Mathematica Balkanica

Mathematical Society of South-Eastern Europe
A quarterly published by
the Bulgarian Academy of Sciences – National Committee for Mathematics

Mathematica
Balkanica

New Series Vol. 2, 1988, Fasc. 1

A Method for Construction of Strongest Cuts and Generation of Minimal Covers for 0-1 Linear Programming Problems

Rumen A. Andonov

Presented by P. Kenderov

A combined algorithm for obtaining tighter equivalent formulation of a given 0-1 linear programming problem is presented. This algorithm incorporates rotation of an original constraint and addition of new constraints issued from minimal covers. A criterion for selecting those minimal covers which chop off given point x^* is proposed. The computational experiments with ten test problems are presented.

Introduction

Consider the following 0-1 linear programming problem:

$$\text{(P)} \qquad \max \{cx:\ Ax \leqq b,\quad x_j \in \{0,\ 1\},\quad j \in N = \{1,\ 2, \dots,\ n\}\},$$

where all data are integers, A is m-by-n matrix, b and c are vectors of length m and n, respectively. (P) belongs to the NP-complete or "hard" combinatorial optimization problem. Nevertheless, impressive computational results for solving problem (P) have been reported recently in numerous papers (see, e. g., [4, 8, 12, 13]). Two of the ideas which contribute to the advance in this field are:

1) To obtain tighter equivalent formulation of a given problem (P) using the rotation of a given constraint without adding or eliminating any integer feasible solution. This method is studied by F. Kianfar in [10, 11] and I. Kaliszewski and S. Walukiewicz in [7, 8, 9].

2) To chop off part of the feasible set of linear relaxation of (P) by cutting planes, which are facets of the underlying polytope. This approach is investigated by E. Balas in [1, 2], M. Padberg in [14, 15, 16], R. Kowal in [13], and H. Crowder, E. Johnson and M. Padberg in [4].

In this paper a hybrid algorithm which incorporates the two approaches mentioned above is presented. In section 1 we give the description of the algorithm which rotates given constraint and generates simultaneously a great number of minimal covers. Using them one obtains valid inequalities. In section 2 a criterion for selecting those minimal covers whose associated minimal cover inequalities chop off given point x^* is proposed. In section 3 the lifting procedure studied by M. Padberg in [14] and E. Zemel in [18] is described. We make some slight improvements. In section 4 we discuss the entire algorithm for solving problem (P) — a combination of constraint rotation, cutting planes, and branch-and-bound techniques. Some results of our computational experiments are presented in section 5.

1. Constructing strongest cuts using dynamic programming and generating minimal covers.

It is well known that any integer programming problem has infinitely many equivalent formulations. In general, the time for solving different equivalent formulations varies substantially. The computational experiments in [7, 8, 11, 17, 19] illustrate the advantage of a previous processing over the constraints of (P) in order to obtain tighter equivalent formulation.

For a matrix A and vector b we denote: $P(A, b)=\{x\in R^n: Ax\leqq b, 0\leqq x_j\leqq 1, j\in N\}$ and $P_I(a, b)=\mathrm{conv}\{x\in P(A, b): x_j\in\{0, 1\}, j\in N\}$. An integer programming problem

$$\text{(P')} \qquad \max\ \{cx: A'x\leqq b',\ x_j\in\{0, 1\},\ j\in N=\{1, 2,\dots, n\}\},$$

is called equivalent to (P) if and only if $P_I(A, b)=P_I(A', b')$. (P′) is tighter equivalent formulation of (P) if and only if

$$P(A', b')\subseteq P(A, b).$$

Usually, to obtain tighter equivalent formulation we process every constraint separately. Any constraint of (P) can be transformed to an equivalent knapsack type constraint:

$$(1) \qquad g(x)=\sum_{j\in N} a_j x_j\leqq a_0, \quad x_j\in\{0, 1\}, \quad j\in N,$$

where $0\leqq a_j\leqq a_0$, $\Sigma_{j\in N}\ a_j>a_0$.

A set $S\subseteq N$ is called cover for (1) if $\Sigma_{j\in S}\ a_j>a_0$. A cover S is called minimal cover for (1) if $\Sigma_{j\in Q}\ a_j\leqq a_0$ for any proper subset Q of S.

Obviously, any solution of (1) satisfies the inequality

$$(2) \qquad \sum_{j\in S} x_j\leqq |S|-1,$$

where $|S|$ denotes the cardinality of the minimal cover S.

The general method for constructing an equivalent inequality by the reduction of coefficients of inequality (1) is presented in [3]. This method is impractical because it uses the set of all minimal covers, the number of which may grow exponentially with n.

Another approach is investigated in [7, 8, 9, 11]. F. Kianfar in [10, 11] described a method for the rotation the hyperplane $g(x)=a_0$ without adding or eliminating any integer point of the feasible solution set. The new hyperplane

(3) $$\bar{g}(x)=\sum_{j\in N}\bar{a}_j x_j=a_0$$

passes through at least as many integer points as $g(x)=a_0$.

If the hyperplane $g(x)=a_0$ cannot be rotated, the corresponding inequality $g(x)\leqq a_0$ is a strongest cut or a strongest constraint. The inequality (1) is called a strongest constraint if for

$$\forall_{i\in N}\quad \exists'_{x\in F}\ :\ \sum_{j\in N} a_j x_j=a_0 \quad\text{and}\quad x_i=1,$$

where F is the feasible set for (1).

In [7, 9] I. Kaliszewski and S. Walukiewicz develop the above method by introducing the so-called best direction of rotation and study a lot of properties of the method.

In this paper we proceed to develop and modify the previous method. By rotating a given constraint we simultaneously generate a great number of minimal covers. In this way we construct minimal cover inequalities (2). The generated new cuts give even still tighter equivalent formulation. Furthermore, the minimal covers have good properties. If some additional conditions are imposed, using them one obtains facets of the underlying polytope. The computational complexity of the modified algorithm is the same as the original algorithm $O(n^2 a_0)$. Below we briefly describe the rotation procedure, emphasizing the modifications.

In order to compute the new coefficients of the strongest cut (3) we have to solve for all $r\in N$ the following knapsack problem:

$$b_r^*=\max\{\sum_{j\in N-\{r\}} a_j x_j : x\in S_r\}$$

where $S_r=\{x\in R^n : \sum_{j\in N-\{r\}} a_j x_j\leqq a_0-a_r,\ x_j\in\{0, 1\}, j\in N-\{r\}\}$. As a new value of a_r we take $\bar{a}_r=a_0-b_r^*$.

Then the inequality $\sum_{j\in N-\{r\}} a_j x_j+\bar{a}_r x_r\leqq a_0$ does not eliminate any element of F. When $b_r^*<a_0-a_r$, the hyperplane $\sum_{j\in N-\{r\}} a_j x_j+\bar{a}_r x_r=a_0$ passes through at least one more binary point than $g(x)=a_0$. The value of b_r^* is computed as the maximal element of the set

$$B_r=\{b : b=\sum_{j\in J} a_j,\quad J\subset N-\{r\},\quad b\leqq a_0-a_r\}.$$

The sets B_r, $r=1, 2, \ldots, n$, are constructed by dynamic programming (see also [10, 9, 7].

Extended Rotation and Constraint Generation Procedure.

Step 1: (Initialization, $r=1$). Set $B_1^n=\{0, a_n\}$ and compute

(4) $$B_1^i=B_1^{i+1}\cup\{b+a_i : b\in B_1^{i+1},\quad b+a_i\leqq a_0\}\quad\text{for}\quad i=n-1,\ldots, 2.$$

Set $B_1=\{b: b\in B_1^2,\ b\leqq a_0-a_1\}$. Compute b_1^* and $\bar{a}_1=a_0-b_1^*$.

The upper bound of the additions and the comparisons at this stage is $0(na_0)$.

In the well-known from [10, 9] algorithm, if the sum $b+a_i$ in (4) is greater than a_0, it is rejected and forgotten. We use this sum in the following manner: Let the set $Q\subset N$ be such that $b=\Sigma_{j\in Q}\ a_j$ and $b+a_i>a_0$ where $b\in B_1^{i+1}$ and $1<i<n$. If we have

$$a_i\leqq\min\{a_j: j\in Q\}, \tag{5}$$

then the set $S=Q\cup\{i\}$ is obviously a minimal cover. To satisfy condition (5) we first have to order the coefficients in (1)

$$a_j\leqq a_{j+1} \quad \text{for} \quad j\in N. \tag{6}$$

Then in step 1 we make the following modification.

For every element $b\in B_1^i$ in (4) save and store the associated index set Q such that $b=\Sigma_{j\in Q\ \cup\{i\}}a_j$.

If $b+a_i>a_0$, then $Q\cup\{i\}$ is a minimal cover. In this way any calculation made in step 1 is used and at the same time through set B_1 we obtain a great number of minimal covers. *Thus, the extended rotation procedure is also a constraint generation procedure.* We illustrate the newly obtained algorithm by the following example.

Example 1.

$$6x_1+15x_2+15x_3+26x_4+38x_5\leqq 45 \tag{7}$$

The set of all minimal cover of (7) is:

$$\{1, 2, 4\},\quad \{1, 3, 4\},\quad \{2, 3, 4\},\quad \{2, 5\},\quad \{3, 5\},\quad \{4, 5\}.$$

Remark: In this example the sets Q are written above the associated sum b. When calculating the sums $b+a_i$, the sets B_1^{i+1} are scanned from right to left.

Following the algorithm we compute:

$$B_1^5=\{0,\ \overset{\overset{5}{|}}{38}\};\quad B_1^4=\{0,\ \overset{\overset{5}{|}}{38},\ \overset{\overset{4}{|}}{26}\}.$$

At this stage the inequality $a_5+a_4>a_0$ implies the minimal cover $\{5, 4\}$. We obtain also:

$$B_1^3=\{0,\ \overset{\overset{5}{|}}{38},\ \overset{\overset{4}{|}}{26},\ \overset{\overset{3}{\overset{4}{|}}}{41},\ \overset{\overset{3}{|}}{15}\}$$

– the inequality $a_5+a_3>a_0$ yields the minimal cover $\{5, 3\}$,

$$\begin{array}{rccccc} & & & 3\;2 & & 2\\ & 5 & 4 & 4\;4 & 3\;2 & 3\\ & | & | & \backslash\,/ & \backslash\;/ & |\\ B_1^2=\{0, & 38, & 26, & 41, & 15, & 30\}\end{array}$$

Here, the inequalities $a_4+a_3+a_2>a_0$ and $a_5+a_2>a_0$ yield the minimal covers $\{4, 3, 2\}$ and $\{5, 2\}$, respectively.

We have at this stage $a_4+a_2=a_4+a_3=41$ and $a_3=a_2=15$. If we save the information for all index sets Q associated to any sum b, we would be able to find all minimal covers. To prevent exponential growth of the computational complexity we choose and store for any b just one set Q, such that $b=\Sigma_{j\in Q}\,a_j$.

Changing this criterion one obtains different subsets of minimal covers. This fact will be essentially used later. At the present moment we use the following criterion:

Rule 1: *For every element $b\in B_1^i$, $i=2, 3,\ldots, n$, choose and store the index set Q latest obtained, such that $b=\Sigma_{j\in Q}\,a_j$.*

Then,

$$\begin{array}{rccccc} & & & 2 & & 2\\ & 5 & 4 & 4 & 2 & 3\\ & | & | & | & | & |\\ B_1^2=\{0, & 38, & 26, & 41, & 15, & 30\}.\end{array}$$

In addition, we use the following:

Rule 2: *Let the index sets Q and L be such that $\Sigma_{j\in Q}\;a_j=\Sigma_{j\in L}\;a_j=b$ and* $\min\{j: j\in L\}=i$, *where the element $b\in B_1^i$, $2\leqq i\leqq n$.*

In case L should replace according to Rule 1 the set Q, then construct the sum $b+a_i$ (if it is not yet done) and after that reject the set Q.

In this step the storing of the sets Q and their eventual replacements according to Rule 1 increase the upper bound of the operations to $0(n^2 a_0)$.

At last from B_1^2 one obtains $B_1=\{0, 38, 26, 15, 30\}$. The inequality $a_4+a_2+a_1>a_0$ yields the minimal cover $\{4, 2, 1\}$. In this way 5 from all 6 minimal covers were found. The minimal cover $\{4, 3, 1\}$ was lost because we abandoned the set $\{4, 3\}$.

Let us calculate $b_1^*=\max\{j: j\in B_1\}=38$ and $\bar{a}_1=7$. The set $Q=\{5\}$ satisfies $\Sigma_{j\in Q}\,a_j=38=b_1^*$.

For this equality the coefficients associated to the index set Q cannot vary. Thus, the saving of the corresponding to every sum b set Q helps to find the new coefficients. It is also a slight improvement to the well-known rotation procedure. This change is incorporated in Step 2.

Step 2: $(2\leqq r\leqq n-1)$

Set $r=r+1$. If $r=n$, then go to Step 3. Otherwise,

$$B_r^r=\{b: b\in B_1^{r+1},\quad b\leqq a_0-a_r\}.$$

For $i=r-1,\ldots, 1$ compute

$$B_r^i = B_r^{i+1} \cup \{b + i\bar{a}_i:\ b \in B_r^{i+1},\quad b + \bar{a}_i \leqq a_0 - a_r\}.$$

Set $B_r = B_r^1$ compute b_r^* and $\bar{a}_r = a_0 - b_r^*$.
For the set $Q \subset N$, such that $b_r^* = \Sigma_{j \in Q}\ a_j$, *set* $\bar{a}_i = a_i$ for $i \in Q$.
Step 3: (Termination, $r = n$). Set $B_n^0 = \{0\}$, $i = 1$ and compute

$$B_n^i = B_n^{i-1} \cup \{b + \bar{a}_i:\ b \in B_n^{i-1},\quad b + \bar{a}_i \leqq a_0 - a_r\}$$

for $i = 1, \ldots, n-1$. Set $B_n = B_n^{n-1}$, compute b_n^* and $\bar{a}_n = a_0 - b_n^*$. Stop.
The calculations for example 1 are presented next.
For $r = 2$ we obtain

$$B_2^2 = \{0, 26, 15\},\quad B_2^1 = \{0, \overset{\overset{4}{|}}{26}, 15, 22, 7\},\quad b_2^* = 26,\quad \bar{a}_2 = 19,\quad \bar{a}_4 = a_4 = 26.$$

Analogously, for $r = 3$ we compute $b_3^* = 26$, $\bar{a}_3 = a_0 - b_3^* = 19$. Thus, the new inequality is

$$7x_1 + 19x_2 + 19x_3 + 26x_4 + 38x_5 \leqq 45.$$

It can be shown [10] that step 2 and step 3 require in the worst case $0(n^2 a_0)$ additions and comparisons. Thus, the computational complexity for step 1, step 2, and step 3 is $0(n^2 a_0)$.

2. Finding a minimal cover cut for given point x^*

The following problem is presented in [3]: Given point x^* find a minimal cover inequality (2), which chops off x^* if such an inequality exists.

Next proposition is stated there as well:

For a given point x^* there exists a minimal cover inequality (2) *if and only if $z^* < 1$, where*

$$z^* = \min\{\sum_{j \in N}(1 - x_j^*)h_j:\ \sum_{j \in N} a_j h_j > a_0,\quad h_j \in \{0, 1\},\quad j \in N\}.$$

Here we present another approach to the above-mentioned problem. The algorithm described in section 1 is used.

Suppose S is minimal cover for (1) such that

$$\sum_{j \in S} x_j^* > |S| - 1 \tag{8}$$

for given point x^*. Let us assume that in the extended rotation procedure we have obtained sets Q and L such that:

$$\sum_{j \in Q} a_j = \sum_{j \in L} a_j = b$$

for any $b \in B_1^i$, $i \in N$. In this case, according to Rule 1, no more than one set is saved. Then, it is possible to lose exactly that set Q or L, which generates the minimal cover S. For that reason Rule 1 has to be modified. We propose the following:

Rule 1A: *Assume for any $b \in B_1^i$, $2 \leqq i \leqq n$, there exist two index sets Q and L such that*

$$\sum_{j \in L} a_j = \sum_{j \in Q} a_j = b \text{ and suppose } \sum_{j \in L} x_j^* \geqq \sum_{j \in Q} x_j^*$$

1) *If $|L| \leqq |Q|$, then save the set L.*
2) *Let $|L| > |Q|$. Denote by d the difference $d = |L| - |Q|$.*

$$\text{If } \sum_{j \in Q} x_j^* + d > \sum_{j \in L} x_j^*, \text{ then save the set } Q.$$

$$\text{If } \sum_{j \in Q} x_j^* + d \leqq \sum_{j \in L} x_j^*, \text{ then save the set } L.$$

Theorem 1. *Assume for given point x^* there exists minimal cover S such that $\Sigma_{j \in S} x_j^* > |S| - 1$.*

The previous described algorithm with Rule 1A finds at least one minimal cover T such that $\Sigma_{j \in T} x_j^* > |T| - 1$.

Proof: Suppose the assumptions in Rule 1A are satisfied. It is easy to observe that in this case the computations in (4) imply either $\min\{j : j \in L\} = i$ or else $\min\{j : j \in Q\} = i$ to be satisfied. Denote by S^* the set of all minimal covers for (1). We will show that applying Rule 1A we remain always in such subsets of S^* for which violated minimal cover inequalities for given point x^* exist.

1. Let $|L| \leqq |Q|$.

Then in Rule 1A we have to choose the set L. Our arguments are the following:

If $L \subset S$, everything is all right.

Suppose $Q \subset S$. There are two cases:

1.1 $i \in S - Q$. Then we have $\min\{j : j \in L\} = i$. According to Rule 2, the set $Q \cup \{i\}$ shall not be rejected. Then it is not possible to lose the minimal cover S in this case.

This argument is analogous to cases 2.2 and 2.1, and for that reason it will not be discussed there.

1.2. $i \notin S - Q$. Denote by S_Q any minimal cover such that for some $R \subset N$ we have $S_Q = Q \cup R$ and $i \notin R$. Obviously, S is such a minimal cover. Since $\Sigma_{j \in Q} a_j = \Sigma_{j \in L} a_j$ and the indices of S_Q decrease, then $S_L = L \cup R$ is a minimal cover as well.

Suppose $\Sigma_{j \in S_Q} x_j^* > |S_Q| - 1$. Since $\Sigma_{j \in L} x_j^* \geqq \Sigma_{j \in Q} x_j^*$, this yields

$$\sum_{j \in S_L} x_j^* = \sum_{j \in L} x_j^* + \sum_{j \in R} x_j^* \geqq \sum_{j \in Q} x_j^* + \sum_{j \in R} x_j^* = \sum_{j \in S_Q} x_j^* > |S_Q| - 1$$
$$= |Q| + |R| - 1 \geqq |L| + |R| - 1 = |S_L| - 1.$$

Hence, $\Sigma_{j\in S_L} x_j^* > |S_L| - 1$ and this inequality chops off also x^*.

2. Let $|L| > |Q|$ and $d = |L| - |Q|$.

2.1. $\Sigma_{j\in Q} x_j^* + d > \Sigma_{j\in L} x_j^*$. Suppose $\Sigma_{j\in S_L} x_j^* > |S_L| - 1$. Then we have

$$\sum_{j\in S_Q} x_j^* = \sum_{j\in Q} x_j^* + \sum_{j\in R} x_j^* > \sum_{j\in L} x_j^* + \sum_{j\in R} x_j^* - d = \sum_{j\in S_L} x_j^* - d > |S_L| - 1 - d$$

$$= |L| + |R| - 1 - d = |Q| + |R| - 1 = |S_Q| - 1.$$

Hence, $\Sigma_{j\in S_Q} x_j^* > |S_Q| - 1$. Consequently, in this case we should save the set Q.

2.2. $\Sigma_{j\in Q} x_j^* + d \leqq \Sigma_{j\in L} x_j^*$. In this case from the assumption $\Sigma_{j\in S_Q} x_j^* > |S_Q| - 1$ the inequality $\Sigma_{j\in S_L} x_j^* > |S_L| - 1$ follows. Indeed, it is easy to see that

$$|S_L| - 1 = |L| + |R| - 1 = |Q| + d + |R| - 1 = |S_Q| + d - 1 < \sum_{j\in S_Q} x_j^* + d$$

$$= \sum_{j\in Q} x_j^* + \sum_{j\in R} x_j^* + d \leqq \sum_{j\in L} x_j^* + \sum_{j\in R} x_j^* = \sum_{j\in S_L} x_j^*.$$

We thus conclude that in the last case we should save the set L.

3. Strong covers, facets of the knapsack polytope and lifting procedure

In this section we briefly present some results of E. Balas [1] concerning a class of minimal covers, which construct facets of the knapsack polytope. These results are incorporated in our algorithm. Here we present also the well-known lifting procedure (see [14, 18]) with slight improvements.

In this section by P_I we denote the set:

$$P_I = \text{conv}\{x \in R^n : \sum_{j\in N} a_j x_j \leqq a_0, \quad x_j \in \{0, 1\}, \quad j \in N\}.$$

Suppose also the inequality $a_j \leqq a_{j+1}$, $j \in N$, satisfied. Let us assume S is minimal cover and denote by S' the set $S' = \{j \in N - S : a_j > a_{j1}\}$, where $a_{j1} = \max\{a_j : j \in S\}$. The set $E(S) = S \cup S'$ is called extension of S to N.

The set $S \subseteq N$ is a strong cover for (1) if and only if S is a minimal cover and either $E(S) = N$ or else

$$\sum_{j\in (S - \{j1\}) \cup \{i1\}} a_j \leqq a_0 \quad \text{is true,}$$

where $j1$ is defined as above and $a_{i1} = \max\{a_j : j \in N - E(S)\}$.

An inequality $fx \leqq f_0$ is called a facetial inequality (or simply a facet) for the polytope P_I if

(i) $x \in P_I$ implies $fx \leqq f_0$

(ii) there exist exactly d affinely independent points x^i of P_I satisfying $fx^i = f_0$ for $i=1, \dots, d$, where $d = \dim P_I$.

The following theorem is due to E. Balas [1].

Theorem 2. *The inequality $\Sigma_{j \in M} x_j \leqq k$, where $|M| > 2$, $M \subseteq N$, $k \geqq 0$, defines a facet of P_I if and only if M is the extension of a strong cover S for* (1) *such that $|S| = k+1$ and $\Sigma_{j \in T} a_j \leqq a_0$, $T = (S - \{j1, j2\}) \cup \{n\}$ with $j1$ defined as above and $j2$ by $a_{j2} = \max \{aj : j \in S - \{j1\}\}$.*

The following theorem [15] answers the question whether a facetial inequality retains its property of defining a facet when the number of variables increases.

Theorem 3. *Denote by P_I^S the polytope: $P_I^S = P_I \cap \{x \in R^n : x_j = 0$ for all $j \in N - S\}$. If the inequality $\Sigma_{j \in S} f_j x_j \leqq f_0$ is a facet of P_I^S, then there exist non-negative numbers $q_j, q_j \leqq f_0$ such that*

$$\sum_{j \in S} f_j x_j + \sum_{j \in N-S} q_j x_j \leqq f_0 \text{ is a facet of } P_I.$$

That is why it is useful to find a facet even for a low dimensional polytope. Moreover, the lifting procedure [14, 18] used to compute the new coefficients q_j, $j \in N - S$, can be applied to any valid inequality for P_I.

Suppose (9) is a valid inequality for P_I or a facet for P_I^S

$$\sum_{j \in S} f_j x_j \leqq f_0. \tag{9}$$

Let $k \in N - S$ and

$$z_k = \max \{ \sum_{j \in S} f_j x_j : \sum_{j \in S} a_j x_j \leqq a_0 - a_k, \quad x_j \in \{0, 1\}, \quad j \in S \}. \tag{10}$$

We redefine S to be $S \cup \{k\}$, set $q_k = f_0 - z_k$, and repeat that until the set $N - S$ is completely exhausted.

In the lifting procedure (10) the order of obtaining the new coefficients q_k is not fixed. The following example illustrates how important it is to define the sequence of scanning the set $N - S$.

Example 2.

$$\max \quad 1200x_1 + 1300x_2 + 1300x_3 + 1200x_4 + 899x_5 + 999x_6 + 899x_7 + 1099x_8$$

$$\text{s. t.} \quad 13x_1 + 13x_2 + 13x_3 + 13x_4 + 9x_5 + 10x_6 + 9x_7 + 11x_8 \leqq 39$$

$$x_j \in \{0, 1\}, \quad j = 1, 2, \dots, 8.$$

Suppose we have found the valid inequality

$$x_1 + x_2 + x_3 + x_4 + x_6 + x_8 \leqq 3. \tag{11}$$

After adding the new constraint (11), the solution of the relaxed problem is

$$z_{LP}=3898.78, \quad x_2^*=x_3^*=x_8^*=1, \quad x_7^*=0{,}2223, \quad x_1^*=x_4^*=x_5^*=x_6^*=0$$

If the lifting procedure (10) scans the set $N-S=\{5, 7\}$ from left to right, then one obtains $q_5=1$ and $q_7=0$. The lifted inequality is

$$x_1+x_2+x_3+x_4+x_5+x_6+x_8\leqq 3$$

It does not chop off x^*.

If the lifting procedure (10) scans the set $N-S=\{5, 7\}$ from right to left, then one obtains $q_7=1$ and $q_5=0$. The lifted inequality is

$$x_1+x_2+x_3+x_4+x_7+x_6+x_8\leqq 3.$$

It chops off x^*.

Any scanning of the set $N-S$ can be connected with a permutation $p=\{p_1, p_2\ldots, p_s\}$, where $s=|N-S|$. It means that, first, we take index $p_1\in N-S$, after that, index $p_2\in(N-S-\{p_1\})$, etc.

Let us assume $p=\{p_1,\ldots, p_s\}$ and $r=\{r_1,\ldots, r_s\}$ are two permutations of the set $N-S$, which satisfy the conditions: $r_i=p_i$ for $i\neq l$, $i\neq m$, for any indices l and m such that $1\leqq l\leqq s$ and $1\leqq m\leqq s$.

Theorem 4. *Assume in the lifting procedure* (10) $k=p_l=r_m$. *If* $l<m$, *then* $f_k^p\geqq f_k^r$, *where* f_k^p *and* f_k^r *are the coefficients computed for the permutations* p *and* r, *respectively.*

Proof: For simplicity of presentation we denote: $A=\{p_j: 1\leqq j\leqq l-1\}$; $B=\{p_j: l+1\leqq j\leqq m-1\}$; $C=\{p_j: m+1\leqq j\leqq s\}$. Then $p=\{A, p_l, B, p_m, C\}$ and $r=\{A, p_m, B, p_l, C\}$. According to the lifting procedure (10) we have: $f_k^p=f_0-z_k^p$, where $z_k^p=\max\{\Sigma_{j\in A}f_jx_j:\Sigma_{j\in A}a_jx_j\leqq a_0-a_k,\ x_j\in\{0,1\},\ j\in A\}$ and $f_k^r=f_0-z_k^r$, where $z_k^r=\max\{\Sigma_{j\in A\cup\{p_m\}\cup B}f_jx_j:\Sigma_{j\in A\cup\{p_m\}\cup B}a_jx_j\leqq a_0-a_k,\quad x_j\in\{0,1\},\quad j\in A\cup\{p_m\}\cup B\}$.

We thus conclude that $z_k^p\leqq z_k^r$. Therefore, $f_k^p\geqq f_k^r$.

From the previous theorem it follows that it is better to compute first the coefficients of the non-zero variables and after that — the coefficients of the zero variables.

4. Realization of the algorithm in the case of general 0-1 linear programming problem

Let us have problem (P). Denote by (a^i, b_i) the i-th row of (A, b) and define by $P_I^i=\operatorname{conv}\{x\in R^n: a^ix\leqq b_i,\ x_j\in\{0,\ 1\},\ j\in N\}$ the convex hull of the zero-one solution to the single inequality $a^ix\leqq b_i$, where $i\in M$. Then the inclusion

$$P_I(A,\ b)\subseteq\bigcap_{i=1}^{m}P_I^i \tag{12}$$

is satisfied. The equality in (12) does not, in general, hold, but it holds if problem (P) decomposes totally into m knapsack problems. For that reason the computational effectiveness increases when matrix A is sparse. This fact is confirmed by the computations of [4], as well as by our computations.

Step 1. First, we relax the zero-one problem (P) to its associated linear problem, i. e., we replace the integrality condition $x_j \in \{0, 1\}$ by $0 \leqq x_j \leqq 1$ for all $j \in N$. The relaxed problem is solved by Dantzig's method. Denote by $f(x_{LP})$ it's solution.

Step 2. If x_{LP} is not integer, then we apply on every constraint $a^i x_{LP} = b_i$ the Extended Rotation and Constraint Generation Procedure with Rule 1A. Every generated minimal cover is checked if the associated minimal cover inequality chops off x_{LP}. If there is not such a minimal cover, we proceed with the next active constraint. Else, if the so obtained violated inequality is not a facet, it is extended using the lifting procedure. Then the new constraint is added and the newly constructed 0-1 linear programming problem is solved in relaxed form. If the new obtained solution x_{LP} is not integer, we continue the Extended Rotation and Constraint Generation Procedure.

Step 3. If every active constraint for the point x_{LP} is rotated and the addition of the new constraints in Step 2 does not yield an integer point x_{LP}, then we proceed with branch-and-bound techniques.

5. Computational experiments

A program based on the method described previously has been developed. It is coded in FORTRAN and is named COFACE. The branch-and-bound phase is realized by the code BRMIP (see [20] for more details).

			BRMIP		COFACE+BRMIP			
N	n	m	t	it	it_2	m_1	t	it
1	11	4	1.6	236	18	5	1.6	201
2	13	4	3.5	440	20	5	1.9	209
3	14	4	4.0	625	30	8	4.2	455
4	11	4	1.4	183	20	5	1.6	156
5	22	4	103.	+20000	40	11	88.3	6929
6	23	4	108.	+20000	38	12	72.3	5251
7	22	3	203.	+20000			88.9	11001
8	23	3	202.	+20000			53.9	5248
9	15	6	4.0	525	22	6	1.4	94
10	30	10	47.6	4420	39	8	15.2	479

Where:

N – number of Haldi's test problems in it's binary expansion
n – number of binary variables
m – number of constraints
t – time in seconds
it – number of all iterations (when + the optimality is not obtained)
it_2 – number of iterations made in COFACE
m_1 – number of added constraints in COFACE

Here we present ten test examples of H. Haldi [5, 6]. These problems are solved in two ways. The first uses only BRMIP. The second uses COFACE and BRMIP. If the addition of new constraints in COFACE does not yield an integer point, then branch-and-bound processing continues in BRMIP. The runs were executed on IBM 370/148. The results obtained are given below in the table.

The program BRMIP has been elaborated in the Institute of Mathematics of the Bulgarian Academy of Sciences for the general integer programming problem. The ten original Haldi's examples were solved easily, but their binary expansions produced lot of troubles of BRMIP. It concerns mostly the problems 5-6, where the computational effectiveness of COFACE is greatest.

These results confirm the thesis that in numerous cases a preliminary preprocessing, like the one described in this paper, decreases considerably the iterations in branch-and-bound methods and at the end speeds up the solving of problem (P).

References

1. E. Balas. Facets of the knapsack polytope. *Math. Programming*, **8**, 1975, 146-164
2. E. Balas, E. Zemel. Facets of the knapsack polytope from minimal covers. *SIAM J. Appl. Math.*, **34**, 1978, 119-148.
3. G. H. Bradley, P. L. Hammer, L. Wolsey. Coefficient reduction for inequalities in (0—1) variables. *Math. Programming*, **7**, 1974, 263-282.
4. H. Crowder, E. Johnson, M. Padberg. Solving large scale linear zero-one programming problems. *Oper. Res.*, **31**, 1983, 803-834.
5. R. Garfinkel, G. Nemhauser. Integer Programming. New York, 1972.
6. H. Haldi. 25 integer programming test problems. Working Paper No 43, Graduate School of Business, Stanford Univ., 1964.
7. I. Kaliszewski, S. Walukiewicz. Tighter equivalent formulations of integer programming problems. — In: Survey of Mathematical programming. Budapest, 1974, 523-536.
8. I. Kaliszewski, S. Walukiewicz. A Transformations of Integer Programs. Preprint-ZPM-2/81. Systems Res. Inst., Polish Acad. Sci., Warsaw, 1981.
9. I. Kaliszewski. Constraint modification method for LIP. *Control and Cybernetics*, **10**, 1981, No 34, 217-229.
10. F. Kianfar. Stronger inequalities for 0-1 integer programming using knapsack function. *Oper. Res.*, **19**, 1971, 1374-1392.
11. F. Kianfar. Stronger inequalities for 0-1 integer programming using knapsack function: computational refinements. *Oper.* Res., **24**, 1976, 581-585.
12. R. Kipp-Martin, L. Schrage. Subset coefficient reduction cuts for 0/1 mixed integer programming. *Oper. Res.*, **33**, 1985.
13. R. Kowal. Generowanie organiczen dodatkowich dla zadan binarnych. PH. D. Thesis, Warsaw, 1986.
14. M. Padberg. A note on zero-one programming. *Oper. Res.*, **23**, 1975, 833-837.
15. M. Padberg. Covering, packing and knapsack problems. *Ann. Discrete Math.*, **4**, 1979, 265-287.
16. M. Padberg. (1, *k*)-configurations and facets for packing problems. *Math. Programming*, **18**, 1980, 94-99.
17. H. P. Williams. Experiments in the formulation of integer programming problems. *Math. Programming Study*, **2**, 1974, 180-197.
18. E. Zemel. Lifting the facets of zero-one polytope. *Math. Programming*, **15**, 1978, 268-277.
19. Р. Андонов. Еквивалентни формулировки на задачата на целочисленото програмиране с булеви променливи. Математика и механика и приложението им в науката и практиката — (сборник резюмета доклади) София, 1985.
20. М. Иванчев, И. Митев, Н. Янев. Реализация метода ветвей и границ для общей задачи частично-целочисленого программирования. *Ж. вычисль. мат. физ.*, 1976, No 3, 788-793

Institute of Mathematics,
Bulgarian Academy of Sciences,
1090 Sofia, P. O. Box 373
BULGARIA

Received 21. 01. 1987