Mathematical Programming 30 (1984) 22–45
North-Holland

NONLINEAR 0–1 PROGRAMMING: II. DOMINANCE RELATIONS AND ALGORITHMS

Egon BALAS*
Carnegie-Mellon University, Pittsburgh, PA, USA

Joseph B. MAZZOLA
University of North Carolina at Chapel Hill, NC, USA

Received 17 August 1981
Revised manuscript received 21 September 1983

A nonlinear 0–1 program can be restated as a multilinear 0–1 program, which in turn is known to be equivalent to a linear 0–1 program with generalized covering (g.c.) inequalities. In a companion paper [6] we have defined a family of linear inequalities that contains more compact (smaller cardinality) linearizations of a multilinear 0–1 program than the one based on the g.c. inequalities. In this paper we analyze the dominance relations between inequalities of the above family. In particular, we give a criterion that can be checked in linear time, for deciding whether a g.c. inequality can be strengthened by extending the cover from which it was derived. We then describe a class of algorithms based on these results and discuss our computational experience. We conclude that the g.c. inequalities can be strengthened most of the time to an extent that increases with problem density. In particular, the algorithm using the strengthening procedure outperforms the one using only g.c. inequalities whenever the number of nonlinear terms per constraint exceeds about 12–15, and the difference in their performance grows with the number of such terms.

Key words: Nonlinear 0–1 Programming Algorithm, Covering Inequalities, Dominance, Compact Linear Equivalent, Strengthening 0–1 Inequalities.

1. Introduction

In this paper we discuss solution methods for the multilinear 0–1 programming problem

$$\max\{f_0(x) \mid f_k(x) \leqslant b_k,\ k \in K,\ x \text{ binary}\} \tag{MLP}$$

where the functions f_0 and f_k, $k \in K$, are of the general form

$$f(x) = \sum_{j \in N} a_j \left(\prod_{i \in Q_j} x_i \right). \tag{1}$$

Here a_j, $j \in N$, are real numbers, and $\prod$ means product. Any nonlinear 0–1 program involving real-valued functions can be restated in this form [14].

Applications of nonlinear 0–1 programming span many areas. Such formulations have been used in modular design [7, 8], media selection [22], project scheduling

* Research supported by the National Science Foundation under grant ECS7902506 and by the Office of Naval Research under contract N00014-75-C-0621 NR 047-048.

[17], capital budgeting under uncertainty [16], cluster analysis [18], diagnostic testing [15], accounting control systems [13], production planning in flexible manufacturing systems [19, 20], etc. Quadratic 0–1 programming, including the quadratic assignment problem, has a host of well known uses.

In a companion paper [6] we have introduced a new linearization of (MLP) (first presented in [4]), which uses only the original variables. Specifically, for a multilinear inequality of the form

$$f(x) = \sum_{j \in N} a_j \left(\prod_{i \in Q_j} x_i \right) \leq b, \tag{2}$$

we defined a family $\mathcal{F}$ of linear inequalities equivalent to (2) in the sense that a 0–1 vector x satisfies (2) if and only if it satisfies every inequality of $\mathcal{F}$, and we identified several proper subfamilies of $\mathcal{F}$ that are also linear eqivalents of (2). The members of $\mathcal{F}$ are associated with covers for (2), and the subfamily of $\mathcal{F}$ corresponding to minimal covers is the set of generalized covering inequalities (set covering inequalities in the original variables and their complements) shown by Granot and Hammer [12] to be equivalent to (2). Some other subsets of $\mathcal{F}$, associated with covers for (2) that are not minimal, give more compact linearizations, i.e., linear equivalents of smaller cardinality.

In the remainder of this section we restate those results of [6] that we will need in the sequel. In Section 2 we examine dominance relations between inequalities of $\mathcal{F}$, with a view toward deriving criteria for generating linear equivalents of (2) as compact as possible. We first give a necessary and sufficient condition for an inequality of $\mathcal{F}$ to imply another one (with respect to binary vectors). We then show that a generalized covering inequality in $\mathcal{F}$, i.e., a member of $\mathcal{F}$ associated with a minimal cover M for (2), can be strengthened by including into M certain indices $j \in N \backslash M$ if and only if an easily verifiable condition holds.

In Section 3 we introduce a class of algorithms for solving multilinear 0–1 programs, based on these results. Like the earlier procedure of Granot and Granot [9] (see also [10, 11]), our algorithms generate linear inequalities sequentially from those constraints of (MLP) violated by the current solution until such time when an optimal binary solution to the current linear constraint set satisfies all constraints of (MLP); such a solution is optimal for (MLP). However, while the procedure of [9, 10, 11] uses generalized covering inequalities only, the main version of our algorithm generates stronger inequalities whenever they are obtainable via the criteria outlined above. Our algorithms use as a subroutine pivot and complement the 0–1 (linear) programming heuristic of Balas and Martin [3].

Section 4 discusses our computational experience on randomly generated multilinear 0–1 programs with up to 20 constraints, 50 variables, and 60 nonlinear terms per constraint. In particular, an algorithm that uses the results of this paper to generate strengthened linear inequalities is compared to one that uses only generalized covering inequalities, like the procedure of [9, 10, 11]. The algorithm that uses only generalized covering inequalities did better on problems that had on the average

less than 12 nonlinear terms per constraint, whereas the one that uses the strengthened inequalities did better on the problems that had on the average 16 or more nonlinear terms per constraint. Furthermore, the difference in performance tends to increase sharply with the number of terms per constraint.

We also present computational results concerning the use of the procedure as a heuristic. Typically, the heuristic solutions obtained were (guaranteed to be) within 3% of optimality, and for those cases in which the optimal solution was known, the heuristic solution was on the average within 0.25% of the optimal integer solution. In the heuristic mode, our algorithm is able to solve substantially larger problems than those noted above, and does so in a reasonable amount of time with practically acceptable accuracy bounds.

The results of this paper were circulated under [5].

Given a multilinear inequality (2), let

$$N^+ = \{j \in N \mid a_j > 0\}, \qquad N^- = \{j \in N \mid a_j < 0\},$$

and

$$f^+(x) = \sum_{j \in N^+} a_j \left(\prod_{i \in Q_j} x_i \right), \qquad f^-(x) = \sum_{j \in N^-} a_j \left(\prod_{i \in Q_j} x_i \right).$$

For any $M \subseteq N$, let $Q_M = \bigcup_{j \in M} Q_j$, $Q = Q_N$, and $q = |Q|$. Let Φ^- be the family of mappings φ that associate to every $j \in N^-$ some $i \in Q_j$, and for each $\varphi \in \Phi$, let

$$h^-_\varphi(x) = \sum_{j \in N^-} a_j x_{\varphi(j)}.$$

For any $\varphi \in \Phi^-$, let Q_φ be the range of φ, i.e.,

$$Q_\varphi = \{i \in Q \mid i = \varphi(j) \text{ for some } j \in N^-\}.$$

A set $M \subseteq N$ is said to be a *cover* for the inequality (2) if

$$\sum_{j \in M} |a_j| > b - \sum_{j \in N^-} a_j.$$

A cover M is *minimal* if T is not a cover for any $T \subsetneq M$.

Thus, a set $M \subseteq N^+$ is a cover for the inequality

$$f^+(x) \leqslant b \tag{2^+}$$

if and only if

$$\sum_{j \in M} a_j > b,$$

i.e., if and only if $M \cup N^-$ is a cover for (2). We will denote

$$\mathscr{C} = \{M \subseteq N^+ \mid M \text{ is a cover for } (2^+)\}.$$

For any $x_i \in \{0, 1\}$, the *complement* of x_i is defined to be $\bar{x}_i = 1 - x_i$.

The following result (Theorem 11 of [6]), presented here without proof, is fundamental in defining the family $\mathscr{F}$ of linear inequalities equivalent to the multilinear inequality (2).

Theorem 1. *The vector* $x \in \{0,1\}^q$ *satisfies* (2) *if and only if it satisfies*

$$\sum_{i \in Q_M} \alpha_i^M \bar{x}_i + \sum_{i \in Q_\varphi} \beta_i^\varphi x_i \geq \alpha_0^M \qquad (3)_{M,\varphi}$$

for every $M \in \mathscr{C}$, $\varphi \in \Phi^-$, *where*

$$\alpha_0^M = \sum_{j \in M} a_j - b, \qquad \alpha_i^M = \min\left\{\alpha_0^M, \sum_{j \in M \mid i \in Q_j} a_j\right\}, \quad i \in Q_M$$

and

$$\beta_i^\varphi = \min\left\{\alpha_0^M, \sum_{j \in N^- \mid i = i(j)} |a_j|\right\}, \quad i \in Q_\varphi.$$

Denoting by $\mathscr{S}$ the family of inequalities $(3)_{M,\varphi}$ for all $M \in \mathscr{C}$ and $\varphi \in \Phi^-$, Theorem 1 states that the multilinear inequality (2) is equivalent to the system of linear inequalities $\mathscr{S}$. Note that all coefficients of the inequalities of $\mathscr{S}$ are nonnegative.

Now let $M \in \mathscr{C}$ and $\varphi \in \Phi^-$ be such that $Q_M \cap Q_\varphi \neq \emptyset$, i.e., there exists some $i \in Q_M \cap Q_\varphi$, for which both $\bar{x}_i$ and x_i have positive coefficients in $(3)_{M,\varphi}$. Then $(3)_{M,\varphi}$ remains essentially the same (in the sense of having the same solution set) if $\min\{\alpha_i^M, \beta_i^\varphi\}$ is subtracted from α_i^M, from β_i^φ and from α_0^M. But then all those remaining coefficients α_j^M, $j \neq i$ and β_k^φ, $k \neq i$, whose value exceeds the new right hand side, $(\alpha_0^M)' = \alpha_0^M - \min\{\alpha_i^M, \beta_i^\varphi\}$, can be replaced by $(\alpha_0^M)'$. Thus the presence of indices $i \in Q_M \cap Q_\varphi$ allows an immediate (trivial) strengthening of the inequality $(3)_{M,\varphi}$.

For any $\varphi \in \Phi^-$, if $M \in \mathscr{C}$ is such that the set $M \cup N^-$ is a minimal cover for (2), then the linear inequality $(3)_{M,\varphi}$ resulting from Theorem 1 can be shown (see [6, Theorem 13]) to be of the form

$$\sum_{i \in Q_M} \bar{x}_i + \sum_{i \in Q_\varphi} x_i \geq 1. \qquad (4)_{M,\varphi}$$

Furthermore, for any such set M, if M and φ are such that $Q_M \cap Q_\varphi \neq \emptyset$, the inequality $(4)_{M,\varphi}$ is vacuous. Thus the only inequalities $(4)_{M,\varphi}$ of interest are those such that $Q_M \cap Q_\varphi = \emptyset$.

The multilinear inequality (2) was also shown in [6] to be equivalent to the system of multilinear inequalities (each having all positive coefficients) defined by

$$f^+(x) + \sum_{j \in N^-} |a_j| \bar{x}_{\varphi(j)} \leq b - \sum_{j \in N^-} a_j, \qquad (5)_\varphi$$

for all $\varphi \in \Phi^-$. Thus, replacing (2) by all inequalities $(5)_\varphi$, $\varphi \in \Phi^-$, and then generating the sets $\mathscr{S}_\varphi$ of linear inequalities equivalent to each inequality $(5)_\varphi$, results in a family $\mathscr{F} = \bigcup_{\varphi \in \Phi} \mathscr{S}_\varphi$ of linear inequalities which linearizes (2). In fact, we observed

in [6] that the previously defined family $\mathcal{S}$ is properly contained in $\mathcal{F}$; namely, while $\mathcal{F}$ contains an inequality for every cover of (2), $\mathcal{S}$ contains an inequality only for every cover of (2^+).

The inequalities in $\mathcal{F}$ corresponding to minimal covers of (2) are the generalized covering inequalities known to be equivalent to (2). Thus all members of $\mathcal{F}$ that are not generalized covering inequalities correspond to covers of (2) that are not minimal.

We are now prepared to investigate the relative strength of inequalities $(3)_{M,\varphi}$ derived from covers M that are minimal and those that are not.

2. Dominance relations

An inequality A is said to *dominate* an inequality B if every nonnegative x satisfying A also satisfies B. Further, A *strictly* dominates B, if A dominates B and there exists some nonnegative $\tilde{x}$ that satisfies B but not A. We shall also find it useful to define the following weaker notion of dominance.

An inequality A is said to *c-dominate* an inequality B if every 0–1 point x satisfying A also satisfies B. Further, A *strictly* c-dominates B, if A c-dominates B and there exists some 0–1 point satisfying B but not A. It is easily verified that an inequality A can c-dominate an inequality B without A dominating B, whereas the converse is of course false. It is also easily verified that A c-dominates B if and only if every cover for B is a cover for A, and A strictly c-dominates B if and only if A c-dominates B and there exists a cover for A that is not a cover for B. Hence the term c-dominance.

We will occasionally call an inequality A *stronger* than B if A strictly c-dominates B.

We have seen that the inequalities of the family $\mathcal{F}$ are intimately related to covers for the multilinear inequality (2). In the context of linear inequalities, it is known [1, 2] that canonical inequalities derived from minimal covers can usually be strengthened, and can never be weakened by extending the covers. Unfortunately in the case of nonlinear inequalities, only the first part of this statement is true: extending a minimal cover may weaken the inequality associated with it.

Example 1. To show that extending a minimal cover can actually weaken the inequality derived from the cover, let

$$7x_2x_5x_6+6x_1x_3x_4+5x_2x_4+2x_1x_3\leqslant 12,$$

with $Q_1=\{2,5,6\}$, $Q_2=\{1,3,4\}$, $Q_3=\{2,4\}$ and $Q_4=\{1,3\}$. Applying Theorem 1 and using the minimal cover $M=\{2,3,4\}$, we obtain the inequality

$$\bar{x}_1+\bar{x}_2+\bar{x}_3+\bar{x}_4\geqslant 1.$$

Now, extending the minimal cover M to $\{1, 2, 3, 4\}$, we obtain the inequality

$$8\bar{x}_1 + 8\bar{x}_2 + 8\bar{x}_3 + 8\bar{x}_4 + 7\bar{x}_5 + 7\bar{x}_6 \geq 8,$$

which is actually weaker than (strictly c-dominated by) the first inequality. □

Fortunately, the phenomenon illustrated by Example 1 can be precisely characterized. Next we address the practically important question as to when an inequality $(3)_{M,\varphi}$, where $M \in \mathscr{C}$, can be strengthened by expanding the cover M.

We will assume that $\varphi \in \Phi^-$ is given, and therefore will write $(3)_M$ for $(3)_{M,\varphi}$ and β_i for β_i^φ. From the discussion at the end of Section 1 it should be clear that the presence of indices $i \in Q_M \cap Q_\varphi$, denoting the presence of positive coefficients for both x_i and $\bar{x}_i$, denotes a 'weakness' of the inequality $(3)_M$, in that it allows for a trivial strengthening. We will therefore assume that M is chosen such that $Q_M \cap Q_\varphi = \emptyset$, and furthermore that M is expanded into a set R such that $Q_R \cap Q_\varphi = \emptyset$ too.

In what follows, summation over the empty set is taken to yield 0.

First we give a necessary and sufficient condition for an inequality

$$\sum_{i \in Q_R} \alpha_i^R \bar{x}_i + \sum_{i \in Q_\varphi} \beta_i x_i \geq \alpha_0^R \qquad (3)_R$$

to c-dominate the inequality $(3)_M$, where $M \in \mathscr{C}$ and $M \subset R \subseteq N^+$. For any 0–1 vector x with support $Q(x) = T$, the difference between the values of the right hand side and the left hand side of $(3)_R$ is

$$\Delta(T)_R = \alpha_0^R - \sum_{i \in Q_R \setminus T} \alpha_i^R - \sum_{i \in Q_\varphi \cap T} \beta_i,$$

while the corresponding difference for $(3)_M$ is

$$\Delta(T)_M = \alpha_0^M - \sum_{i \in Q_M \setminus T} \alpha_i^M - \sum_{i \in Q_\varphi \cap T} \beta_i.$$

Clearly, the inequality $(3)_R$ ($(3)_M$) is violated by x if and only if $\Delta(T)_R > 0$ ($\Delta(T)_M > 0$).

By definition, the inequality $(3)_R$ c-dominates $(3)_M$ if and only if every 0–1 vector x that violates $(3)_M$ also violates $(3)_R$. Hence $(3)_R$ c-dominates $(3)_M$ if and only if

$$\Delta(T)_R > 0 \text{ for all } T \subseteq Q \text{ such that } \Delta(T)_M > 0. \qquad (6)$$

Condition (6) can be used to prove the following dominance property for inequalities of the type under discussion.

Theorem 2. *Let $M \in \mathscr{C}$ and $M \subset R \subseteq N^+$. The inequality $(3)_R$ c-dominates $(3)_M$ if an only if*

$$\sum_{j \in R \setminus M} (|Q_j \setminus T| - 1) a_j < \Delta(T)_M \qquad (7)$$

for all $T \subseteq Q_M \cup Q_\varphi$ (including $T = \emptyset$) such that $\Delta(T)_M > 0$.

Proof. We will show that the condition of the Theorem is equivalent to (6). From the definitions (see Theorem 1),

$$\alpha_0^R = \alpha_0^M + \sum_{j \in R \setminus M} a_j.$$

Further, for $i \in Q_M$,

$$\alpha_i^R = \min\left\{ \alpha_0^M + \sum_{j \in R \setminus M} a_j, \quad \alpha_i^M + \sum_{j \in R \setminus M \mid i \in Q_j} a_j \right\}$$

$$= \alpha_i^M + \sum_{j \in R \setminus M \mid i \in Q_j} a_j \quad (\text{since } \alpha_i^M \leq \alpha_0^M),$$

and for $i \in Q_R \setminus Q_M$,

$$\alpha_i^R = \min\left\{ \alpha_0^M + \sum_{j \in R \setminus M} a_j, \sum_{j \in R \setminus M \mid i \in Q_j} a_j \right\} = \sum_{j \in R \setminus M \mid i \in Q_j} a_j.$$

Thus for any $T \subseteq Q$,

$$\Delta(T)_R = \Delta(T)_M + \sum_{j \in R \setminus M} a_j - \sum_{i \in Q_R \setminus T} \left(\sum_{j \in R \setminus M \mid i \in Q_j} a_j \right)$$

$$= \Delta(T)_M + \sum_{j \in R \setminus M} a_j - \sum_{j \in R \setminus M} |Q_j \setminus T| a_j \tag{8}$$

and therefore condition (6) holds if and only if (7) holds for all $T \subseteq Q$ such that $\Delta(T)_M > 0$.

It remains to be shown that (7) holds for all $T \subseteq Q$ such that $\Delta(T)_M > 0$ if and only if it holds for all $T \subseteq Q_M \cup Q_\varphi$ such that $\Delta(T)_M > 0$. The 'only if' part is obvious. To show the 'if' part, let $T_0 \subseteq Q$, $T_0 \not\subseteq Q_M \cup Q_\varphi$, be such that $\Delta(T_0)_M > 0$, and let $T_1 = T_0 \cap (Q_M \cup Q_\varphi)$. Then $\Delta(T_0)_M = \Delta(T_1)_M$, and since $a_j > 0$ for all $j \in R$ and $T_1 \subset T_0$, if (7) is violated for $T = T_0$, it is also violated for $T = T_1$. □

Given a cover M, Theorem 2 can in principle be used to find an extension R of M (if one exists), such that the inequality $(3)_R$ c-dominates $(3)_M$. However, the condition of Theorem 2 is in general not easy to check. We will show below that for the family of covers M that are minimal, the condition of Theorem 2 reduces to a simpler one, which can be checked in linear time. Before discussing that case, however, we wish to note that, unlike in the case of the sequential lifting procedures for linear inequalities in 0–1 variables (see, for instance, [1]), if one wishes to strengthen an inequality $(3)_M$ by expanding the cover M into a larger cover R with the required properties, this cannot always be done sequentially, i.e., by introducing the elements of $R \setminus M$ one at a time. This is illustrated by the following example.

Example 2. In the inequality

$$8x_1x_2 + 5x_1x_5 + 5x_1x_6 + 4x_2x_3x_4 + x_3x_5 + x_4x_6 \leq 12,$$

let the sets Q_j, $j=1,\ldots,6$, be indexed from left to right. Taking $M=\{1,2,3,4\}$, we obtain the inequality

$$10\bar{x}_1+10\bar{x}_2+4\bar{x}_3+4\bar{x}_4+5\bar{x}_5+5\bar{x}_6\geqslant 10.$$

If we attempt to expand M by setting $R=M\cup\{5\}$, condition (7) is not met for $T=\{3,5\}$. Similarly, if we set $R=M\cup\{6\}$, (7) is violated for $T=\{4,6\}$. However, if we set $R=M\cup\{5,6\}$, then (7) is satisfied for all T such that $\Delta(T)_M>0$. □

For any $M\in\mathscr{C}$, let $C(M)=M\cup N^-$. As mentioned earlier, $C(M)$ is a cover for (2) if and only if $M\in\mathscr{C}$. If $C(M)$ is a minimal cover for (2), M is a minimal cover for (2^+) (but the converse is not necessarily true).

We now focus on the case when $C(M)$ is a minimal cover for (2). Recall from Section 1 that in this case $\alpha_i^M=\beta_j^\varphi=\alpha_0^M$ for all $i\in Q_M$ and $j\in Q_\varphi$ in $(3)_{M,\varphi}$. Further, if $Q_M\cap Q_\varphi\neq\emptyset$, then $(3)_{M,\varphi}$ is vacuous.

Corollary 2.1. *Let $M\subset R\subseteq N^+$, $Q_R\cap Q_\varphi=\emptyset$, and let $C(M)$ be a minimal cover for (2). Then the inequality*

$$\sum_{i\in Q_R}\alpha_i^R\bar{x}_i+\sum_{i\in Q_\varphi}\beta_i x_i\geqslant\alpha_0^R \qquad (3)_R$$

c-dominates $(3)_M$ if and only if

$$\sum_{i\in Q_R\setminus Q_M}\alpha_i^R<\alpha_0^R. \qquad (9)$$

Further, $(3)_R$ strictly c-dominates $(3)_M$ if and only if (9) holds and either $\alpha_k^R<\alpha_0^R$ for some $k\in Q_M$, or $\beta_l<\alpha_0^R$ for some $l\in Q_\varphi$.

Proof. If $C(M)$ is a minimal cover for (2), then since $\alpha_i^M=\beta_j=\alpha_0^M$, $\forall i\in Q_M$, $j\in Q_\varphi$, from the definition of $\Delta(T)_M$ it follows that $\Delta(T)_M>0$ implies $(Q_M\setminus T)\cup(Q_\varphi\cap T)=\emptyset$ and $\Delta(T)_M=\alpha_0^M$. Further, for any $T\subseteq Q_M\cup Q_\varphi$, $\Delta(T)_M>0$ implies $T=Q_M$. Therefore in this case

$$\sum_{j\in R\setminus M}|Q_j\setminus T|a_j=\sum_{i\in Q_R\setminus Q_M}\left(\sum_{j\in R\setminus M\,|\,i\in Q_j}a_j\right),$$

and thus condition (7) of Theorem (2) becomes

$$\sum_{i\in Q_R\setminus Q_M}\left(\sum_{j\in R\setminus M\,|\,i\in Q_j}a_j\right)<\alpha_0^M+\sum_{j\in R\setminus M}a_j$$

which is the same as (9). Thus (9) is necessary and sufficient for $(3)_R$ to c-dominate $(3)_M$.

Assume now that $(3)_R$ strictly c-dominates $(3)_M$. Then there exists a 0–1 vector x with support $Q(x)=T$ that satisfies $(3)_M$ but not $(3)_R$. Thus there exists either $k\in Q_M\setminus T$ such that $\alpha_k^R<\alpha_0^R$, or $l\in Q_\varphi\cap T$ such that $\beta_l<\alpha_0^R$.

Conversely, if there exists $k \in Q_M$ such that $\alpha_k^R < \alpha_0^R$, then $x^0 \in \{0,1\}^q$ defined by $Q(x^0) = Q \backslash \{k\}$ satisfies $(3)_M$ but not $(3)_R$; and if there exists $l \in Q_\varphi$ such that $\beta_l < \alpha_0^R$, then $x^* \in \{0,1\}^q$ defined by $Q(x^*) = \{l\}$ satisfies $(3)_M$ but not $(3)_R$. □

An important practical consequence of Corollary 2.1, which is used in the Algorithm of the next section, can be stated as follows. For $M \subseteq N^+$, we define

$$E_i(M) = \{j \in N^+ \mid |Q_j \backslash Q_M| = i\}, \quad i = 0, 1, \ldots, p,$$

where $p = \max_{j \in N^-} |Q_j \backslash Q_M|$, and denote $E(M) = E_0(M) \cup E_1(M)$. (This set $E(M)$ is the same as the one used in [6].)

Corollary 2.2. *Let M and R be as in Corollary 2.1. If $R \subseteq E(M)$, then $(3)_R$ c-dominates $(3)_M$.*

Proof. Let $R \subseteq E(M)$ and denote $R_i = R \cap E_i(M)$, $i = 0, 1$. Then $R = R_0 \cup R_1$, and

$$|(Q_R \backslash Q_M) \cap Q_j| = \begin{cases} 0 & \text{for } j \in R_0, \\ 1 & \text{for } j \in R_1. \end{cases}$$

Hence

$$\sum_{i \in Q_R \backslash Q_M} \alpha_i^R = \sum_{j \in R_1} a_j < \alpha_0^M + \sum_{j \in R} a_j = \alpha_0^R$$

i.e., condition (9) of Corollary 2.1 is satisfied. □

Thus any minimal cover $M \subseteq N^+$ for (2^+) can safely be extended to include all terms in $E(M)$, without weakening the inequality $(3)_M$. The computational effort required to identify $E(M)$ and extend M to $E(M)$ is linear in q. However, we can often go beyond $E(M)$, as will be clear when we restate Corollary 2.1 in slightly different form.

Corollary 2.3. *Let M and R be as in Corollary 2.1, and let $R_i = R \cap E_i(M)$, $i = 1, \ldots, p$. Then the inequality $(3)_R$ c-dominates $(3)_M$ if and only if*

$$\sum_{i=2}^{p} \left[(i-1) \sum_{j \in R_i} a_j \right] < \sum_{j \in R_0} a_j - b. \tag{10}$$

Further, $(3)_R$ strictly c-dominates $(3)_M$ if and only if (10) holds and either

$$\sum_{j \in R \mid k \notin Q_j} a_j > b \tag{11}$$

for some $k \in Q_M$, or

$$\sum_{j \in R} a_j + \sum_{j \in N^- \mid l = \varphi(j)} a_j > b \tag{12}$$

for some $l \in Q_\varphi$.

Proof. We show that conditions (10), (11) and (12) are equivalent to the conditions of Corollary 2.1.

For $i \in Q_R \backslash Q_M$,

$$\alpha_i^R = \sum_{j \in R \backslash M \mid i \in Q_j} a_j = \sum_{j \in R \mid i \in Q_j} a_j \qquad \text{(since for } j \in M, i \notin Q_j),$$

while

$$\alpha_0^R = \sum_{j \in R} a_j - b.$$

Thus condition (9) of Corollary 2.1 amounts to

$$\left(\sum_{i \in Q_R \backslash Q_M} \alpha_i^R = \right) \sum_{i \in Q_R \backslash Q_M} \left(\sum_{j \in R \mid i \in Q_j} a_j \right)$$

$$= \sum_{j \in R} |(Q_R \backslash Q_M) \cap Q_j| \, a_j < \sum_{j \in R} a_j - b \quad (= \alpha_0^R). \tag{13}$$

Now $R = \bigcup_{i=0}^{p} R_i$, and for $j \in R_i$, $|(Q_R \backslash Q_M) \cap Q_j| = i$, $i = 0, 1, \ldots, p$. Hence (13) can be written as

$$\sum_{j \in R_1} a_j + \sum_{i=2}^{p} \left(i \sum_{j \in R_i} a_j \right) < \sum_{j \in R_0} a_j + \sum_{j \in R_1} a_j + \sum_{i=2}^{p} \left(\sum_{j \in R_i} a_j \right) - b$$

or, equivalently,

$$\sum_{j \in R_0} a_j - b > \sum_{i=2}^{p} \left(i \sum_{j \in R_i} a_j \right) - \sum_{i=2}^{p} \left(\sum_{j \in R_i} a_j \right) = \sum_{i=2}^{p} \left[(i-1) \sum_{j \in R_i} a_j \right],$$

which is precisely (10). Thus (10) is equivalent to (13), hence to (9), and this proves the first statement.

On the other hand, the condition $\alpha_k^R < \alpha_0^R$ (for some $k \in Q_M$) of Corollary 2.1 can be restated as

$$\sum_{j \in R \mid k \in Q_j} a_j < \sum_{j \in R} a_j - b$$

which is equivalent to (11). Also, the condition $\beta_l < \alpha_0^R$ (for some $l \in Q_\varphi$) can be written as

$$\sum_{j \in N^- \mid l = \varphi(j)} |a_j| < \sum_{j \in R} a_j - b,$$

which is the same as (12). This proves the second statement. □

Condition (10) of Corollary 2.3 gives the precise extent to which a minimal cover M for (2^+) (such that $Q_M \cap Q_\varphi = \emptyset$) can be extended *beyond* the sets $E_0(M)$ and $E_1(M)$, into sets $E_i(M)$ for $i \geq 2$. This is extensively used in the Algorithm described in Section 3.

In particular, replacing the inequality (2) by the set of inequalities

$$f^+(x)+\sum_{j\in N^-}|a_j|\bar{x}_{\varphi(j)}\leqslant \hat{b}\left(=b-\sum_{j\in N^-}a_j\right),\quad \varphi\in\Phi^-, \tag{5}_\varphi$$

we have

Corollary 2.4. *Let M be a minimal cover for $(5)_\varphi$ for some $\varphi\in\Phi^-$, let $M\subset R\subseteq N$, and $R_i=R\cap E_i(M)$, $i=0,1,\ldots,p$. Assume R satisfies $Q_{R\cap N^+}\cap Q_\varphi=\emptyset$. Then the inequality*

$$\sum_{i\in Q_{R\cap N^+}}\alpha_i^R\bar{x}_i+\sum_{i\in Q_{R\cap N^-}}\alpha_i^R x_i\geqslant\alpha_0^R \tag{14}_R$$

c-dominates the generalized covering inequality

$$\sum_{i\in Q_{M\cap N^+}}\bar{x}_i+\sum_{i\in Q_{M\cap N^-}}x_i\geqslant 1 \tag{14}_M$$

if and only if

$$\sum_{i=2}^{p}\left[(i-1)\sum_{j\in R_i}a_j\right]<\sum_{j\in R_0}a_j-\hat{b}; \tag{15}$$

and $(14)_R$ strictly c-dominates $(14)_M$ if and only if (15) holds and there exists $k\in Q_M$ such that

$$\sum_{j\in R\mid k\notin Q_j}a_j>\hat{b}. \tag{16}$$

Proof. Specialize Corollary 2.3 to inequality $(5)_\varphi$. □

Again, the computational effort involved in checking whether conditions (15), (16) are satisfied for some $R\subseteq N\backslash M$ is linear in q.

The following example illustrates the usefulness of these results for obtaining a more compact linear system equivalent to a nonlinear inequality (2) than the set of generalized covering inequalities.

Example 3. Consider the multilinear inequality

$$6x_1x_2x_3x_4+4x_1x_5-3x_3x_4x_6+2x_1x_2x_4x_7+2x_3x_4x_8\leqslant 7. \tag{17}$$

This inequality is equivalent to the system defined by the six generalized covering inequalities

$$\begin{aligned}
&\bar{x}_1+\bar{x}_2+\bar{x}_3+\bar{x}_4+\bar{x}_5+x_6 &&\geqslant 1,\\
&\bar{x}_1+\bar{x}_2+\bar{x}_3+\bar{x}_4+\bar{x}_5 \quad +\bar{x}_7 &&\geqslant 1,\\
&\bar{x}_1+\bar{x}_2+\bar{x}_3+\bar{x}_4+\bar{x}_5 \quad\quad +\bar{x}_8 &&\geqslant 1,
\end{aligned}$$

$$\bar{x}_1+\bar{x}_2+\bar{x}_3+\bar{x}_4 \quad +x_6+\bar{x}_7 \quad \geq 1,$$

$$\bar{x}_1+\bar{x}_2+\bar{x}_3+\bar{x}_4 \quad +x_6 \quad +\bar{x}_8 \geq 1,$$

$$\bar{x}_1+\bar{x}_2+\bar{x}_3+\bar{x}_4+\bar{x}_5+x_6+\bar{x}_7+\bar{x}_8 \geq 1,$$

derived from the minimal covers $\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 3, 5\}$, and $\{2, 3, 4, 5\}$, respectively, of the implied inequality

$$6x_1x_2x_3x_4+4x_1x_5+3\bar{x}_6+2x_1x_2x_4x_7+2x_3x_4x_8 \leq 10. \qquad (17)_\varphi$$

The last generalized covering inequality is the only redundant one.

Letting $M=\{1, 2, 3\}$, it follows that $Q_M=\{1, 2, 3, 4, 5, 6\}$. Thus $E_0(M)=\{1, 2, 3\}$ and $E_1(M)=\{4, 5\}$. Therefore, letting $R=M\cup E_0(M)\cup E_1(M)=\{1, \ldots, 5\}$, from Corollary 2.2, the inequality

$$7\bar{x}_1+7\bar{x}_2+7\bar{x}_3+7\bar{x}_4+4\bar{x}_5+3x_6+2\bar{x}_7+2\bar{x}_8 \geq 7 \qquad (3)_{R,\varphi}$$

c-dominates inequality $(3)_{M,\varphi}$, which is the first of the above six. Since the condition $\alpha_k^M < \alpha_0^M$ of Corollary 2.1 is satisfied for $k=6$, $(3)_{R,\varphi}$ strictly c-dominates $(3)_{M,\varphi}$. In fact, $(3)_{R,\varphi}$ (strictly) c-dominates all of the generalized covering inequalities and is thus equivalent to (17). □

We conclude this section by defining an alternative linearization of (2). To be specific, if we consider (2) to be a linear inequality in the 0–1 variables

$$y_j=\prod_{i\in Q_j} x_i, \quad j\in N$$

and denote it by $(2)_y$, then w.l.o.g. we may assume that $a_j>0$ for all $j\in N$, and that $a_1\geq a_2\geq\cdots\geq a_n$. Then applying the results of [1, 2], we can replace $(2)_y$ by the equivalent set of canonical inequalities

$$\sum_{j\in\mathscr{E}(S)} y_j \leq |S|-1, \quad S\in\mathscr{H}. \qquad (18)_S$$

Here $\mathscr{E}(S)$ is the *extension* of S, defined as

$$\mathscr{E}(S)=S\cup\{j\in N\backslash S \mid j<j_1\},$$

with $j_1=\min_S j$; while $\mathscr{H}$ is the family of *strong* covers for $(2)_y$, where a minimal cover S is called strong if there exists no minimal cover $T\neq S$ such that $|T|=|S|$ and $\mathscr{E}(S)\subseteq\mathscr{E}(T)$.

For any given $S\in\mathscr{H}$, rewriting $(18)_S$ in terms of x, we can linearize it using the above results. If we do this for every $S\in\mathscr{H}$, we obtain a new linearization of (2), different from the one discussed earlier. Naturally, the question arises as to how this new linearization compares with the one discussed above. Both approaches were implemented and tested, and the computational results are reported in Section 4.

3. An algorithm for solving multilinear 0–1 programs

We now describe several variants of an algorithm for the problem (MLP) stated at the beginning of this paper. If the objective function f_0 has rational coefficients, then it can be linearized by introducing a new (integer) variable z (or its binary expansion), and amending the constraint set by one new inequality involving z and the nonlinear part of f_0. Thus, w.l.o.g. the multilinear 0–1 program can be stated in the form

$$\begin{aligned} &\text{Max} \sum_{i \in Q} c_i x_i, \\ &\sum_{j \in N_k} a_{kj} \left(\prod_{i \in Q_{kj}} x_i \right) \leqslant b_k, \quad k \in K, \\ &x_i = 0 \text{ or } 1, \quad i \in Q, \end{aligned} \tag{MLP}$$

where the set Q is now defined as

$$Q = \bigcup_{\substack{k \in K \\ j \in N_k}} Q_{kj}.$$

The algorithm that we present below, like the one by Granot and Granot [9] (see also [10], and [11]), generates some linear inequalities implied by the constraint set of (MLP), and solves the resulting linear 0–1 program, which is a relaxation of (MLP). At iteration t, let this linear 0–1 program be denoted (P_t). If an optimal solution to (P_t) is feasible for (MLP), then it is optimal for (MLP) and we stop. Otherwise we generate a new set of linear inequalities implied by the constraints of (MLP), such that the new inequalities cut off the solution to (P_t), and solve the linear 0–1 program (P_{t+1}) obtained from (P_t) by adding the new inequalities. Since at every iteration the solution to the current problem (P_t) is cut off, the algorithm is obviously finite.

Our procedure differs from that of [9, 10, 11] mainly in that we use a more compact linearization, based on the theory of Section 2. To be more specific, we start with a set covering inequality associated with a minimal cover, but then use Theorem 2 and its corollaries to extend the cover so as to obtain as strong an inequality as the conditions of the corollaries permit.

The reason for starting with inequalities associated with minimal covers, is that for this class we can check in linear time whether the inequality is dominated by another one and if so, generate a dominating inequality. Experience shows that the proportion of minimal covers that can be extended is high (90% is a typical case) and tends to increase with the number of terms per constraint. Since the use of extended covers tends to produce smaller cardinality linear equivalents of each nonlinear inequality, it can also be expected to reduce the number of iterations needed to solve (MLP). This is indeed the case, except for problems with few nonlinear terms per constraint, as shown by the computational experience discussed in the next section.

While the procedure outlined above is finite, it may take many iterations. We found it therefore preferable not to solve (P_t) exactly at every iteration, but use a heuristic to find an approximate solution. We proceed this way until, at some iteration t, an approximate solution to (P_t) is found to be feasible to (MLP). At that point we replace the heuristic by an exact algorithm. The particular heuristic that we use on the sequence of linear 0–1 programs (P_t) is the Pivot and Complement procedure of Balas and Martin [3]. When we switch to an exact algorithm, we use a branch and bound/implicit enumeration procedure implemented by Clarence H. Martin.

Another deviation from the above outline is that we found it convenient to periodically remove some of the linear inequalities generated earlier. This is done according to a particular procedure so as to insure that convergence is maintained.

Finally, to facilitate the search for minimal covers and their extensions, used in the linearization procedure, we start the algorithm by ordering once and for all the terms of each constraint according to decreasing absolute values of their coefficients.

As a starting solution we use the optimal solution to the unconstrained problem, i.e., x^0 defined by $x_i^0 = 1$ if $c_i > 0$ and $x_i^0 = 0$ otherwise.

A flowchart of the algorithm is shown in Figure 1.

The heart of our procedure is of course the generation of linear inequalities. The rules to be described below are essentially based on Corollary 2.4.

First, it should be stated that at every iteration we generate one linear inequality from every inequality of (MLP) violated by the current solution x^0, except for the first iteration, when we generate one linear inequality (using the cover $M = N$) from every constraint of (MLP), whether violated or not (the exception was adopted as a result of computational experimentation).

To describe the procedure, let

$$\sum_{j \in N} a_j \left(\prod_{i \in Q_j} x_i \right) \leq b \tag{2}$$

be one of the inequalities violated by x^0, and let $|a_1| \geq |a_2| > \cdots \geq |a_n|$.

Denote

$$P^+(x^0) = \left\{ j \in N^+ \,\middle|\, \prod_{i \in Q_j} x_i^0 = 1 \right\}, \qquad P^-(x^0) = \left\{ j \in N^- \,\middle|\, \prod_{i \in Q_j} x_i^0 = 0 \right\},$$

with $P(x^0) = P^+(x^0) \cup P^-(x^0)$. Recall that inequality (2) gives rise to the family of (all positive) inequalities

$$\sum_{j \in N^+} a_j \left(\prod_{i \in Q_j} x_i \right) + \sum_{j \in N^-} |a_j| \bar{x}_{\varphi(j)} \leq \hat{b} \left(= b - \sum_{j \in N^-} a_j \right), \tag{5}_\varphi$$

where $\varphi \in \Phi^-$. Define

$$\Phi^-(x^0) = \{ \varphi \in \Phi^- \mid x^0 \text{ violates } (5)_\varphi \}.$$

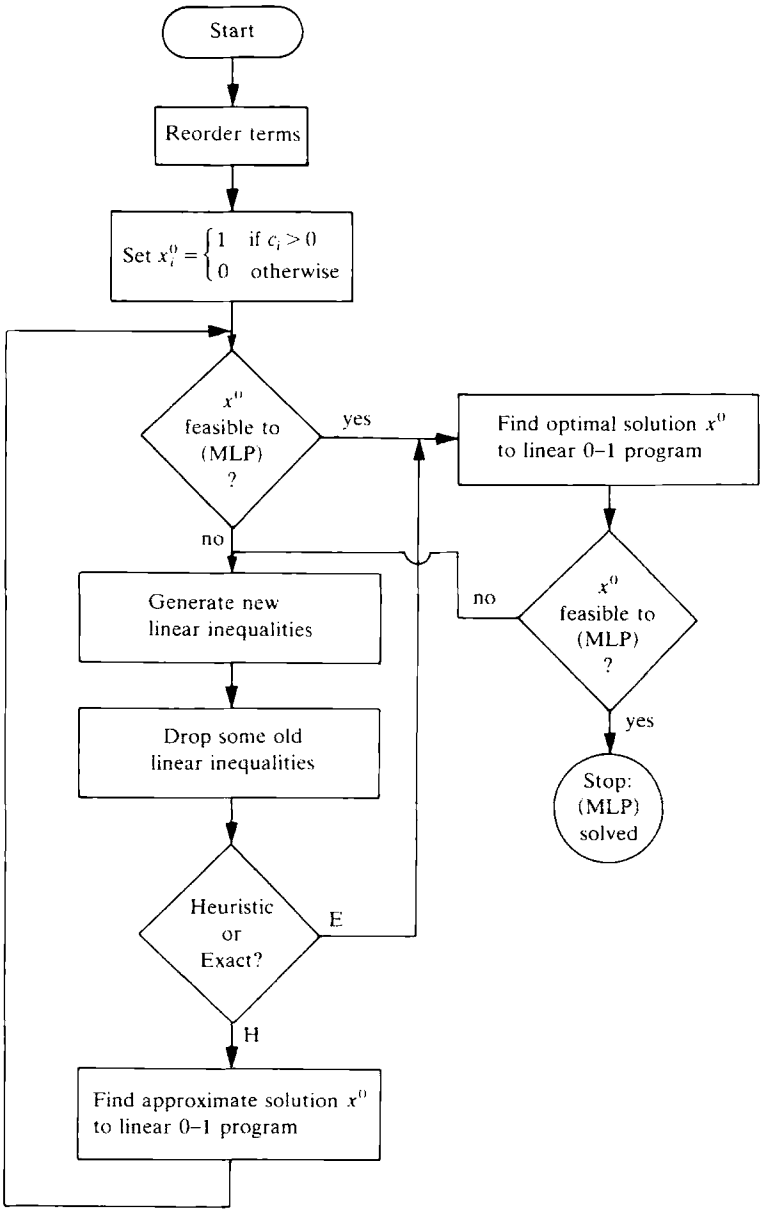

Fig. 1. Flowchart of the algorithm.

It is easy to show that $\Phi^-(x^0) \neq 0$; in fact, the algorithm outlined below generates a mapping $\varphi \in \Phi^-(x^0)$. Thus, given x^0 and the family of nonlinear inequalities $(5)_\varphi$, $\varphi \in \Phi^-(x^0)$ (corresponding to a particular inequality (2) violated by x^0), our cut generating algorithm consists of the following sequence of steps:

1. Finding a minimal cover $M \subseteq P(x^0)$ and a set of $\varphi(j)$, $j \in M \cap N^-$, such that x^0 violates the corresponding generalized covering inequality.

2. Extending M to a maximal set R satisfying condition (15), and such that $(R\backslash M)\subseteq N^+$.

3. Choosing $\varphi(j)$ for $j\in N^-\backslash R$ in a way that avoids as much as possible producing nonzero coefficients for complementary pairs of variables, and using it to determine the remaining coefficients of the inequality.

A discussion of each step follows.

1. Let $P(x^0)=\{i_1,\ldots,i_t\}$ be ordered by the same rule as N, i.e., $i_k<i_{k+1}$, $k=1,\ldots,t-1$. Let $j\in\{1,\ldots,t\}$ be the largest integer such that $\{i_j, i_{j+1},\ldots,i_t\}$ is a cover, and let $l\in\{j, j+1,\ldots,t\}$ be the smallest integer such that $M=\{i_j, i_{j+1},\ldots,i_l\}$ is a cover for (2). Then, obviously, M is minimal.

Next choose $\varphi(j)$ for $j\in M\cap N^-$ to be the first $i\in Q_j$ such that $x_i^0=0$, a choice consistent with the requirement that $\varphi\in\Phi^-(x^0)$. For any $\varphi\in\Phi^-(x^0)$ chosen in this way, M satisfies the requirement of Corollary 2.4, i.e., $Q_{M\cap N^-}\cap Q_\varphi=\emptyset$, and the generalized covering inequality

$$\sum_{i\in Q_{M\cap N^-}}\bar{x}_i+\sum_{i\in Q_{M\cap N^-}}x_i\geq 1$$

corresponding to M is violated by x^0.

2. Construct the extension R of M into N^+ as follows. As in Section 2, define

$$E_i(M)^+=\{j\in N^+\,|\,|Q_j\backslash Q_M|=i\},\quad i=0,1,\ldots,p,$$

and set $E(M)^+=E_0(M)^+\cup E_1(M)^+$. First add to R the set $E(M)^+$. Next for $i=2,\ldots,p$, consider the elements of $E_i(M)^+$ in order of increasing a_j, and include into R as many as can be included without violating condition (15). If all $j\in E_i(M)^+$ can be added to R, set $i\leftarrow i+1$ and repeat. Otherwise stop with the last element of $E_i(M)^+$ whose inclusion into R does not lead to a violation of (15).

3. To define $\varphi(j)$ for the remaining indices, i.e., for $j\in N^-\backslash R$, we proceed as follows. Let R be the extended set resulting at the end of step 2 and let α_i^R, $i\in Q\cup\{0\}$, be the corresponding coefficient values, where R and the α_i^R are updated by combining variables and their complements whenever such pairs occur. Since it is possible for either $\bar{x}_i$ or x_i (but never for both) to appear in the resulting inequality $(3)_{R,\varphi}$, we partition Q into Q_R^+, Q_R^- and Q^0, where $Q_R^+=\{i\in Q_R\,|\,\bar{x}_i$ appears in $(3)_{R,\varphi}\}$, $Q_R^-=\{i\in Q_R\,|\,x_i$ appears in $(3)_{R,\varphi}\}$, and $Q^0=Q\backslash(Q_R^+\cup Q_R^-)$. We then choose $\varphi(j)$ according to the following rule:

If $Q_j\backslash Q_R\neq\emptyset$, let $\varphi(j)$ be the first index in $Q_j\backslash Q_R$.

If $Q_j\backslash Q_R=\emptyset$, but $Q_j\cap Q_R^-\neq\emptyset$, let $\varphi(j)=h$, where $\alpha_h^R=\max\{\alpha_i^R\,|\,i\in Q_j\cap Q_R^-\}$.

Otherwise, let $\varphi(j)=k$, where $\alpha_k^R=\min_{i\in Q_j}\alpha_i^R$.

Once $\varphi(j)$ is selected, set $R\leftarrow R\cup\{\varphi(j)\}$, update $\alpha_{\varphi(j)}^R$ and α_0^R, as well as Q_R^+, Q_R^- and Q^0 (combining variables, if necessary), and proceed to the next $j\in N^-\backslash M$. This choice is again consistent with the requirement that $\varphi\in\Phi^-(x^0)$.

Having generated the linear inequality, we eliminate the complemented variables, i.e., restate the inequality in the original variables, and add it to the current linear 0–1 program.

Next we illustrate the procedure on an example.

Example 4. Consider the multilinear inequality

$$16x_2x_4x_5 - 10x_2x_6 + 10x_1x_2x_3 + 5x_1x_5 - 4x_5x_7 + 4x_3x_5 \leq 1, \tag{19}$$

which is violated by $x^0 = (1, 1, 1, 0, 1, 1, 0)$. We have $Q_1 = \{2, 4, 5\}$, $Q_2 = \{2, 6\}$, $Q_3 = \{1, 2, 3\}$, $Q_4 = \{1, 5\}$, $Q_5 = \{5, 7\}$, $Q_6 = \{3, 5\}$. Further, $N^+ = \{1, 3, 4, 6\}$, $N^- = \{2, 5\}$, and $P^+(x^0) = \{3, 4, 6\}$, $P^-(x^0) = \{5\}$, $P(x^0) = \{3, 4, 5, 6\}$.

The corresponding inequality with positive coefficients is (in general form)

$$16x_2x_4x_5 + 10\bar{x}_{\varphi(2)} + 10x_1x_2x_3 + 5x_1x_5 + 4\bar{x}_{\varphi(5)} + 4x_3x_5 \leq 15, \tag{20}_\varphi$$

where $\varphi(2)$ and $\varphi(5)$ depend on the choice of $\varphi \in \Phi^-(x^0)$.

1. We identify the minimal cover $M = \{3, 4, 5\}$ for (19), which is also a minimal cover for $(20)_\varphi$. We then choose $\varphi(5) = 7$ and thus obtain

$$Q_{M \cup N^+} = \{1, 2, 3, 5\}, \qquad Q_{M \cup N^-} = \{7\},$$

Applying Theorem 1 to $(20)_\varphi$, we derive from the minimal cover M the generalized covering inequality

$$\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_5 + x_7 \geq 1 \tag{21}$$

violated by x^0.

2. We identify the sets $E_0(M)^+ = \{3, 4, 6\}$, $E_1(M)^+ = \{1\}$, and since $E_i(M)^+ = \emptyset$ for $i \geq 2$, we have $R = M \cup E(M)^+ = \{1, 3, 4, 5, 6\}$, $Q_R = \{1, 2, 3, 4, 5, 7\}$.

3. For $j = 2$ ($j \in N^- \backslash R$), we set $\varphi(2) = 6$, since $Q_2 \backslash Q_R = \{6\}$, and update R by including $\{2\}$. Thus $R = \{1, 2, 3, 4, 5, 6\}$, and by applying Theorem 1 to $(20)_\varphi$ (with $\varphi(2) = 6$ and $\varphi(5) = 7$), we obtain from the extended cover R the inequality

$$15\bar{x}_1 + 26\bar{x}_2 + 14\bar{x}_3 + 16\bar{x}_4 + 25\bar{x}_5 + 10x_6 + 4x_7 > 34$$

which is also violated by x^0, and which strictly c-dominates the generalized covering inequality (21). □

As mentioned earlier, we found it necessary to periodically remove inequalities from the linear 0–1 program in order to keep its size within manageable limits. The cut dropping procedure operates as follows. The set V of all inequalities generated during the procedure is partitioned into three subsets. V_1 contains exactly one inequality generated at each iteration, namely the one derived from the most violated constraint of (MLP). Cuts in V_1 are never removed, as a guarantee that every solution to the linear 0–1 program generated during the procedure is cut off by at least one inequality. V_2 consists of all inequalities associated with extended covers and not contained in V_1, whereas V_3 consists of the remaining inequalities (i.e., those associated with minimal covers that could not be extended).

Whenever the number of inequalities in the linear 0–1 program attains a predetermined threshold value Δ, all inequalities in V_3 not binding at the current solution are dropped. The subset V_3 is our first preference for dropping, since it usually consists of the weakest inequalities of the current system. If removing the nonbinding

inequalities in V_3 is not sufficient for accommodating all the inequalities generated at the current iteration, then the nonbinding inequalities in V_2 are also dropped. Finally, if removing all the nonbinding inequalities of V_3 and V_2 is still insufficient, we drop an appropriate number of binding inequalities in V_3 and, if necessary, in V_2.

This completes the description of the main version of our algorithm, henceforth called Algorithm I. Two additional versions of the algorithm were implemented, which will now be briefly described.

Algorithm II differs from Algorithm I in that it generates linear inequalities not directly from an inequality (2) of (MLP), but from an extended canonical inequality implied by $(2)_y$, as described at the end of Section 2. The choice of the inequality (2), respectively $(2)_y$, as well as that of the minimal cover M, is the same as in Algorithm I. Another minimal cover C is then identified, such that $|C| = |M|$ and $\mathcal{E}(M) \subseteq \mathcal{E}(C)$ (preferably, but not necessarily, $C \neq M$). The cut generating procedure described above is then applied to the canonical inequality defined by $\mathcal{E}(M)$ and expressed in terms of x, for which M is still a minimal cover. Everything else is as in Algorithm I.

Finally, Algorithm III differs from the other two versions by the fact that it derives only generalized covering inequalities corresponding to minimal covers without attempting to strengthen them by extending the covers. For this version, the choice of the minimal cover is done differently, namely by setting $M = \{i_1, \ldots, i_k\}$, where k is the smallest integer such that M is a cover. As a result, M (which is of course minimal) is of smaller cardinality than the cover selected in Algorithm I which in the absence of the extension procedure is preferable. The superiority of this choice of minimal cover for this particular algorithm was unequivocally supported in the computational testing. The other ingredients of Algorithm III are the same as those of I and II. Algorithm III may be viewed as our version of the algorithm of Granot and Granot [9]; the differences from the latter (improvements in our view) having been adopted in order to make it comparable with Algorithms I and II.

Algorithm I, which for all but very sparse problems is the most efficient of the three procedures implemented, was also run in the heuristic mode, i.e. by removing all steps subsequent to the finding of a feasible solution to (MLP). The purpose of this exercise was to obtain information on the quality of the solutions obtainable by such an approach.

4. Computational results

The algorithms discussed above were coded in FORTRAN and tested on a series of randomly generated test problems, using an IBM 3081 Model K computer and a FORTRAN H level compiler.

The first set of test problems consists of 30 multilinear 0–1 programs, 5 in each of 6 classes that differ among themselves in the number of terms per constraint. The number of constraints and variables (denoted by m and n respectively) is the

same in all of these problems ($m = 10, n = 30$), and the number of terms per constraint is randomly drawn from a uniform distribution on the interval $[3, T]$, where T is shown in Table 1. The constraint coefficients a_{kj} are integers uniformly distributed on $[-5, 15]$, while the b_k are integers drawn from a uniform distribution on $(0.3s_k, 0.8s_k)$, where $s_k = \sum_j a_{kj}$. The cost functions are linear, with integer coefficients uniformly distributed on $[1, 20]$. Finally, the number of variables per term is uniformly distributed on $[2, 6]$. The results are shown in Table 1. Our remaining 55 test problems were generated in the same manner as the first set, with the values of m, n and T as indicated in the tables. The test problems are available upon request. Algorithm I was also tested on a set of problems from the literature.

Table 1
Number of problems solved and average CPU time (seconds)[a]

m	n	T	Algorithm I		Algorithm II		Algorithm III	
			No. solved	Time	No. solved	Time	No. solved	Time
10	30	10	5	2.6	5	1.3	5	0.6
10	30	20	5	0.2	5	0.2	5	0.2
10	30	30	5	2.8	5	10.8	5	14.6
10	30	40	5	2.6	3	24.6	3	24.8
10	30	50	5	8.5	3	33.2	2	29.9
10	30	60	5	4.5	4	25.2	2	26.9

[a] 5 problems per class.
Limit set to 1 minute CPU time or 150 iterations per problem.
Time averaged for all 5 problems. Time for problems not solved within 1 minute taken to be 1 minute.

All test problems were run under two kinds of limitations (as indicated in the tables): a time limit (1 or 5 minutes, depending on the phenomenon studied) and a limit (150 or 200) on the number of iterations, hence on the number of nonremovable inequalities generated, due to space limitations. The latter limit is different from the threshold value Δ that triggers the cut dropping routine. In Algorithms I and II, after some experimentation Δ was set to $2n$, i.e., twice the number of variables; whereas in Algorithm III computational tests indicated a higher value, and Δ was set equal to the maximum number of iterations (150 or 200).

All CPU times reported are exclusive of input/output time. The maximum input time for any of the test problems was 0.02 seconds.

Table 1 shows that although Algorithm III performs somewhat better than Algorithm I on the problems with $T = 10$ and $T = 20$ (i.e., with 6 and 12 terms per constraint on the average, respectively), its performance quickly deteriorates for higher values of T, as reflected in the sharply decreasing number of problems solved within the limits allowed. This concurs with the observation of Granot, Granot, and Vaessen [11] that for their algorithm (which performs phenomenally well on sparse problems), CPU time appears to grow exponentially with T. At the same time, the

performance of Algorithm I is only moderately affected by the increase of T. As for Algorithm II, its performance is not better than that of III on the problems with small T, and considerably worse than that of Algorithm I on the problems with large T. Thus the performance of Algorithm II will not be further pursued.

Table 2 compares the performance of Algorithms I and III on the same set of problems with the time and iteration limits for Algorithm III increased to 5 minutes and 200 iterations, respectively.

Table 2

Number of problems solved and average CPU time (seconds)[a]

m	n	T	Algorithm I[b]		Algorithm III[c]	
			No. solved	Time	No. solved	Time
10	30	10	5	2.6	5	0.6
10	30	20	5	0.2	5	0.2
10	30	30	5	2.8	5	14.6
10	30	40	5	2.6	4	111.2
10	30	50	5	8.5	3	91.5
10	30	60	5	4.5	4	76.9

[a] 5 problems per class. Time averaged for all 5 problems.
[b] Limit set to 1 minute or 150 iterations per problem.
[c] Limit set to 5 minutes or 200 iterations per problem. Time for problems not solved within 5 minutes taken to be 5 minutes.

The results show an even sharper contrast between the sensitivity of the two algorithms to an increase in the number of terms per constraints. We conclude that the more compact linearization based on the theory of Section 2 definitely pays off for problems with more than 12–15 terms per constraint.

In Table 3 we compare the average number of iterations and cuts (linear inequalities) generated, in order to better understand the difference in the performance of the two algorithms. We see that as T is increased from, say, 30 to 60, the number of iterations and cuts increases by more than 400% for Algorithm III, as opposed to 8–17% for Algorithm I. On the other hand, while the percentage of covers that can be extended (in Algorithm I) increases with T, the increase is only modest, since this percentage is high to begin with (i.e., for all problem classes). This modest increase cannot fully account for the sharply increasing difference in the number of iterations required by the two Algorithms. What the table does not show, however, is that as the number of terms per constraint increases, not only does the percentage of covers that can be extended increase, but more importantly, there is a significant increase in the *extent* to which every minimal cover can be extended: with more terms per constraint, many more indices are included in the extension of each cover.

Table 3
Number of iterations and of cuts[a]

m	n	T	Algorithm I[b]			Algorithm III[c]	
			Iterations	Cuts	Percent covers extended	Iterations	Cuts
10	30	10	5.8	30.8	89.6	8.6	30.0
10	30	20	4.0	25.0	94.5	7.8	28.4
10	30	30	8.4	39.4	95.3	23.0	92.8
10	30	40	9.6	40.0	95.1	45.4[d]	172.0[d]
10	30	50	12.0	46.8	99.5	75.6[e]	303.0[e]
10	30	60	9.8	42.6	98.3	109.0[d]	436.2[d]

[a] 5 problems per class. Values averaged for all 5 problems.
[b] Limit set to 1 minute or 150 iterations per problem.
[c] Limit set to 5 minutes or 200 iterations per problem.
[d] Only 4 problems solved to optimality.
[e] Only 3 problems solved to optimality.

In Tables 4 and 5 we illustrate the effect of an increase in the number of variables and constraints, respectively, on the performance of Algorithm I.

Table 4
Effect of an increase in the number of variables (Algorithm I)[a]

m	n	T	No. solved	Time (seconds)	Iterations	Cuts	Percent covers extended
10	30	30	5	2.8	8.4	39.4	95.3
10	40	30	5	6.4	11.8	44.2	96.5
10	50	30	5	17.7	11.4	45.2	96.3

[a] 5 problems per class.

Table 5
Effect of an increase in the number of constraints (Algorithm I)[a]

m	n	T	No. solved	Time (seconds)	Iterations	Cuts	Percent covers extended
5	30	30	5	0.4	5.8	17.0	95.1
10	30	30	5	2.8	8.4	39.4	95.3
15	30	30	4	75.2	14.8	81.4	95.3
20	30	30	5	40.7	22.2	109.6	92.6

[a] 5 problems per class.
Limit set to 5 minutes or 150 iterations per problem.
Values averaged for all 5 problems. Time for problems not solved within 5 minutes taken to be 5 minutes.

Table 4 shows that as the number of variables increases from 30 to 50, there is a corresponding increase in the time required to solve the problems. This is of course to be expected, since the number of variables increases to the same extent in the linear 0–1 program as in (MLP). Note, however, that the number of iterations increases by only about $\frac{1}{3}$ as the number of variables increases three times. Table 5 shows a marked increase in computing time as well as the number of iterations and cuts as the number of constraints increases. This is due to the fact that the number of inequalities in the linear equivalent of (MLP) sharply rises with the number of constraints of (MLP), hence so does the number of iterations required to generate an appropriate subset of the linear inequalities.

Algorithm I was also tested on the series of problems solved by Taha [21]. Although these problems are not large and have relatively few terms per constraint, we ran them in order to observe the performance of the algorithm on a known set of multilinear 0–1 problems with nonlinear objective functions. As described in Section 3, we chose to linearize the objective function by introducing one new constraint and an appropriate number of new 0–1 variables. The results of this test are reported in Table 6. The symbols m and n denote the number of constraints and variables, respectively, of the original problems (before the above mentioned transformation). Problem 2C, which took 41.4 seconds to solve, seems to be very tightly constrained.

Table 6
Algorithm I tested on Taha's [21] problems

Problem	m	n	Average no. of terms per constraint	Time (seconds)	Iterations	Percent covers extended
1A	3	5	4.3	0.02	4	75.0
1B	3	10	4.3	0.86	36	16.2
1C	3	20	4.3	5.42	56	9.4
2A	7	5	6	0.07	5	82.8
2B	7	10	6	0.15	6	69.2
2C	7	20	6	41.40	91	42.6
2D	7	30	6	6.81	37	46.1
2E	7	5	6	0.03	1	100.0
3A	6	10	4.7	0.02	2	100.0
3B	6	10	4.7	0.02	3	100.0
3C	6	10	4.7	0.04	6	100.0
3D	6	10	4.7	0.18	9	95.8
3E	6	10	4.7	0.17	8	100.0

Finally, in the last two tables we examine the performance of Algorithm I in the heuristic mode. When used as a heuristic, Algorithm I stops at the first (approximate) solution of the linear 0–1 program found by Pivot and Complement that is feasible to (MLP). When Pivot and Complement fails to find a feasible solution, the branch and bound procedure is applied until it finds a first feasible solution.

Table 7
Algorithm I in the heuristic mode[a]

m	n	T	No. solved	Iterations	Time (seconds)	Proximity to LP bound (%)	Proximity to integer optimum (%)
10	30	10	5	5.0	0.3	3.5	0.00
10	30	20	5	4.0	0.1	1.8	0.25
10	30	30	5	8.2	0.6	2.7	0.08
10	30	40	5	8.8	0.6	2.4	0.07
10	30	50	5	11.2	1.1	2.4	0.14
10	30	60	5	9.4	0.8	2.6	0.14

[a] 5 problems per class. Values averaged for all 5 problems.

The linear programming solution to the last linear 0–1 program (more precisely, the lowest value of any LP solved during the procedure), rounded down to the nearest integer, provides an upper bound for the optimum of (MLP), which we call the LP bound. This bound is guaranteed, but in most cases not tight. For the problems of Table 7 the integer optimum is also known, so the quality of the heuristic solution can be measured against the actual optimum. For the problems of Table 8 this is not the case, and the only measure available is the LP bound. On both counts, the quality of the solutions obtained by using Algorithm I in the heuristic mode seems excellent, and the computational effort is modest.

Table 8
Additional tests with the heuristic[a]

m	n	T	No. solved	Iterations	Time (seconds)	Proximity to LP bound (%)
10	30	70	5	23.8	1.4	2.8
10	40	30	5	10.4	1.0	3.0
10	50	30	5	11.2	1.9	1.8
10	50	40	5	14.0	3.3	1.8
10	50	50	5	9.0	1.2	1.6
5	100	30	5	7.0	0.6	0.4
5	150	30	5	6.4	1.0	0.4
5	100	50	5	15.4	6.2	0.8

[a] 5 problems per class. Values averaged for all 5 problems.

We conclude from this computational study that Algorithm I, based on the linearization of [6] and Section 2 is an efficient procedure for solving multilinear 0–1 programs to optimality. In particular, problems having more than 20 terms per constraint have now been opened up to exact solution. The use of the first phase of the algorithm as a heuristic is also an attractive option for problems with many

constraints and/or variables, in that high quality solutions can be obtained at a modest computational cost.

References

[1] E. Balas, "Facets of the knapsack polytope", *Mathematical Programming* 8 (1975) 146–164.

[2] E. Balas and R.G. Jeroslow, "Canonical cuts on the unit hypercube," *SIAM Journal of Applied Mathematics* 23 (1972) 61–69.

[3] E. Balas and C.H. Martin, "Pivot and complement – A heuristic for 0–1 programming," *Management Science* 26 (1980) 86–96.

[4] E. Balas and J.B. Mazzola, "Linearizing nonlinear 0–1 programs: Some new techniques". Paper presented at the ORSA/TIMS Meeting in Milwaukee, October 17–19, 1979.

[5] E. Balas and J.B. Mazzola, "Linearizing nonlinear 0–1 programs". MSRR No. 467, Carnegie-Mellon University, Pittsburgh, PA, October 1980.

[6] E. Balas and J.B. Mazzola, "Nonlinear 0–1 programming: I. Linearization techniques", *Mathematical Programming* 30 (1984) 1–21 (this issue).

[7] D.H. Evans, "Modular design—a special case in nonlinear programming", *Operations Research* 11 (1963) 637–647.

[8] D. H. Evans, "A note on 'Modular design—A special case in nonlinear programming'", *Operations Research* 18 (1970) 562–564.

[9] D. Granot and F. Granot, "Generalized covering relaxation for 0–1 programs", *Operations Research* 28 (1980) 1442–1449.

[10] D. Granot, F. Granot and J. Kallberg, "Covering relaxation for positive 0–1 polynomial programs", *Management Science* 25 (1979) 264–273.

[11] D. Granot, F. Granot and W. Vaessen, "An accelerated covering relaxation algorithm for solving 0–1 positive polynomial programs", Working Paper No. 718, University of British Columbia, 1980.

[12] F. Granot and P. L. Hammer, "On the use of Boolean functions in 0–1 programming", *Methods for Operations Research* 12 (1971) 154–184.

[13] S.S. Hamlen, "A chance-constrained mixed integer programming model for internal control design", *The Accounting Review* LV (1980) 578–593.

[14] P.L. Hammer and S. Rudeanu, *Boolean methods in operations research and related areas* (Springer, Berlin, New York, 1968).

[15] P. Kolesar, "Testing for vision loss in glaucoma suspects", *Management Science* 26 (1980) 439–450.

[16] D.E. Peterson and D. Laughhunn, "Capital expenditure programming and some alternative approaches to risk", *Management Science* 17 (1971) 320–336.

[17] A. Pritsker, L.J. Watters and F. Wolfe, "Multiproject scheduling with limited resources: A zero–one programming approach", *Management Science* 16 (1969) 622–626.

[18] M.R. Rao, "Cluster analysis and mathematical programming", *Journal of the American Statistical Association* 66 (1971) 622–626.

[19] K.E. Stecke, "Nonlinear MIP formulations of production planning problems in flexible manufacturing systems", Working Paper No. 293, GSBA, University of Michigan, March 1982.

[20] K.E. Stecke and J.J. Solberg, "The optimality of unbalanced workloads and machine group sizes for flexible manufacturing systems", Working Paper No. 290, GSBA, University of Michigan, January 1982.

[21] H.A. Taha, "A Balasian-based algorithm for zero–one polynomial programming", *Management Science* 18B (1972) 328–343.

[22] W. Zangwill, "Media selection by decision programming", *Journal of Advertising Research* 5 (1965) 30–36.