Math. Program., Ser. A 89: 437–448 (2001)

Digital Object Identifier (DOI) 10.1007/s101070000179

Jonathan H. Owen* · Sanjay Mehrotra**

A disjunctive cutting plane procedure for general mixed-integer linear programs***

Received: January 1999 / Accepted: March 2000
Published online December 15, 2000 –

Abstract. In this paper we develop a cutting plane algorithm for solving mixed-integer linear programs with general-integer variables. A novel feature of the algorithm is that it generates inequalities at all γ-optimal vertices of the LP-relaxation at each iteration. The cutting planes generated in the procedure are found by considering a natural generalization of the 0-1 disjunction used by Balas, Ceria, and Cornuéjols in the context of solving binary mixed-integer linear programs [3,4].

Key words. mixed integer programming

1. Introduction

In recent years we have seen increased research activity directed towards using disjunctive programming techniques to solve linear programs with binary integer variables [3, 7,8]. This research provides ways to describe the integer hull of linear programs with binary integer variables by using a finite set of linear inequalities. More importantly, it has resulted in computationally better ways of generating cuts in a branch-and-cut method for solving general binary integer linear programs [4] as well as structured binary integer linear programs [5].

For mixed binary linear problems, the two related developments have been the lift-and-project approach of Balas, Ceria and Cornuéjols [3] and the reformulation-linearization technique of Sherali and Adams [8,9]. Each of these developments have their roots in disjunctive programming (see [2]). Both Balas, Ceria and Cornuéjols [3] and Sherali and Adams [8,9] showed that the convex hull of feasible mixed binary solutions could be generated in a finite number of steps. Balas, Ceria and Cornuéjols also demonstrated that the cuts generated from the lift-and-project scheme could be used to efficiently solve these problems in a branch-and-cut framework [3,4]. The

J.H. Owen: GM R&D Center, Mail Code 480-106-359, 30500 Mound Road, Warren, MI 48090-9055, USA, e-mail: jonathan.owen@gm.com

S. Mehrotra: Department of Industrial Engineering and Management Sciences, Robert R. McCormick School of Engineering, Northwestern University, Evanston, Illinois 60208, USA, e-mail: mehrotra@iems.nwu.edu

Mathematics Subject Classification (1991): 90C10, 90C11

* The work was performed when the first author was at Northwestern University.

** The author thanks the Supreme Lord for all revelations leading to this paper. All its fruits are surrendered to Him.

*** This work was supported in part by the ONR grant N00014-95-1-0743, and NSF grant DMI-9908038.

convexification procedure of Balas, Ceria and Cornuéjols [3] has been extended by Stubbs and Mehrotra [10] for convex mixed binary problems. The developments of this paper follow more closely the description of the lift-and-project approach of Balas, Ceria and Cornuéjols [3].

In this paper we consider using a variable disjunction to solve general mixed-integer linear programs (MILP). We describe the variable disjunction and define our notation in Sect. 2. In Sect. 3 we consider a convexification procedure for MILP that is based on sequential application of the variable disjunction. We show that the procedure (1) finitely generates the convex hull with respect to any single general-integer variable and (2) yields in the limit the complete convex hull of mixed-integer solutions. In Sect. 4 we describe a cutting plane algorithm based on the variable disjunction that finds an ε-optimal solution of MILP in a finite number of iterations. A novel feature of the algorithm is that it generates cuts at all γ-optimal vertices of the relaxation of MILP at each iteration. Practical implementations of the branch-and-cut algorithm for structured binary problems typically decide to branch when improvements in the objective value tail off. Our analysis suggests that generating cuts at neighboring vertices in addition to the current optimal vertex may be a viable alternative to branching.

2. Variable disjunctions

Consider the general mixed-integer linear programming problem

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && x \in F, \end{aligned} \tag{MILP}$$

where

$$\begin{aligned} & && F = \{x \in K \mid x_i \in \mathbb{Z} \quad \text{for } i \in \{1, \ldots, p\}\} \\ &\text{and} && K = \{x \in \mathbb{R}^n \mid Ax \geq b\}. \end{aligned}$$

That is, K is the feasible region of the usual LP-relaxation for MILP, and F is the set of feasible mixed-integer solutions. Let $\text{conv}(F)$ denote the convex hull of points in F. We assume that $x \geq 0$ and that K is bounded, and thus F is bounded. Under this assumption, without loss of generality, we assume that the bound constraints

$$\begin{aligned} & x_j \geq 0 && \text{for } j = 1, \ldots, n, \\ \text{and} \quad & x_j \leq M_j && \text{for } j = 1, \ldots, p, \end{aligned}$$

are included in the constraint set defining K.

For $\beta \in \mathbb{Z}$ and $j \in \{1, \ldots, p\}$, define

$$\mathcal{P}^j_\beta(K) = \text{conv}\{x \in K \mid x_j \leq \beta \text{ or } x_j \geq \beta + 1\}.$$

This variable disjunction is a natural generalization of the disjunction in Balas, Ceria and Cornuéjols [3]. The set $\mathcal{P}^j_\beta(K)$ can be described as follows:

$$\mathcal{P}^j_\beta(K) = \text{Proj}_x \left\{ (x, u^0, u^1, \lambda^0, \lambda^1) \;\middle|\; \begin{array}{ll} x = \lambda^0 u^0 + \lambda^1 u^1 & \lambda^0 + \lambda^1 = 1 \\ u^0 \in K,\ u^0_j \leq \beta & \lambda^0, \lambda^1 \geq 0 \\ u^1 \in K,\ u^1_j \geq \beta + 1 & \end{array} \right\}.$$

This approach for describing $\mathcal{P}^j_\beta(K)$ requires the solution of a related *projection problem* (for example, see [3]).

3. Convexification procedure

In this section, we develop a procedure for constructing a set arbitrarily close to $\text{conv}(F)$. First we demonstrate a sequential procedure for constructing the convex hull $\mathcal{C}_j(K) \equiv \text{conv}\{x \in K \mid x_j \in \mathbb{Z}\}$ for any given $j \in \{1, \ldots, p\}$. Then we extend and modify this single variable convexification procedure into an infinitely-convergent approach for generating $\text{conv}(F)$.

3.1. Single variable convexification

The following theorem provides a mechanism for generating $\mathcal{C}_j(K) \equiv \text{conv}\{x \in K | x_j \in \mathbb{Z}\}$ by taking the intersection of the sets $\mathcal{P}^j_\beta$ for arbitrary values of $\beta \in \{0, \ldots, M_j - 1\}$.

Theorem 1. *For $j \in \{1, \ldots, p\}$ and $L \subseteq \{0, \ldots, M_j - 1\}$,*

$$\bigcap_{i \in L} \text{conv}(K_i) = \text{conv}\left(\bigcap_{i \in L} K_i\right),$$

where $K_i \equiv \{x \in K \mid x_j \leq i \text{ or } x_j \geq i + 1\}$.

Proof. Without loss of generality, assume that $L = \{i_1, \ldots, i_q\}$ where $i_1 < \cdots < i_q$. The inclusion "$\supseteq$" is trivial: $K_i \subseteq \text{conv}(K_i) \Rightarrow \cap_i K_i \subseteq \cap_i \text{conv}(K_i) \Rightarrow \text{conv}(\cap_i K_i) \subseteq \cap_i \text{conv}(K_i)$, where the last implication follows since the intersection of convex sets is a convex set. To prove that $\cap_{i \in L} \text{conv}(K_i) \subseteq \text{conv}(\cap_{i \in L} K_i)$, select an arbitrary point $x \in \cap_{i \in L} \text{conv}(K_i)$. If $x \in \cap_{i \in L} K_i$ then the result follows. Otherwise, there must exist $i_t \in L$ such that $x \notin K_{i_t}$. Since $x \in K \backslash K_{i_t}$, we have that $i_t < x_j < i_t + 1$. Furthermore, since $x \in \text{conv}(K_{i_t})$ there exist $y, z \in K$ such that $y_j \leq i_t$, $z_j \geq i_t + 1$, and x belongs to the line segment (y, z). By moving towards x along the line segment (y, z), we can assume that $y_j = i_t$ and $z_j = i_t + 1$. It follows that both y and z are in $\cap_{i \in L} K_i$, thus we have that $x \in \text{conv}(\cap_{i \in L} K_i)$, and the result follows.

□

Corollary 1. *For $j \in \{1, \ldots, p\}$, $\beta \in \{0, \ldots, M_j - 1\}$, and $k \in \mathbb{Z}_+$,*

$$\bigcap_{i=\beta}^{\beta+k} \text{conv}(K_i) = \text{conv}\left(\begin{array}{c}\{x \in K \mid x_j \leq \beta\}, \{x \in K \mid x_j = \beta + 1\}, \ldots, \\ \{x \in K \mid x_j = \beta + k\}, \{x \in K \mid x_j \geq \beta + k + 1\}\end{array}\right),$$

where $K_i \equiv \{x \in K \mid x_j \leq i \text{ or } x_j \geq i + 1\}$.

□

We now state the main result of this section, which gives a sequential procedure for generating $\mathcal{C}_j(K)$.

Theorem 2. *For $j \in \{1, \dots, p\}$ and any permutation $\{i_1, i_2, \dots, i_{M_j}\}$ of $\{0, 1, \dots, M_j - 1\}$,*

$$\mathcal{C}_j(K) = \bigcap_{i \in \{0, \dots, M_j - 1\}} \operatorname{conv}(K_i) = \mathcal{P}^j_{i_1}\big(\mathcal{P}^j_{i_2}\big(\cdots\big(\mathcal{P}^j_{i_{M_j}}(K)\big)\cdots\big)\big),$$

where $K_i \equiv \{x \in K \mid x_j \le i \text{ or } x_j \ge i + 1\}$.

Proof. Let C be a linearly-constrained convex set. By definition of $\mathcal{P}^j_i(\cdot)$, we have that $\mathcal{P}^j_i(C)$ contains all vertices of C except for those fractional vertices whose j-th component is strictly between i and $i + 1$. Furthermore, when generating $\mathcal{P}^j_i(\cdot)$, no new vertices are added with fractional j-th components. It follows that

$$\mathcal{P}^j_{i_1}\big(\mathcal{P}^j_{i_2}\big(\dots\big(\mathcal{P}^j_{i_{M_j}}(K)\big)\dots\big)\big) = \operatorname{conv}\left(\begin{array}{c}\{x \in K \mid x_j \le 0\}, \{x \in K \mid x_j = 1\}, \dots, \\ \{x \in K \mid x_j = M_j - 1\}, \{x \in K \mid x_j \ge M_j\}\end{array}\right).$$

Then from Corollary 1, the result follows.

□

3.2. An ∞-convergent convexification procedure

In the previous section we have described an approach for generating the convex hull with respect to a single general-integer variable. In this section we describe a procedure based on the sequential application of this approach, and we show that the procedure converges in the limit to the convex hull of feasible mixed integer solutions for general MILP problems. Let $P^0 = K$, and define P^i for $i = 1, 2, 3, \dots$ as follows:

$$P^i = \mathcal{C}_1\big(\mathcal{C}_2\big(\cdots\big(\mathcal{C}_p\big(P^{i-1}\big)\big)\cdots\big)\big)$$

The main result of this section is Theorem 3, which states that $\lim_{i\to\infty} P^i = \operatorname{conv}(F)$.

Let $\{C^i\}$ be a sequence of compact nested sets. We say that $\{C^i\}$ *converges to* $\bar{C}$ (i.e., $\lim_{i\to\infty} C^i = \bar{C}$) if for any given $\epsilon > 0$ there exists an integer $\hat{k} \in \mathbb{Z}_+$ such that

$$\min_{x \in \bar{C}} \|x^k - x\| < \epsilon$$

for all $k \ge \hat{k}$ and any $x^k \in C^k$.

Lemma 1. *Let $\{C^i_1\}$ and $\{C^i_2\}$ be convergent sequences of nested compact sets. If $\lim_{i\to\infty} C^i_1 = \bar{C}_1$ and $\lim_{i\to\infty} C^i_2 = \bar{C}_2$, then*

$$\lim_{i\to\infty} \operatorname{conv}\big(C^i_1, C^i_2\big) = \operatorname{conv}(\bar{C}_1, \bar{C}_2).$$

Proof. Let $x \in \operatorname{conv}(\bar{C}_1, \bar{C}_2)$. Since $C^{i+1}_1 \subseteq C^i_1$ and $C^{i+1}_2 \subseteq C^i_2$ for all $i \ge 0$, it is clear that $x \in \operatorname{conv}(C^i_1, C^i_2)$ for all $i \ge 0$, which implies that $x \in \lim_{i\to\infty} \operatorname{conv}(C^i_1, C^i_2)$. It follows that $\operatorname{conv}(\bar{C}_1, \bar{C}_2) \subseteq \lim_{i\to\infty} \operatorname{conv}(C^i_1, C^i_2)$.

Let $x \in \lim_{i\to\infty} \text{conv}(C_1^i, C_2^i)$. By nestedness of the sets C_1^i and C_2^i, this implies that $x \in \text{conv}(C_1^i, C_2^i)$ for all $i \geq 0$. It follows that there exist sequences

$$\{x^{i_1}\} \subseteq C_1^i, \{x^{i_2}\} \subseteq C_2^i, \text{ and } \{\lambda_i\} \subseteq [0,1] \tag{1}$$

such that $\{\lambda_i x^{i_1} + (1-\lambda_i)x^{i_2}\} \to x$. Since the sequences in (1) are bounded, there exist subsequences such that $\{x^{i_1}\} \to x^1 \in \bar{C}_1$, $\{x^{i_2}\} \to x^2 \in \bar{C}_2$, and $\{\lambda_i\} \to \lambda \in [0,1]$. Thus $x \in \text{conv}(\bar{C}_1, \bar{C}_2)$, and the result follows.

□

Theorem 3. *For $i = 0, 1, 2, \ldots$ and P^i defined as above,*

$$\lim_{i\to\infty} P^i = \text{conv}(F) = \text{conv}\{x \in K \mid x_j \in \mathbb{Z} \text{ for } j = 1, \ldots, p\}.$$

Proof. Since $\{P^i\}$ is a sequence of compact convex sets such that $\text{conv}(F) \subseteq P^{i+1} \subseteq P^i$ for all $i \geq 0$, we know that $\{P^i\}$ converges to some set $\hat{P}$ such that $\text{conv}(F) \subseteq \hat{P}$. By construction,

$$P^{i+1} \subseteq \text{conv}\left(\left\{x \in P^i \mid x_j \leq \beta\right\} \cup \left\{x \in P^i \mid x_j \geq \beta + 1\right\}\right)$$

for all $j \in \{1, \ldots, p\}$ and any integer β. Taking the limit and applying Lemma 1, we have that

$$\begin{aligned}
\hat{P} = \lim_{i\to\infty} P^{i+1} &\subseteq \lim_{i\to\infty} \text{conv}\left(\left\{x \in P^i \mid x_j \leq \beta\right\} \cup \left\{x \in P^i \mid x_j \geq \beta + 1\right\}\right) \\
&= \text{conv}\left(\lim_{i\to\infty} \left\{x \in P^i \mid x_j \leq \beta\right\} \cup \lim_{i\to\infty} \left\{x \in P^i \mid x_j \geq \beta + 1\right\}\right) \\
&= \text{conv}\left(\left\{x \in \hat{P} \mid x_j \leq \beta\right\} \cup \left\{x \in \hat{P} \mid x_j \geq \beta + 1\right\}\right),
\end{aligned}$$

for all $j \in \{1, \ldots, p\}$ and any integer β. It follows that

$$\hat{P} = \text{conv}\{x \in K \mid x_j \in \mathbb{Z} \text{ for } j = 1, \ldots, p\}.$$

□

Theorem 3 shows that the sequential disjunctive procedure yields the convex hull of mixed-integer solutions in the limit. This is a weaker result than that obtained for the binary-integer case [3]; this is due to the fact that the inequalities arising from simple variable disjunctions are not facial for general-integer problems [2].

4. A cutting plane algorithm for MILP

In the previous section we described a procedure for generating in the limit the convex hull of solutions to MILP. Rather than generating the entire hull $\text{conv}(F)$, it is desirable to use constraints from this description of the hull as cutting planes. Although these constraints are not immediately available, it is natural to consider using the variable disjunctions described in Sect. 2 in a cutting plane procedure for solving MILP. The usual approach for such a procedure involves generating cutting planes by using variable

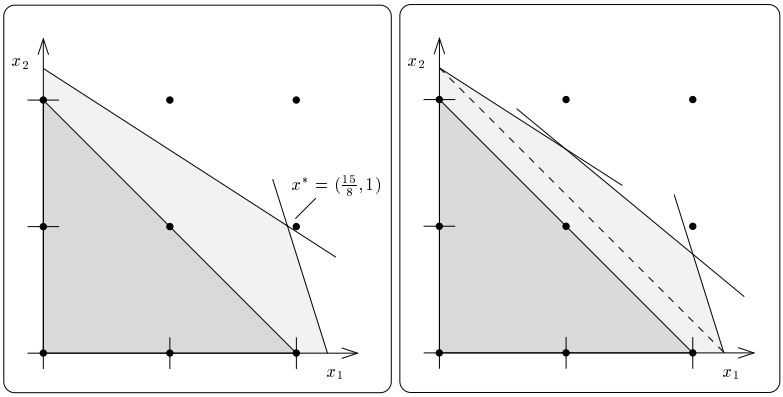
disjunctions obtained from a fractional component of the current optimal solution. It has been shown by Balas, Ceria, and Cornuéjols (see [3]) that such a procedure is sufficient to solve MILP when integer variables are restricted to binary values. A simple example, however, shows that such a procedure is not sufficient in the case where variables can take on general-integer values. The remainder of this section is in two parts. In the first part, we give a two-variable example that demonstrates the inadequacy of the straightforward variable disjunction cutting plane procedure. In the second part of this section we describe a modified cutting plane procedure that generates inequalities at multiple vertices of the incumbent LP-relaxation. We prove that this modified cutting plane procedure is guaranteed to find an ε-optimal solution finitely (or terminate finitely if no such solution exists).

4.1. Geometrical example

We motivate the need for our modified cutting plane algorithm with a simple two-variable example. Consider the region

$$K = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{array}{rl} 8x_1 + 12x_2 \leq & 27 \\ 8x_1 + 3x_2 \leq & 18 \\ x_1 \geq & 0 \\ x_2 \geq & 0 \end{array} \right\},$$

the set $F = \{x \in K \mid x_1, x_2 \in \mathbb{Z}\}$, and the cost vector $c = [-1, -1]$. The regions K and $\text{conv}(F)$ are pictured in Fig. 1a. Observe that minimizing $c^T x$ over the region K yields an optimal solution which is fractional (see the solution x^* in Fig. 1a).



(a) The regions K and F (b) Resulting region after first cutting plane

Fig. 1. Motivation for the disjunctive cutting plane procedure

Consider a straightforward cutting plane procedure for MILP where all cutting plane inequalities are generated from variable disjunctions that are violated at an optimal

solution to the current LP-relaxation. We leave it to the reader to see that such a procedure will not converge to an integer solution for the given example. In particular, such a procedure will not cut away the fractional extreme points (2.25, 0) or (0, 2.25). The geometrical observations that are central to the needed arguments are as follows:

1. Using the deepest cuts from simple variable disjunctions, it is only possible to cut away the extreme point (2.25, 0) by using the variable disjunction "$x_1 \leq 2$ or $x_1 \geq 3$." Similarly, the extreme point (0, 2.25) can only be eliminated by using the variable disjunction "$x_2 \leq 2$ or $x_2 \geq 3$."
2. Since the cutting plane procedure generates inequalities only at an optimal (fractional) solution, the points (2.25, 0) and (0, 2.25) can only be cut away if a current optimal solution $x^* = (x_1^*, x_2^*)$ is used to generate inequalities, where $x_1^* \in (2, 3)$ or $x_2^* \in (2, 3)$ respectively.
3. Each cutting plane inequality introduces at least one new extreme point $\hat{x}$ such that $\hat{x}_1 + \hat{x}_2 > 2.25$. Furthermore, for each new extreme point $\hat{x}$, either
   (a) $\hat{x}_1 \in \{1, 2\}$ and $\hat{x}_2 \notin (2, 3)$, or
   (b) $\hat{x}_1 \notin (2, 3)$ and $\hat{x}_2 \in \{1, 2\}$.

Thus it follows that neither (2.25, 0) nor (0, 2.25) will be optimal solutions to the LP-relaxation in any iteration (and thus cannot be cut away directly), and that every optimal solution x^* to the LP-relaxation will be fractional with $x_1^* \in [0, 2]$ and $x_2^* \in [0, 2]$. Note that in the limit the LP-relaxation will converge to the region $\{x \in K \mid x_1 + x_2 \leq 2.25\}$.

4.2. The cutting plane algorithm

In this section we describe a modified cutting plane algorithm for solving MILP. In contrast to the usual cutting plane approach, which generates inequalities only at the current optimal solution, the modified algorithm generates inequalities at multiple vertices of the incumbent LP-relaxation at each iteration. These vertices are generated by exploring a set of near-optimal solutions of the current relaxation.

Cutting planes in the algorithm are generated based on the simple variable disjunctions described in Sect. 2. In particular, given a relaxation S and a vertex solution $\bar{x}$ that is fractional in the j-th component, we solve the separation problem

$$\begin{aligned} \min \quad & \|x - \bar{x}\| \\ \text{s. t.} \quad & x \in \mathcal{P}^j_{\lfloor \bar{x}_j \rfloor}(S), \end{aligned} \tag{2}$$

and generate the inequality

$$\xi^T x \geq \xi^T x^*, \tag{3}$$

where x^* is an optimal solution of (2), ξ is a subgradient of $\|x - \bar{x}\|$ at x^*, and $\|\cdot\|$ is any norm function. The fact that (3) provides a valid cutting plane follows from Theorem 3.4.3 of [6].

In addition to the problem information (c and K), our cutting plane procedure requires three input parameters, $\gamma > 0$, $\varepsilon > 0$, and $\bar{\varepsilon} \in (0, .5]$. The value γ determines the solution space that we explore at each iteration, and the value ε determines our optimality

tolerance. The parameter $\bar{\varepsilon}$ is used for integer rounding. A pseudocode description of the algorithm is given in Fig. 2. The main result of this section is Theorem 4, in which we prove that this modified cutting plane algorithm is guaranteed to find an ε-optimal integer solution finitely, or that it will terminate finitely if no such solution exists. For a convex set S, let ext(S) represent the set of extreme points of S.

```
algorithm disjunctive cutting planes (c, K, γ, ε, ε̄)
{
    i := 0; S^0 := K; /* initialization */

    while (S^i ≠ ∅) {
        let z^i be the optimal objective value of min_{x∈S^i} c^T x;
        Ω^i := {x ∈ ext(S^i) | c^T x − z^i ≤ γ};

        /* check for ε-optimal solution */
        Ω_ε̄ := {x ∈ Ω^i | min{(x_j − ⌊x_j⌋), (⌈x_j⌉ − x_j)} < ε̄  ∀ j ∈ {1,2,...,p}};
        if (Ω_ε̄ ≠ ∅) {
            /* look for close points in F */
            for each (x ∈ Ω_ε̄) {
                /* rounding procedure */
                for each (j ∈ {1,...,p}) {
                    if (x_j − ⌊x_j⌋ < ε̄) then d_j := ⌊x_j⌋;
                    else d_j := ⌈x_j⌉;
                }

                /* fix first p components and reoptimize */
                let S̄ := {x ∈ S^i | x_j = d_j   for j = 1,2,...,p};
                if (S̄ ≠ ∅) then {
                    let x̄ be an optimal extreme point solution of min_{x∈S̄} c^T x;
                    if (c^T x̄ − z^i < ε) then STOP; /* x̄ is ε-optimal */
                }
            }
        }

        /* generate cutting planes */
        S^{i+1} := S^i; /* initialize region S^{i+1} */
        for each (x̄ ∈ Ω^i and j ∈ {1,...,p} such that x̄_j ∉ ℤ) {
            if ({x ∈ S^i | x_j ≤ ⌊x̄_j⌋} = {x ∈ S^i | x_j ≥ ⌈x̄_j⌉} = ∅) then STOP; /* infeasible */

            /* generate a cutting plane and add it to S^{i+1} */
            let x* be an optimal solution of min_{x∈P^j_{⌊x̄_j⌋}(S^i)} ‖x − x̄‖;
            let ξ be some subgradient of ‖x − x̄‖ at x*;
            S^{i+1} := {x ∈ S^{i+1} | ξ^T x ≥ ξ^T x*}; /* update region S^{i+1} */
        }
        i := i + 1;
    }
    STOP; /* infeasible (since S^i = ∅) */
}
```

Fig. 2. Pseudocode description of disjunctive cutting plane procedure

Lemma 2. *Let $\{S^i\}$ be a convergent sequence of bounded convex sets such that $S^{i+1} \subseteq S^i$ for all $i \geq 0$ and $\lim_{i\to\infty} S^i = \hat{S}$. For each $\hat{x} \in \text{ext}(\hat{S})$, there exists some sequence $\{x^i\}$ of points in $\text{ext}(S^i)$ with a subsequence converging to $\hat{x}$.*

Proof. Consider $\hat{x} \in \text{ext}(\hat{S})$ and a sequence $\{x^i\} \to \hat{x}$ such that $x^i \in S^i$ for all $i \geq 0$. Write x^i as follows:

$$x^i = \sum_{j=1}^{n+1} \lambda_{ij} x^{ij}, \text{ where } \lambda_{ij} \geq 0, \sum_{j=1}^{n+1} \lambda_{ij} = 1, \text{ and } x^{ij} \in \text{ext}(S^i). \tag{4}$$

Each of the sequences in (4) is bounded and thus have convergent subsequences. Passing to subsequences, we have that

$$\hat{x} = \sum_{j=1}^{n+1} \lambda_j x^j, \text{ where } \{\lambda_{ij}\} \to \lambda_j \geq 0, \sum_{j=1}^{n+1} \lambda_j = 1, \text{ and } \{x^{ij}\} \to x^j \in \hat{S}.$$

Since $\hat{x}$ is an extreme point of $\hat{S}$ it can not be written as a convex combination of other points in $\hat{S}$, thus $\{x^{ij}\} \to x^j = \hat{x}$ for some $j \in \{1, \ldots, n\}$ and the result follows.

□

Theorem 4. *The algorithm either generates an ε-optimal solution to MILP or detects infeasibility (i.e., $F = \emptyset$) in a finite number of iterations.*

Proof. For input γ, ε, $\bar{\varepsilon}$, c, and K, let S^i, z^i, and Ω^i be defined as in the algorithm. Assume, by contradiction, that the algorithm does not terminate in a finite number of iterations. Then, since S^0 $(= K)$ is bounded, and since $S^{i+1} \subseteq S^i$ for all $i \geq 0$, the sequence $\{S^i\}$ is convergent to some set, say $\hat{S}$. The remainder of this proof is given in two parts; we first consider the case where $\hat{S} = \emptyset$, then we proceed to the case where $\hat{S}$ is non-empty. In both cases we show that since $\{S^i\} \to \hat{S}$ the procedure will terminate finitely, a contradiction.

First consider the case where $\{S^i\} \to \hat{S} = \emptyset$. If there exists some finite integer k such that $S^k = \emptyset$, then the procedure will terminate (at line 41 of Fig. 2). Otherwise, there exists a finite integer k such that $\{x \in S^k \mid x_j \in \mathbb{Z}\} = \emptyset$ for some $j \in \{1, \ldots, p\}$. Let $\hat{j}$ be such an index. Thus for all $x \in \Omega^k$ we have that

$$P^{\hat{j}}_{\lfloor x_{\hat{j}} \rfloor}(S^k) \equiv \left\{ y \in S^k \mid y_{\hat{j}} \leq \lfloor x_{\hat{j}} \rfloor \right\} = \emptyset$$

$$\text{and} \qquad P^{\hat{j}}{}_{\lfloor x_{\hat{j}} \rfloor +}(S^k) \equiv \left\{ y \in S^k \mid y_{\hat{j}} \geq \lfloor x_{\hat{j}} \rfloor + 1 \right\} = \emptyset.$$

It follows that the procedure will terminate execution in the k-th iteration (at line 32 in Fig. 2). We have thus shown that if $\{S^i\} \to \hat{S} = \emptyset$, then the algorithm will terminate finitely after detecting infeasibility.

Now consider the case where $\{S^i\} \to \hat{S} \neq \emptyset$. In this case let $\hat{x}$ be an optimal extreme point solution to the optimization problem

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && x \in \hat{S}. \end{aligned}$$

Since $\{S^i\} \to \hat{S}$ we have that $\{z^i\} \to c^T \hat{x}$. Furthermore, by Lemma 2 there exists a convergent subsequence of points $\{x^i\} \to \hat{x}$ where $x^i \in \text{ext}(S^i)$. We now examine separately the case where $\hat{x} \notin F$ and the case where $\hat{x} \in F$.

In the case where $\hat{x} \notin F$, there must exist some $j \in \{1, \ldots, p\}$ such that $\hat{x}_j \notin \mathbb{Z}$. Since $\hat{x} \in \text{ext}(\hat{S})$ we know that $\delta > 0$, where δ is defined as follows:

$$\delta \equiv \min_{x \in \mathcal{P}^j_{\lfloor \hat{x}_j \rfloor}(\hat{S})} \|\hat{x} - x\|.$$

Thus since $\{S^i\} \to \hat{S} \neq \emptyset$ and since there exists a convergent subsequence $\{x^i\} \to \hat{x}$ where $x^i \in \text{ext}(S^i)$, it follows that there exists a finite integer k such that

1. $x^k \in \Omega^k$,
2. $\lfloor x^k_j \rfloor = \lfloor \hat{x}_j \rfloor$,
3. $\|\hat{x} - x^k\| < \frac{\delta}{2}$, and
4. $\min_{x \in \mathcal{P}^j_{\lfloor x^k_j \rfloor}(S^k)} \|x^k - x\| > \frac{\delta}{2}$.

It follows that in step k the algorithm will generate a valid inequality (lines 29–38 of Fig. 2) that is violated by $\hat{x}$. As a result, $\hat{x} \notin S^{k+1}$, contradicting $\hat{x} \in \hat{S} \subseteq S^{k+1}$.

We now consider the case where $\hat{x} \in F$. Since $\{S^i\} \to \hat{S} \neq \emptyset$ and since there exists a convergent subsequence $\{x^i\} \to \hat{x}$ where $x^i \in \text{ext}(S^i)$, it follows that there exists a finite integer k such that

1. $x^k \in \Omega^k$,
2. $|x^k_j - \hat{x}_j| < \bar{\varepsilon}$ for all $j \in \{1, \ldots, p\}$, which implies that $x^k \in \Omega_{\bar{\varepsilon}}$ in the k-th iteration, and
3. $c^T \hat{x} - z^k < \varepsilon$.

Thus

$$z^k \leq \min_{\{x \in S^k | x_j = \hat{x}_j \ \forall j \in \{1, \ldots, p\}\}} c^T x \leq c^T \hat{x}, \tag{5}$$

where the optimization problem in the middle of (5) is solved in step 23 of the k-th iteration. Since this will yield a solution $\bar{x}$ such that $c^T \bar{x} - z^K < \varepsilon$, the algorithm will terminate in the k-th iteration (line 24) with an ε-optimal solution to MILP.

□

The modified algorithm requires that the set of γ-optimal solutions, Ω^i, be generated at each iteration. In theory, this set can be generated by first finding the optimal solution to the LP-relaxation and then performing an exhaustive recursive search through the neighboring γ-optimal vertex solutions obtained by using single Simplex pivots. In practice, the search can be terminated with a subset of γ-optimal solutions. The practical performance of our algorithm will depend on the choice of γ. For larger choices of γ we obtain more rapid convergence of S^i to S at the cost of exploring more vertices and generating larger LP subproblems. Computational experiments are required in order to determine the degree to which the best choice of γ is problem dependent.

Practical implementations of the branch-and-cut algorithm typically decide to branch when improvements in the objective value tail-off. The example given in Sect. 4.1 indicates one possible reason for such tail-off behavior. As seen from the analysis of our modified algorithm, generating inequalities at near-optimal vertices results in a convergent procedure without branching. This suggests that when practical implementations

of branch-and-cut procedures encounter tail-off in the objective value improvement, generating cuts at near-optimal vertices may be a viable alternative to delay (or avoid) branching.

5. Conclusions

In this paper we have studied the natural single-variable disjunction. We have shown that this disjunction can be sequentially applied to find the convex hull with respect to a single general-integer variable finitely, or the convex hull of all mixed-integer solutions in the limit. In the case where all integer variables are restricted to taking binary values, this procedure is equivalent to the sequential convexification procedure of Balas, Ceria, and Cornuéjols [3].

We have also developed a finitely convergent cutting plane algorithm for solving general mixed integer linear program, MILP. To the best of our knowledge this is first such algorithm. The convergence results are obtained by generating cuts at all γ-optimal vertices of the relaxation of MILP at each iteration. New cuts in this algorithm are generated by solving a separation problem over a region that includes all cutting planes found thus far. This is different from the strategy used by Balas, Ceria, and Cornuéjols to obtain their convergence result for the 0-1 specialized cutting plane algorithm, as described in [3], and is closer to the strategy implemented in practice.

As a result of our analysis we provide a theoretical foundation to the fundamental idea of generating cuts at near optimal, possibly alternate optimal, vertices. Although the analysis of the algorithm is based on generating cuts at all γ-optimal vertices, a practical implemenation of our algorithm is likely to generate cuts only at a few "promising" near-by vertices.

Practical implementations of the branch-and-cut algorithm for structured binary problems typically decide to branch when improvements in the objective value tail off. Analysis of our modified cutting plane algorithm suggests that generating cuts at neighboring vertices in addition to the current optimal vertex may be a viable alternative to branching, or delaying branching in these implementations. Detailed computational experiments are necessary to determine if practical implementations would benefit from incorporating this modification.

Acknowledgements. We would like to thank the two referees and the associate editor handling this paper for their helpful comments. In particular, we would like to acknowledge Referee #2 for suggestions on simplifying several proofs and help towards making the paper more readable.

References

1. Adams, W.P., Sherali, H.D. (1995): Explicit convex hull representations of discrete sets. Talk at Fall INFORMS meeting, New Orleans, LA, USA
2. Balas, E. (1979): Disjunctive programming. In: Hammer, ed., Discrete Optimization, Part II. Ann. Discrete Math. **5**, 3–51
3. Balas, E., Ceria, S., Cornuéjols, G. (1993): A lift-and-project cutting plane algorithm for mixed 0-1 programs. Math. Program. **58**, 295–324
4. Balas, E., Ceria, S., Cornuéjols, G. (1996): Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. Manage. Sci. **42**, 1229–1246

5. Balas, E., Ceria, S., Cornuéjols, G., Pataki, G. (1994): Polyhedral methods for the maximum clique problem. Unpublished paper. Carnegie-Mellon University, Graduate School of Industrial Administration, Pittsburgh, PA 15213, February 1994
6. Bazaraa, M.S., Sherali, H.D., Shetty, C.M. (1993): Nonlinear Programming, Theory and Algorithms, 2nd edn. John Wiley & Sons, Inc., New York
7. Lovász, L., Schrijver, A. (1991): Cones of matrices and set-functions and 0-1 optimization. SIAM J. Optim. **1**, 166–190
8. Sherali, H.D., Adams, W.P. (1990): A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM J. Discrete Math. **3**, 411–430
9. Sherali, H.D., Adams, W.P. (1994): A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. Discrete Appl. Math. **52**, 83–106
10. Stubbs, R.A., Mehrotra, S. (1999): A branch-and-cut method for 0-1 mixed convex programming. Math. Program. A **86**, 515–532