^F = \bar{G}\backslash\{\bar{G}_i{:}\ i \in (\bar{F}_0 \cup \bar{F}_1)\}$, and the right-hand side can be calculated accordingly, as $\bar{b}^F = \bar{b} - \sum_{i\in F_1}\bar{G}_i$. Finally, the rows in matrices $\bar{A}$ and $\bar{G}^F$, and vector $\bar{b}^F$, that correspond to the upper and lower bounds of the fixed binary variables are removed. After the above operations have been performed, we can assume, without loss of generality, that $F_1 = \phi$, because if $F_1 \ne \phi$, all the variables y_k in F_1 can be complemented by $1 - y_k$, which is equivalent to replacing the columns G_k and the right-hand side with $-G_k$ and $b - G_k$, respectively. Before we proceed to the next part, we assume that we have already done those complementing operations on the linear constraint set of the aforementioned LP. More clearly, we use the following equations to show the aforementioned reduction process for the linear constraint set of the LP approximation at the current node. After complementing, the LP constraint set can be described as

$$\bar{A}x + \sum_{i\in F\cup F_0}\bar{G}_i y_i + \sum_{i\in F_1}(-\bar{G}_i y_i) \le \bar{b} - \sum_{i\in F_1}\bar{G}_i$$

which we denote by $\bar{A}x + \bar{G}^Q y \le \bar{b}^Q$. The reduced LP constraint set after removing the fixed binary variables becomes

$$\bar{A}^F x + \bar{G}^F y^F \le \bar{b}^F \equiv \begin{cases} Ax + \sum_{i\in F} G_j y_j \le b - \sum_{i\in F_1} G_j, \\ \nabla g^x(\bar{x},\bar{y})x + \sum_{i\in F}\nabla g_i^y(\bar{x},\bar{y})y_j \le \nabla g(\bar{x},\bar{y})\begin{pmatrix}\bar{x}\\ \bar{y}\end{pmatrix} - g(\bar{x},\bar{y}) - \sum_{i\in F_1}\nabla g_i^y(\bar{x},\bar{y}), \\ -y_i \le 0 \quad i \in F, \\ y_i \le 1 \quad\ \ i \in F, \end{cases}$$

where $\bar{A}^F \in \mathscr{R}^{(m+2|F|+l)\times n}$, $\bar{G}^F \in \mathscr{R}^{(m+2|F|+l)\times |F|}$, $\bar{b}^F \in \mathscr{R}^{m+2|F|+l}$. The feasible region of the aforementioned LP then can be reformulated as

$$K = \{(x,y^F) \in \mathscr{R}^{n+|F|}|\bar{A}^F x + \bar{G}^F y^F \le \bar{b}^F\}$$

By virtue of Theorem 2.1 in ref 21, we have the following theorem.

Theorem 3.2. *The convex hull of $P_j(K)$, i.e., $conv(P_j(K))$, can be described by*

$$\mathrm{conv}(P_j(K)) = \left\{ (x,y^F) \in \mathcal{R}^{n+|F|} \;\middle|\; \begin{array}{l} x = u_0 + u_1 \\ y^F = v_0 + v_1 \\ \bar{A}^F u_0 + \bar{G}^F v_0 - \bar{b}^F \lambda_0 \le 0 \\ v_{0,j} \le 0 \\ \bar{A}^F u_1 + \bar{G}^F v_1 - \bar{b}^F \lambda_1 \le 0 \\ -v_{1,j} \le -\lambda_1 \\ \lambda_0 + \lambda_1 = 1 \\ \lambda_0,\lambda_1 \ge 0,\ u_0,u_1 \in \mathcal{R}^n,\ v_0,v_1 \in \mathcal{R}^{|F|} \end{array} \right\}$$

Proof. *The bounded property of $P_j(K)$ is implied by the assumptions that all continuous variables are bounded. Obviously, the same is true for conv($P_j(K)$). The above theorem then is a direct result of Theorem 2.1 in ref 21 being applied to the disjunctive set $P_j(K)$.*

Let $(\bar{x},\bar{y})$ be the optimal solution when solving NLP($c,\bar{F}_0,\bar{F}_1$). First, we assume that $(\bar{x},\bar{y})$ is not feasible to problem (P) and let j be the binary variable index such that $0 < \bar{y}_j < 1$. To find a strong valid inequality for $P_j(K)$ that cuts away $(\bar{x}, \bar{y})$ as much as possible, Balas et al.[16] proposed to solve a cut generation LP that maximizes the Euclidean distance between $(\bar{x}, \bar{y})$ and some facet of $P_j(K)$. In fact, such generation procedure is exactly the dual linear program of the projection problem, i.e., the problem of minimizing the distance between the point $(\bar{x},\bar{y})$ and the convex hull of $P_j(K)$ measured by some norm.[20] In this paper, the l_1-norm is used to generate a valid cut for $P_j(K)$:

$$(\mathrm{CP(F)})\quad \begin{array}{|ll} \min & f(x,y^F) \\ \text{s.t.} & x = u_0 + u_1 \\ & y^F = v_0 + v_1 \\ & \bar{A}^F u_0 + \bar{G}^F v_0 - \bar{b}^F \lambda_0 \le 0 \\ & v_{0,j} \le 0 \\ & \bar{A}^F u_1 + \bar{G}^F v_1 - \bar{b}^F \lambda_1 \le 0 \\ & -v_{1,j} \le -\lambda_1 \\ & \lambda_0 + \lambda_1 = 1 \\ & \lambda_0,\lambda_1 \ge 0 \\ & u_0,u_1 \in \mathcal{R}^n, v_0,v_1 \in \mathcal{R}^{|F|} \end{array}$$

where $f(x,y^F) = ||(x,y^F) - (\bar{x},\bar{y}^F)||_1 = \sum_{i\in N}|x_i - \bar{x}_i| + \sum_{i\in F}|y_i^F - \bar{y}_i^F|$. The following theorem ensures that a valid inequality can be obtained after solving the aforementioned projection problem. Note that the only convexity of the aforementioned program is implied by the piecewise objective function.

Theorem 3.3. *Let $(\tilde{x},\tilde{y}^F)$ be an optimal solution to the above CP(F) problem. Then, there exists a valid inequality $\alpha x + \beta^F y^F \le \gamma$, which cuts away $(\bar{x},\bar{y}^F)$ from conv($P_j(K)$). The coefficients of this valid inequality are given by*

$$\alpha = -\partial f_x(\tilde{x},\tilde{y}^F)$$

$$\beta^F = -\partial f_{y^F}(\tilde{x},\tilde{y}^F)$$

$$\gamma = -\partial f_x(\tilde{x},\tilde{y}^F)\tilde{x} - \partial f_{y^F}(\tilde{x},\tilde{y}^F)\tilde{y}^F$$

Proof. *Since $(\tilde{x},\tilde{y}^F)$ is an optimal solution to the above projection problem, then by virtue of Theorem 3.4.3 on page 101 in ref 25, we have*

$$-\partial f(\tilde{x},\tilde{y}^F)\begin{pmatrix} x - \tilde{x} \\ y^F - \tilde{y}^F \end{pmatrix} \le 0 \qquad (\text{for all } (x,y^F) \in \mathrm{conv}(P_j(K)))$$

where $\partial f(\tilde{x},\tilde{y}^F)$ is the subdifferential of $f(x,y^F)$ at $(\tilde{x},\tilde{y}^F)$, and then we get the cut stated in the above theorem by doing some rearrangements. It is obvious to observe that function $f(x,y^F)$ is convex, so according to the definition of the subdifferential of the convex function, we have

$$f(\tilde{x},\tilde{y}^F) + \partial f(\tilde{x},\tilde{y}^F)\begin{pmatrix} x - \tilde{x} \\ y^F - \tilde{y}^F \end{pmatrix} \le f(x,y^F)$$

By letting $(x,y^F) = (\bar{x},\bar{y}^F)$ and noting $(\tilde{x},\tilde{y}^F) \ne (\bar{x},\bar{y}^F)$, we have

$$-\partial f(\tilde{x},\tilde{y}^F)\begin{pmatrix} \bar{x} - \tilde{x} \\ \bar{y}^F - \tilde{y}^F \end{pmatrix} \ge f(\tilde{x},\tilde{y}^F) - f(\bar{x},\bar{y}^F) \ge ||(\tilde{x},\tilde{y}^F) - (\bar{x},\bar{y}^F)||_1 > 0$$

Therefore, the valid inequality denoted by $\alpha x + \beta^F y^F \le \gamma$ in the theorem cuts away $(\tilde{x},\tilde{y}^F)$ from $P_j(K)$.

The objective function defined in CP(F), i.e., $f(x,y^F)$, is convex but not differentiable at point $(\bar{x}, \bar{y}^F)$. It is easy to observe that the subdifferential at this point belongs to a cone set. However, there exists only one element in the subdifferential set at any other point, i.e., the gradient of the function $f(x,y^F)$. Because the relation $(\tilde{x},\tilde{y}^F) \ne (\bar{x},\bar{y}^F)$ always holds in the aforementioned projection problem, we then can change the subdifferential stated in the above theorem into the gradient:

$$\alpha = -\nabla f_x(\tilde{x},\tilde{y}^F)$$

$$\beta^F = -\nabla f_{y^F}(\tilde{x},\tilde{y}^F)$$

$$\gamma = -\nabla f_x(\tilde{x},\tilde{y}^F)\tilde{x} - \nabla f_{y^F}(\tilde{x},\tilde{y}^F)\tilde{y}^F$$

The l_1-norm convex function $f(x,y^F)$, or piecewise linear function, can be easily transformed to a linear function by introducing some additional variables, and the resulting projection problem is equivalent to an LP by increasing additional linear constraints on those newly introduced variables, denoted as LP(F):

$$\begin{array}{|ll} \min \sum_{i\in N} z_i + \sum_{i\in F} w_i & \\ \text{s.t.} & x = u_0 + u_1 \\ & y^F = v_0 + v_1 \\ & \bar{A}^F u_0 + \bar{G}^F v_0 - \bar{b}^F \lambda_0 \le 0 \\ & v_{0,j} \le 0 \\ & \bar{A}^F u_1 + \bar{G}^F v_1 - \bar{b}^F \lambda_1 \le 0 \\ & -v_{1,j} \le -\lambda_1 \\ & \lambda_0 + \lambda_1 = 1 \\ & -z + x \le \bar{x} \\ & -z - x \le -\bar{x} \\ & -w + y^F \le \bar{y}^F \\ & -w - y^F \le \bar{y}^F \\ & \lambda_0,\lambda_1 \ge 0 \\ & x,u_0,u_1,z \in \mathcal{R}^n, y^F,v_0,v_1,w \in \mathcal{R}^{|F|} \end{array}$$

This linear program has $4n + 4|F| + 2$ variables with $2m + 3n + 7|F| + 2l + 3$ equality or inequality constraints. By noting the coefficients of those correlations, it is clear that the problem scale of LP(F) can be largely reduced by decreasing the free binary variable set. After solving this LP, we get its solutions, denoted by $(\tilde{x},\tilde{y}^F,\tilde{u}_0,\tilde{u}_0,\tilde{v}_1,\tilde{v}_1,\tilde{z},\tilde{w},\tilde{\lambda}_0,\tilde{\lambda}_1)$, as well as the dual multipliers. Note that those dual multipliers are equivalent to the Lagrange multipliers if we solve the convex program directly. Using the symbols π_x^F, π_y^F, δ_λ^F for the multipliers for the equality constraints, δ_0^F and δ_1^F for the disjunctive inequality constraints, μ_0^F and μ_1^F for the inequality original constraints, and ϵ_+^F, ϵ_-^F, φ_+^F, and φ_-^F for the additional constraints in LP(F), we have the following dual linear program of LP(F), denoted as DLP(F):

$$
\left|
\begin{array}{ll}
\max & -\delta_\lambda^F - \epsilon_+^F\bar{x} + \epsilon_-^F\bar{x} - \varphi_+^F\bar{y}^F + \varphi_-^F\bar{y}^F \\
\text{s.t.} & \pi_x^F + \epsilon_+^F - \epsilon_-^F \geq 0 \\
& \pi_y^F + \varphi_+^F - \varphi_-^F \geq 0 \\
& -\pi_x^F + \mu_0^F\bar{A}^F \geq 0 \\
& -\pi_x^F + \mu_1^F\bar{A}^F \geq 0 \\
& -\pi_y^F + \mu_0^F\bar{G}^F + \delta_0^F e_j \geq 0 \\
& -\pi_y^F + \mu_1^F\bar{G}^F - \delta_1^F e_j \geq 0 \\
& -\mu_0^F\bar{b}^F + \delta_\lambda^F \geq 0 \\
& -\mu_1^F\bar{b}^F + \delta_\lambda^F + \delta_1^F \geq 0 \\
& -\epsilon_+^F - \epsilon_-^F \geq -1 \\
& -\varphi_+^F - \varphi_-^F \geq -1 \\
& \mu_0^F,\mu_1^F,\epsilon_+^F,\epsilon_-^F, \varphi_+^F,\varphi_-^F \geq 0 \\
& \pi_x^F,\epsilon_+^F,\epsilon_-^F \in \mathscr{R}^n, \pi_y^F,\varphi_+^F,\varphi_-^F \in \mathscr{R}^{|F|} \\
& \mu_0^F,\mu_1^F \in \mathscr{R}^{m+2|F|+l}, \delta_0^F,\delta_1^F,\delta_\lambda^F \in \mathscr{R}
\end{array}
\right.
$$

The corresponding dual multipliers to LP(F) can also be obtained by solving this dual linear program directly. Following the aforementioned notations, we have $(\tilde{\pi}_x^F,\tilde{\pi}_y^F,\tilde{\mu}_0^F,\tilde{\delta}_0^F,\tilde{\mu}_1^F,\tilde{\delta}_1^F,\tilde{\delta}_\lambda^F,\tilde{\epsilon}_+^F,\tilde{\epsilon}_-^F,\tilde{\varphi}_+^F,\tilde{\varphi}_-^F)$ as the solution to the aforementioned DLP(F). We have assumed that the MINLP is bounded; therefore, the corresponding LP(F) generated at the current node also is bounded. The following relation can be easily obtained from strong duality:

$$1 \cdot \tilde{z} + 1 \cdot \tilde{w} = -\tilde{\delta}_\lambda^F - \tilde{\epsilon}_+^F\bar{x} + \tilde{\epsilon}_-^F\bar{x} - \tilde{\varphi}_+^F\bar{y}^F + \tilde{\varphi}_-^F\bar{y}^F$$

By virtue of the Kuhn−Tucker condition of the CP(F), we have

$$-\nabla f_x(\tilde{x},\tilde{y}^F) = \pi_x^F$$

$$-\nabla f_{y^F}(\tilde{x},\tilde{y}^F) = \pi_y^F$$

The cut generated in Theorem 3.2 (i.e., $\alpha x + \beta^F y^F \leq \gamma$) can be reformulated by the dual multipliers and the primal solutions to LP(F), as

$$\pi_x^F x + \pi_y^F y^F \leq \pi_x^F \tilde{x} + \pi_y^F \tilde{y}^F$$

By virtue of the dual linear program (i.e., DLP(F)), we have

$$-\pi_x^F + \mu_0^F\bar{A}^F \geq 0$$

$$-\pi_x^F + \mu_1^F\bar{A}^F \geq 0$$

$$-\pi_y^F + \mu_0^F\bar{A}^F + \delta_0^F e_j \geq 0$$

$$-\pi_y^F + \mu_1^F\bar{A}^F - \delta_1^F e_j \geq 0$$

$$\delta_\lambda^F - \mu_0^F\bar{b}^F \geq 0$$

$$\delta_\lambda^F - \mu_1^F\bar{b}^F + \delta_1^F \geq 0$$

Note that, for an optimal solution to LP(F), which is also dual feasible, LP(F) is always bounded, according to the former assumptions. We then have

$$\alpha_i = \min\{\alpha_i^1, \alpha_i^2\} \qquad (\textit{for } i \in N)$$

$$\beta_i = \min\{\beta_i^1, \beta_i^2\} \qquad (\text{for } i \in F)$$

where

$$\alpha^1 = \mu_0^F\bar{A}^F$$

$$\alpha^2 = \mu_1^F\bar{A}^F$$

and

$$\beta^1 = \mu_0^F\bar{A}^F + \delta_0^F e_j$$

$$\beta^2 = \mu_1^F\bar{A}^F - \delta_1^F e_j$$

By virtue of the above derivation, we know that the cut denoted by $\alpha x + \beta^F y^F \leq \gamma$ at the current node (F_0, F_1) is a valid inequality for the MILP problem at that node, because it was produced according to the linear relaxation of the MILP problem at that node, as well as the 0−1 integral condition imposed on a free binary variable. Because every nonlinear function of the MINLP is assumed to be convex, we have

$$g_i(\bar{x},\bar{y}) + \nabla g_i(\bar{x},\bar{y})\begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix} \leq g_i(x,y) \leq 0 \qquad (\text{for } i = 1, ..., l)$$

This relation implies that the feasible set of the MINLP at the node (F_0, F_1) is contained in the feasible set of the MILP at that node. Therefore, the inequality $\alpha x + \beta^F y^F \leq \gamma$ is valid and proper for the MINLP problem at the current node denoted by (F_0, F_1), and its descendents, where the variables in (F_0, F_1) remain fixed.

3.3. Cut Lifting. Generally, a cutting plane derived at some point defined by the subset (F_0, F_1) in an enumeration tree is only valid at that node and its descendants, as stated in the previous section. Such a cut may not be valid throughout the enumeration tree and, therefore, might be violated by other nodes. So it is extremely important to lift the cut generated at that node to make it valid throughout the enumeration tree, because it not only can reduce the need for extensive bookkeeping, but also can possibly improve the bounds at other nodes. A cut generated at the current node can be made valid for the entire MINLP problem by computing appropriate coefficients for the fixed binary variables belonging to the subset (F_0, F_1), but the sequential lifting procedure[13] generally is a daunting task, which may require the solution to an integer program for every coefficient. An important advantage of the cut generated by the lift-and-project technology is that the multipliers (i.e., μ_0^F, δ_0^F, μ_1^F, δ_1^F) obtained along with the solution $(\tilde{x},\tilde{y}^F)$ by solving LP(F) can be used to calculate the closed form expressions of the coefficients β_i for the binary variables in the index set $F_0 \cup F_1$. First, we lift the inequality obtained at the current node into the complemented original space of the MILP problem; that is,

$$\left\{(x,y) \in \mathscr{R}^{n+q}\text{:}\ \left|\begin{matrix}\bar{A}x + \bar{G}^Q & y \le \bar{b}^Q \\ y \in \{0,1\}^q & \end{matrix}\right.\right\}$$

Let $j \in \{1, ..., q\}$ be an index such that $0 < \bar{y}_j < 1$ and consider the inequality $\alpha^q x + \beta^q y \le \gamma^q$ generated over the complemented original space of the MILP problem; this is done to solve the linear program LP(Q). For clarity, the corresponding CP(Q) to this linear program can be expressed as

$$(\text{CP(Q)})\quad \left|\begin{array}{ll} \min & f(x,y) \\ \text{s.t.} & x = u_0 + u_1 \\ & y = v_0 + v_1 \\ & \bar{A}u_0 + \bar{G}^Q v_0 - \bar{b}^Q\lambda_0 \le 0 \\ & v_{0,j} \le 0 \\ & \bar{A}u_1 + \bar{G}^Q v_1 - \bar{b}^Q\lambda_1 \le 0 \\ & -v_{1,j} \le -\lambda_1 \\ & \lambda_0 + \lambda_1 = 1 \\ & \lambda_0,\lambda_1 \ge 0 \\ & x,u_0,u_1 \in \mathscr{R}^n, y,v_0,v_1 \in \mathscr{R}^q \end{array}\right.$$

Compare this convex program with CP(F), the new cut vector $(\alpha^q,\beta^q,\gamma^q)$ can be calculated by the cut vector (α,β^F,γ) and its multipliers, i.e., $\tilde{\mu}_0^F,\tilde{\mu}_1^F$. By virtue of the Kuhn−Tucker condition of the aforementioned convex programming, we have

$$-\nabla f_x(\hat{x},\hat{y}) = \hat{\pi}_x^Q$$

$$-\nabla f_y(\hat{x},\hat{y}) = \hat{\pi}_y^Q$$

where $(\hat{x},\hat{y})$ are the solutions to the aforementioned convex program, and $(\hat{\pi}_x^Q,\hat{\pi}_y^Q)$ are the corresponding Lagrange multipliers. Those values can be obtained by the corresponding LP-(Q) to the above CP(Q), but first, we have the following notations compared with those that appeared in LP(F):

$$\bar{A} = \begin{pmatrix}\bar{A}^F \\ 0\end{pmatrix}$$

$$\bar{G}^Q = \begin{pmatrix}\bar{G}^F & \bar{G}^{QF} \\ 0 & \bar{G}^{QQ}\end{pmatrix}$$

$$\bar{b}^Q = \begin{pmatrix}\bar{b}^F \\ \bar{b}^{QF}\end{pmatrix}$$

$$y = \begin{pmatrix}y^F \\ y^{QF}\end{pmatrix}$$

where the set $QF = F_0 \cup F_1$ represents the node set (F_0,F_1). The linear program LP(Q) corresponding to CP(Q) can be presented as

$$\left|\begin{array}{ll} \min \sum_{i\in N} z_i + \sum_{i \in Q} w_i & \\ \text{s.t.} & x = u_0 + u_1 \\ & y = v_0 + v_1 \\ & \bar{A}u_0 + \bar{G}^Q v_0 - \bar{b}^Q\lambda_0 \le 0 \\ & v_{0,j} \le 0 \\ & \bar{A}u_1 + \bar{G}^Q v_1 - \bar{b}^Q\lambda_1 \le 0 \\ & -v_{1,j} \le -\lambda_1 \\ & \lambda_0 + \lambda_1 = 1 \\ & -z + x \le -\bar{x} \\ & -z - x \le -\bar{x} \\ & -w + y \le \bar{y} \\ & -w - y \le -\bar{y} \\ & \lambda_0,\lambda_1 \ge 0 \\ & x,u_0,u_1,z \in \mathscr{R}^n, y,v_0,v_1,w \in \mathscr{R}^q \end{array}\right.$$

Note that more variables and constraints are added into this linear program, compared to LP(F). However, using the solutions to LP(F) and its multipliers, we can obtain the optimal solution to the above LP(Q). The following theorem shows how those solutions can be obtained and how the inequality $\alpha x + \beta^F y^F \le \gamma$ can be lifted into a valid cut $\alpha^q x + \beta^q y \le \gamma^q$ throughout the enumeration tree.

Theorem 3.4. *Let $(\hat{x},\hat{y},\hat{u}_0,\hat{v}_0,\hat{u}_1,\hat{v}_1,\hat{z},\hat{w},\hat{\lambda}_0,\hat{\lambda}_1)$ be the solution to the linear program LP(Q), which can be constructed by the solution to the LP(F), as $\hat{x} = \tilde{x}$, $\hat{y} = (\tilde{y}^F,0)$, $\hat{u}_0 = \tilde{u}_0$, $\hat{u}_1 = \tilde{u}_1$, $\hat{v}_0 = (\tilde{v}_0,0)$, $\hat{v}_1 = (\tilde{v}_1,0)$, $\hat{\lambda}_0 = \tilde{\lambda}_0$, $\hat{\lambda}_1 = \tilde{\lambda}_1$, $\hat{z} = \tilde{z}$, and $\hat{w} = (\tilde{w},0)$. The corresponding dual multipliers of LP(Q), denoted by $(\hat{\pi}_x^Q, \hat{\pi}_y^Q,\hat{\mu}_0^Q,\hat{\delta}_0^Q,\hat{\mu}_1^Q,\hat{\delta}_1^Q,\hat{\delta}_\lambda^Q,\hat{\epsilon}_+^Q,\hat{\epsilon}_-^Q,\hat{\varphi}_+^Q,\hat{\varphi}_-^Q)$, which is also the solution to the dual linear program DLP(Q), then can be constructed by those to DLP(F), as $\hat{\pi}_x^Q = \tilde{\pi}_x^F$, $\hat{\pi}_{y,i}^Q = \tilde{\pi}_{y,i}^F$ for $i \in F$, $\hat{\pi}_{y,i}^Q = \min\{\tilde{\mu}_0^F\,\bar{G}_i^Q,\tilde{\mu}_1^F\bar{G}_i^Q\}$ for $i \in F_0 \gg F_1$, $\hat{\delta}_0^Q = \tilde{\delta}_0^F$, $\hat{\delta}_1^Q = \tilde{\delta}_1^F$, $\hat{\delta}_\lambda^Q = \tilde{\delta}_\lambda^F$, $\hat{\mu}_0^Q = (\tilde{\mu}_0^F,0)$, $\hat{\mu}_1^Q = (\tilde{\mu}_1^F,0)$, $\hat{\epsilon}_+^Q = \tilde{\epsilon}_+^Q$, $\hat{\epsilon}_-^Q = \tilde{\epsilon}_-^Q$, $\hat{\varphi}_+^Q = (\tilde{\varphi}_+^F,0)$, and $\hat{\varphi}_-^Q = (\tilde{\varphi}_-^F,0)$. The inequality $\alpha^q x + \beta^q y \le \gamma^q$ described by the relation $\hat{\pi}_x^Q x + \hat{\pi}_y^Q y \le \hat{\pi}_x^Q\hat{x} + \hat{\pi}_y^Q\hat{y}$ is valid for the entire enumeration tree and cuts away $(\bar{x},\bar{y})$.*

Note that the cut is "optimally" lifted by Theorem 3.4, in the sense that the resulting cut, $\hat{\pi}_x^Q x + \hat{\pi}_y^Q y \le \hat{\pi}_x^Q\hat{x} + \hat{\pi}_y^Q\hat{y}$, is identical to that which would be obtained by solving LP(Q) directly.

3.4. Cut Strengthening. The mixed-integer inequality $\alpha^q x + \beta^q y \le \gamma^q$ obtained in the preceding section can be further strengthened by imposing integrality on the left free binary variables, and this process can be easily implemented using the Gomory mixed-integer rounding.[13] However, Balas et al.[16] introduced a cut strengthening process[28,29] that originated from the disjunctive program for mixed-integer cut and showed that the Gomory mixed-integer rounding is just a special case of this general method. Here, we apply the original idea of the disjunction by Balas (given as Theorem 7.1 in ref 28) into the primal problem and strengthen our lift-and-project inequality.

Theorem 3.5. *The inequality $\alpha x + \beta y \le \gamma$ that is strengthened from the valid inequality $\alpha^q x + \beta^q y \le \gamma^q$ is also valid for MINLP, and the cut vector is given by*

$$\alpha_i = \alpha_i^q \qquad (\text{for } i = 1, ..., n)$$

$$\beta_i = \begin{cases} \max \{\beta_i^1 + \delta_0^F \lceil \bar{m}_i \rceil, \beta_i^2 - \delta_1^F \lfloor \bar{m}_i \rfloor\} & (\text{for } i \in F\backslash\{j\}) \\ \beta_i^q & (\text{for } i \in F_0 \cup F_1 \cup \{j\}) \end{cases}$$

$$\gamma = \gamma^q$$

where $\beta_i^1, \beta_i^2, \delta_0^F, \delta_1^F$ are obtained via the solution to LP(F), and

$$\bar{m}_i = \frac{\beta_i^2 - \beta_i^1}{\delta_0^F + \delta_1^F} \qquad (\text{for } i \in F\backslash\{j\})$$

By virtue of the argument of the above theorem, the strengthening gap is determined by $|\beta_i^2 - \beta_i^1|$, and this absolute value becomes zero when the ith binary is taken at 0 or 1.[22] As commented by Balas et al.[16] for cut strengthening, a linear transformation (i.e., complementing) of the linear constraint set $\bar{A}x + \bar{G}y \leq \bar{b}$ leaves K, $P_j(K)$, and its facets unchanged; however, it can change the effect of the strengthening procedure. Note that the aforementioned cut strengthening can be done after the solution to LP(F) is defined; the disjunction then is changed only a posteriori. Remember that we always assume that the index set F_1 is empty. If this is not true for the MINLP problem at the current node, complementing is done on this set to change it to be empty. A simple reverse linear transformation can get the cut that is valid for the original MINLP problem before complementing:

$$\alpha x + \sum_{i \in F \cup F_0} \beta_i y_i + \sum_{i \in F_1} (-\beta_i) y_i \leq \gamma + \sum_{i \in F_1} (-\beta_i)$$

Finally, this cut is added into the feasible sets of the NLP and the cut generation LP that are described in the procedure of the branch-and-cut algorithm.

3.5. Cut Quality and Facet-Defining Inequality. For cutting-plane algorithms, it is always difficult to evaluate the cut quality and its role during the iteration process, because some cuts may not work well immediately after their generation but will become very tight after some iterations. Then, although the closed gap gained by the introduction of the cut is used, it is still not certain to measure the cut quality. In this paper, the distance between the point and the inequality that the former violates is used to measure the cut quality; however, the conditions under which this method can produce a facet-defining inequality (i.e., the strongest cut) still are not known. From Theorem 3.2, or Theorem 2.1 in ref 21, if the projection of the point $(\bar{x}, \bar{y}^F)$ onto conv($P_j(K)$) lies on the relative interior of a $(n + |F|)$-dimensional face, then the cut generated by Theorem 3.3 is a facet-defining inequality. However, in Theorem 3.3, the higher-dimensional polyhedron, instead of conv($P_j(K)$) itself, is used to get the projection, and a relative interior of a facet of this higher-dimensional polyhedron does not necessarily correspond to a relative interior of a facet of its projection onto $(\bar{x}, \bar{y}^F)$ space, i.e., conv($P_j(K)$); then, the cut generated in Theorem 3.3 does not necessarily correspond to a facet-defining inequality of conv($P_j(K)$), although the projection of $(\bar{x}, \bar{y}^F)$ onto the higher-dimensional polyhedron is a relative interior of a facet. In regard to the projection used to obtain conv($P_j(K)$) from a higher-dimensional polyhedron, the calculation involves the use of the polyhedral cone:

$$W = \left\{ (\pi_x, \pi_y, \mu_0, \delta_0, \mu_1, \delta_1, \delta_\lambda) \middle| \begin{array}{l} -\pi_x + \mu_0 A^F = 0 \\ -\pi_x + \mu_1 A^F = 0 \\ -\pi_y + \mu_0 G^F + \delta_0 e_j = 0 \\ -\pi_y + \mu_1 G^F - \delta_1 e_j = 0 \\ -\mu_0 b^F + \delta_\lambda = 0 \\ -\mu_0 b^F + \delta_1 + \delta_\lambda = 0 \\ \mu_0 \geq 0, \mu_1 \geq 0, \delta_0 \geq 0, \delta_1 \geq 0 \\ \pi_x \in \mathcal{R}^n, \pi_y \in \mathcal{R}^{|F|}, \mu_0, \mu_1 \in \mathcal{R}^{n+2|F|+l} \\ \delta_0, \delta_1, \delta_\lambda \in \mathcal{R} \end{array} \right\}$$

The projection of the higher-dimensional polyhedron onto $(\bar{x}, \bar{y}^F)$ space (i.e., conv($P_j(K)$)) can be described by

$$\text{conv}(P_j(K)) = \{(x, y^F): \ \pi_x x + \pi_y y^F \leq \delta_\lambda \text{ for all } (\pi_x, \pi_y, \delta_\lambda) \in \text{extr}(\text{Proj}(w))\}$$

where extr(Proj(W)) is the set of extreme rays of the projection of W onto $(\pi_x, \pi_y, \delta_\lambda)$ space, which is also a polyhedral cone. According to a well-known result on linear inequalities (see Theorem 22.3 in ref 24), we have the following intuitive result, given in Theorem 3.6.

Theorem 3.6. *The inequality $\pi_x x + \pi_y y^F \leq \delta_\lambda$ is valid for conv($P_j(K)$) if and only if there exist vectors $(\pi_x, \pi_y, \mu_0, \delta_0, \mu_1, \delta_1, \delta_\lambda)$ that satisfy the conditions defining W.*

Just as Balas[21] discussed, on the basis of reverse cone, we have the following condition under which the facet-defining inequality is able to be obtained.

Theorem 3.7. *If the cut vector $(\pi_x, \pi_y, \delta_\lambda)$ corresponds to an extreme ray of the polyhedral cone of the projection of W, the cut $\pi_x x + \pi_y y^F \leq \delta_\lambda$ is a facet-defining inequality to conv-($P_j(K)$), and vice versa.*

However, an extreme ray of the higher-dimensional cone does not necessarily correspond to an extreme ray of its projection onto the original space. This is still an open problem for the lift-and-project cutting plane algorithm;[22] hence, the resulting cutting plane produced in the branch-and-cut algorithm may be not the strongest one (i.e., the facet-defining inequality), but they are still valid to conv($P_j(K)$).

4. Computational Results for Process Design Problems

In this section, the effectiveness of the disjunctive cut or lift-and-project cut in a branch-and-bound framework is demonstrated on the basis of the algorithm developed at the preliminary stage. The algorithm has been implemented using standard C language on a Pentium III personal computer with a 800 MHz computer processing unit (CPU) and 128 MB of random access memory (RAM). To compare the computational efficiency on different computers, the number of iterations or the enumerated nodes and the LPs solved rather than the CPU time are used as the main factors between a pure branch-and-bound algorithm and the proposed branch-and-cut algorithm, which has imbedded the disjunctive cut. In regard to the specific implementation aspects, the binary tree is searched according to the depth-first principle and the cuts are generated at each enumerated node if there is a binary variable fraction. Note that the latter choice has answered the question presented in the algorithm that was described in Section 2 (i.e., the decision regarding branching versus cutting). However, for large problems, it is recommended for MILP by Balas et al.[17] that a skip factor obtained by the calculation results at the root node be introduced to promote

Table 1. Characteristics of the Test Problems

	Number of Variables		Number of Constraints	
test problem	continuous[a]	binary	linear[a]	nonlinear[a]
1	4	3	10	3
2	7	5	22	4
3	10	8	39	5
4	6	25	15	26

[a] These numbers have been recounted after the original mixed-integer nonlinear programming (MINLP) formulation described by Duran and Grossmann[2] was converted to the standard one presented in Section 1.

Table 2. Computational Results for the Test Problems

	Branch-and-Bound		Branch-and-Cut		
test problem	number of enumerated nodes[a]	CPU time (s)	number of enumerated nodes[a]	number of cutting planes	CPU time (s)
1	6 (5)	0.016	6 (5)	3	0.047
2	9 (8)	0.171	9 (8)	9	0.953
3	23 (22)	8.781	17 (15)	22	3.046
4	56 (47)	10.641	9 (8)	71	4.703

[a] The number given in the parentheses is the node where the optimum was reached.

the algorithmic efficiency. However, in this paper, we just simply chose to let the value of that parameter be unity, to observe the roles played by the disjunctive cuts versus the branching process. The continuous relaxation problems are solved using an NLP solver LSGRG2C,[30,31] and the cut generation linear programming problems are solved by a dual simplex algorithm, because a dual feasible basis is immediately available by noting that all the cost coefficients of the objective function in the cut generation linear program are non-negative. Four process design problems presented by Duran and Grossmann[2] are applied. These 0−1 MINLP problems have nonlinear objective functions; as a result, the additional continuous variables are introduced to transform them to the standard formulation described in Section 1. The problem characteristics are given in Table 1, and the computational results are presented in Table 2.

The computational results in Table 2 show that the proposed branch-and-cut algorithm for four MINLP problems has converged on the optima, and there is no observation that the optimal solution was cut away. For the latter two slightly large problems, the effect of the cutting planes has been observed. For the third problem, after adding 22 cutting planes, 6 NLP runs are saved in the enumeration tree. For the last problem with 25 binary variables, after producing 71 cutting planes, 47 NLP runs are reduced. In regard to the comparison with the method developed by Stubbs and Mehrotra,[20] for the last problem, their best result is that 67 NLPs are solved, including 44 for branching nodes and 25 for cut generation. Our result for this test problem is that 9 NLPs for branching nodes and 71 LPs for cut generation are solved. This result states that the proposed branch-and-cut algorithm in this paper is a good prospect for large-scale 0−1 MINLP problems, because an LP is always computationally cheaper than an NLP for large problems.

5. Conclusion

A branch-and-cut algorithm is developed in this paper to solve the 0−1 mixed-integer nonlinear programming (MINLP) problem, where the disjunctive cuts are generated and incorporated into an enumeration process. The main novelty of the approach is the use of a linear approximation for the 0−1 MINLP at each node. In other words, the feasible region of the 0−1 mixed-integer convex set is approximated by a polyhedral set that was pioneered by Tawarmalani and Sahinidis.[18] The main distinguishing feature of the approach of the current paper is the use of a polyhedral relaxation for the generation of the lift-and-project cuts. The lift-and-project cut generation is performed via linear programming, as opposed to the convex nonlinear approach used by Stubbs and Mehrotra.[20] This new approach has the advantage of making the cut generation computationally less expensive and overcoming the nondifferential problems, although the cuts are weaker than those of Stubbs and Mehrotra,[20] because the actual feasible region is being replaced by a larger set. The computational results for four test examples show that the cut is effective to accelerate the branching process and the proposed branch-and-cut algorithm based on this valid cut is comparable to that of Stubbs and Mehrotra[20] and is very promising for large-scale 0−1 MINLP problems.

Acknowledgment

Y.Z. gratefully acknowledges financial support from the Japan Society for the Promotion of Science (JSPS) and JSPS Fellow Project No. 01040 and sincerely appreciates the kind help from Prof. E. Balas (Carnegie−Mellon University) for his patient explanation of the cut lifting and strengthening. Y.Z. is a JSPS Fellow. He is also gratefully supported by a Grant-in-Aid for Scientific Research from JSPS. The enormous improvements of this paper made according to the excellent comments of three anonymous reviewers are sincerely acknowledged. This project is also supported by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, Ministry of Education, China, and by National Natural Science Foundation of China with Grant No. 20506013.

Literature Cited

(1) Grossman, I. E.; Westerberg, A. W. Research challenges in process systems engineering. *AIChE J.* **2000**, *46*, 1700−1703.

(2) Duran, M. A.; Grossmann, I. E. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.* **1986**, *36*, 307−339.

(3) Floudas, C. A. *Nonlinear and Mixed-Integer Optimization, Fundamentals and Applications*; Oxford University Press: New York, 1995.

(4) Zhu, Y.; Kuno, T. Global Optimization of Nonconvex MINLP by a Hybrid Branch-and-Bound and Revised General Benders Decomposition Approach. *Ind. Eng. Chem. Res.* **2003**, *42*, 528−539.

(5) Mohideen, M. J.; Perkins, J. D.; Pistikopoulos, E. N. Optimal design of dynamic systems under uncertainty. *AIChE J.* **1996**, *42*, 2251−2272.

(6) Barton, P. I.; Allgor, R. J.; Feehery, W. F.; Galan, S. Dynamic Optimization in a Discontinuous World. *Ind. Eng. Chem. Res.* **1998**, *37*, 966−981.

(7) Bemporad, A.; Morari, M. Control of systems integrating logic, dynamics, and constraints. *Automatica* **1999**, *35*, 407−427.

(8) Zhu, Y.; Xu, Z. Calculation of Liquid−Liquid Equilibrium Based on the Global Stability Analysis for Ternary Mixtures by Using a Novel Branch and Bound Algorithm: Application to UNIQUAC Equation. *Ind. Eng. Chem. Res.* **1999**, *38*, 3549−3556.

(9) Zhu, Y.; Inoue, K. Calculation of chemical and phase equilibrium based on stability analysis by QBB algorithm: Application to NRTL equation. *Chem. Eng. Sci.* **2001**, *56*, 6915−6931.

(10) Zhu, Y.; Kuno, T. A global optimization method, QBB, for twice-differentiable nonconvex optimization problem. *J. Global Optim.* **2005**, *33*, 435−464.

(11) Geoffrion, A. M. Generalized Benders decomposition. *J. Opt. Theory Appl.* **1972**, *10*, 237−260.

(12) Fletcher, R.; Leyffer, S. Solving mixed-integer nonlinear programs by outer approximation. *Math. Program.* **1994**, *66*, 327−349.

(13) Nemhauser, G. L.; Wolsey, L. A. *Integer and Combinatorial Optimization*; Wiley: New York, 1988.

(14) Sherali, H.; Adams, W. A hierarchy of relaxations between the continuous and convex hull representation for zero-one programming problems. *SIAM J. Discrete Math.* **1990**, *3*, 411−430.

(15) Lovasz, L.; Schrijver, A. Cones of matrixes and set functions and 0−1 optimization. *SIAM J. Optim.* **1991**, *1*, 166−190.

(16) Balas, E.; Ceria, S.; Cornuéjols, G. A lift-and-project cutting plane algorithm for mixed-integer 0−1 programs. *Math. Program*. **1993**, *58*, 295−324.

(17) Balas, E.; Ceria, S.; Cornuejols, G. Mixed 0−1 programming by lift-and-project in a branch-and-cut framework. *Manage. Sci*. **1996**, *42*, 1229−1246.

(18) Tawarmalani, M.; Sahinidis, N. V. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Math. Program., Ser. A* **2004**, *99*, 563−591.

(19) Tawarmalani, M.; Sahinidis, N. V. A polyhedral branch-and-cut approach to global optimization. *Math. Program., Ser. B* **2005**, *103*, 225−249.

(20) Stubbs, R. A.; Mehrotra, S. A branch-and-cut method for 0−1 mixed convex programming. *Math. Program.* **1999**, *86*, 515−532.

(21) Balas, E. Disjunctive programming: properties of the convex hull of feasible points, Technical Report No. MSRR 348, Carnegie Mellon University, Pittsburgh, PA, 1974. (Also in *Discrete Appl. Math*. **1998**, *89*, 3−44).

(22) Balas, E.; Perregaard, M. Lift-and-project for mixed 0−1 programming: recent progress. *Discrete Appl. Math*. **2002**, *123*, 129−154.

(23) Hiriart-Urruty, J.-B.; Lemaréchal, C. *Convex Analysis and Minimization Algorithm*−*1. Fundamentals*; Springer−Verlag: Berlin, 1993; Vol. 305.

(24) Rockafellar, R. T. *Convex Analysis*; Princeton University Press: Princeton, NJ, 1970.

(25) Bazaraa, M. S.; Sherali, H. D.; Shetty, C. M. *Nonlinear Programming: Theory and Algorithms*, Second Edition; Wiley: New York, 1993.

(26) Gomory, R. E. An algorithm for the mixed-integer problem, Report No. RM-2597, The Rand Corporation, Santa Monica, CA, 1960.

(27) Balas, E.; Ceria, S.; Cornuejols, G.; Natraj, N. Gomory cuts revisited. *Oper. Res. Lett.* **1996**, *19*, 1−9.

(28) Balas, E. Disjunctive Programming. *Ann. Discrete Math*. **1979**, *5*, 3−51.

(29) Balas, E.; Jeroslow, R. G. Strengthening cuts for mixed integer programming. *Eur. J. Oper. Res*. **1980**, *4*, 224−234.

(30) Smith, S.; Lasdon, L. Solving Large Sparse Nonlinear Programs Using GRG. *ORSA J. Comput.* **1992**, *4*, 2−15.

(31) Lasdon, L. *LSGRG Version 3.0 Release Notes*; MSIS Department, College of Business Administration, University of Texas at Austin, Austin, TX, 2000.

Received for review November 1, 2004
Revised manuscript received October 5, 2005
Accepted October 14, 2005

IE0402719