ELSEVIER

Operations Research Letters 16 (1994) 255-263

operations research letters

Lifted cover facets of the 0–1 knapsack polytope with GUB constraints

George L. Nemhauser [a,*,1], Pamela H. Vance [b]

[a] School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA

[b] Department of Industrial Engineering, Auburn University, Auburn, AL 36849-5346, USA

Received 23 February 1994; revised 9 August 1994

Abstract

Facet-defining inequalities lifted from minimal covers are used as strong cutting planes in algorithms for solving 0–1 integer programming problems. In this paper we extend the result of Balas and Zemel by giving a set of inequalities that determines the lifting coefficients of facet-defining inequalities of the 0–1 knapsack polytope for any ordering of the variables to be lifted. We further generalize the result to obtain facet-defining inequalities for the 0–1 knapsack problem with generalized upper bound constraints.

Keywords: Cover facets; Knapsack polytope; GUB constraints

1. Introduction

Facet-defining inequalities lifted from minimal covers are used as strong cutting planes in algorithms for solving 0–1 integer programming problems. We extend the results of Balas [1] and Balas and Zemel [2] by giving a set of inequalities that determines the lifting coefficients of facet-defining inequalities of the 0–1 knapsack polytope for any ordering of the variables to be lifted. We further extend these results to obtain facet-defining inequalities for the 0–1 knapsack problem with *nonoverlapping* constraints of the form

$$\sum_{j \in S_l} x_j \leqslant 1 \quad \forall l \in L,$$

which we refer to as *generalized upper bound* (GUB) constraints. By nonoverlapping we mean that each variable may appear in at most one GUB constraint. Without loss of generality, we assume that every variable belongs to some GUB, since $|S_l| = 1$ yields trivial GUBs. Wolsey [7] presents some valid inequalities for the 0–1 knapsack with GUBs.

* Corresponding author.

[1] This research was supported by NSF Grant DDM9115768.

SSDI 0167-6377(94)00051-4

For 0-1 integer linear programs, strong cutting planes are derived from lifted cover inequalities. Suppose a 0-1 program contains the inequality

$$\sum_{j\in N} a_j x_j \leqslant b$$

with $x_j \in \{0,1\}$ for all $j \in N$. Without loss of generality, since 0-1 variables can be complemented, assume $a_j > 0$ for all $j \in N$ and suppose that $a_1 \geqslant a_2 \geqslant \cdots \geqslant a_n > 0$ and $a_1 \leqslant b$.

The set $C \subseteq N$ is called a *cover* if $\sum_{j\in C} a_j > b$. The cover C is *minimal* if for each $k \in C$, $\sum_{j\in C} a_j - a_k \leqslant b$. Covers give rise to a class of valid inequalities of the form

$$\sum_{j\in C} x_j \leqslant |C| - 1,$$

called *cover inequalities*. However, these cover inequalities generally need to be *lifted* in order to obtain a facet of the knapsack polytope associated with the original constraint.

A *lifted cover inequality* (LCI) is an inequality of the form

$$\sum_{j\in C} x_j + \sum_{j\in \bar{C}} \alpha_j x_j \leqslant r - 1,$$

where C is a minimal cover, $\bar{C} = N \setminus C$, and $r = |C|$. The *lifting coefficients*, α_j, are nonnegative integers. Let $k_1, \ldots, k_{n-r}$ be any ordering of the elements of $\bar{C}$. Then LCI is a valid inequality if and only if

$$\begin{aligned}
&\alpha_{k_i} \leqslant r - 1 - \max\left\{\sum_{j\in C} x_j + \sum_{j=1}^{i-1} \alpha_{k_j} x_{k_j}\right\},\\
&\sum_{j\in C} a_j x_j + \sum_{j=1}^{i-1} a_{k_j} x_{k_j} \leqslant b - a_{k_i},\\
&x_j \in \{0,1\} \quad \forall j \in C,\\
&x_{k_j} \in \{0,1\} \quad \forall j = 1,\ldots,i-1,
\end{aligned} \tag{1}$$

for $i = 1,\ldots,n-r$. LCI defines a facet if and only if equality holds for $i = 1,\ldots,n-r$.

Results concerning the complexity of obtaining lifted cover inequalities can be found in [4] and [8]. Calculating the α_{k_i} involves solving a set of knapsack problems for which an upper bound is necessary to find a valid inequality and an exact solution is necessary to find a facet-defining inequality. An upper bound is easily obtained by solving the linear programming relaxation of the knapsack problem in (1) and rounding up the optimal value. This is what is done in OSL [6]. It is fast, but may not give the largest value of α_{k_i}. Another approach is to use dynamic programming to solve the parametric problem

$$\begin{aligned}
\min\ &\sum_{j\in C} a_j x_j + \sum_{j=1}^{i-1} a_{k_j} x_{k_j},\\
&\sum_{j\in C} x_j + \sum_{j=1}^{i-1} \alpha_{k_j} x_{k_j} \geqslant t,\\
&x_j \in \{0,1\} \quad \forall j \in C,\\
&x_{k_j} \in \{0,1\} \quad \forall j = 1,\ldots,i-1
\end{aligned} \tag{2}$$

as a function of t to find the largest value of the objective function that does not exceed $b - a_{k_i}$. Solving (2) yields the largest value of α_{k_i} and it can be done in polynomial time since the α_{k_j} are nonnegative integers between 0 and $r - 1 < n$. This is done in MINTO [5].

Still another approach to obtaining the lifting coefficients is to use a theorem of Balas [1], as generalized in Balas and Zemel [2], that gives a formula for determining the largest value of some of the α_{k_i} exactly and the largest values of the remaining coefficients within one. Using this formula, we can set one of the undetermined coefficients to the higher value and the rest to the lower value. We are guaranteed to get a valid inequality which is close to defining a facet in the sense that if an unspecified set of coefficients were each increased by one, the inequality would define a facet.

Here we improve this result by giving a set of inequalities that specifies exactly which subsets of the undetermined lifting coefficients can be set to the higher value simultaneously. Our result suggests heuristics for finding orderings for lifting the members of $\bar{C}$ that are likely to generate most violated facet-defining inequalities. However, for the knapsack problem without GUB constraints, the result may not be of computational significance since exact lifting can be performed efficiently by using the Balas and Zemel result to fix some coefficients and then applying dynamic programming to determine the rest.

Our extension of the Balas and Zemel result also applies to knapsack problems with nonoverlapping GUB constraints. When there are GUB constraints in addition to a knapsack inequality, the lifting coefficients can be made larger since the optimization in (1) is further constrained.

In Section 2, we present our result for this more general case, the 0-1 knapsack problem with nonoverlapping GUB constraints. When GUB constraints are present, dynamic programming is not as efficient for determining the lifting coefficients so the result may be of more practical use. The corresponding result for the 0-1 knapsack without GUBs is presented as a corollary. In Section 3, we use the result to find LCIs for an example problem. In Section 4, we present a heuristic for finding violated LCIs.

2. Facets of the 0-1 knapsack polytope with nonoverlapping GUB constraints

Consider the constraints of a 0-1 knapsack problem with nonoverlapping GUBs given by

$$\begin{aligned}
&\sum_{j\in N} a_j x_j \leqslant b,\\
&\sum_{j\in S_l} x_j \leqslant 1 \quad \forall l \in L,\\
&x_j \in \{0,1\} \quad \forall j \in N,\\
&S_l \cap S_j = \emptyset \quad \forall j \text{ and } \forall l \neq j,\\
&\bigcup_{l\in L} S_l = N.
\end{aligned} \tag{3}$$

Define $C \subseteq N$ to be a *cover* if $\sum_{j\in C} a_j > b$ and each member of C is in a different GUB.

Given a minimal cover $C = \{j_1, j_2, \ldots, j_r\}$, $a_{j_1} \geqslant a_{j_2} \geqslant \cdots \geqslant a_{j_r}$, let $\bar{C} = N \setminus C$, $\mu_0 = 0$, $\mu_h = \sum_{i=1}^{h} a_{j_i}$ for $h = 1, \ldots, r$, $\mu_t = \mu_r$ for $t > r$, and $\lambda = \mu_r - b > 0$. For $k \in \bar{C}$, let $\beta_k = h$ if $\mu_h \leqslant a_k \leqslant \mu_{h+1} - 1$. For $Q \subseteq \bar{C}$, define $\beta(Q) = \sum_{i\in Q}(\beta_i + 1)$.

Let

$$C_1(Q) = \{j \in C : \exists\, k \in Q \text{ and } l \in L \text{ such that } j, k \in S_l\},$$

$$C_2(Q) = \{j \in C_1(Q) : a_j \leqslant a_{j_{\beta(Q)+1-|C_2(Q)|}}\}.$$

$C_1(Q)$ is the set of all members of C that are contained in a GUB that also contains an item in Q. $C_2(Q)$ is easily determined by considering the elements of $C_1(Q)$ in nondecreasing order of size a_j.

For the 0–1 knapsack problem without GUB constraints, the sets $C_1(Q)$ and $C_2(Q)$ are both empty. For this case, Balas and Zemel showed that $\alpha_j \leqslant \beta_j + 1$ for all $j \in \bar{C}$ is a necessary condition for LCI

$$\sum_{j\in C} x_j + \sum_{j\in\bar{C}} \alpha_j x_j \leqslant r - 1$$

to be valid, and $\alpha_j \geqslant \beta_j$ is a necessary condition for LCI to define a facet. Moreover, they showed that $\alpha_j \leqslant \beta_j$ is a necessary condition for validity if $a_j \leqslant \mu_{\beta_j+1} - \lambda$. This leaves open the question of whether $\alpha_j = \beta_j$ or $\beta_j + 1$ in a facet-defining inequality if $\mu_{\beta_j+1} - \lambda < a_j < \mu_{\beta_j+1}$. Balas and Zemel showed that for at least one such j, it is valid to set $\alpha_j = \beta_j + 1$. We call these *B–Z inequalities* since they are the strongest ones guaranteed by the Balas and Zemel results.

Given an ordering $k_1, k_2, \ldots, k_{n-r}$ of the members of $\bar{C}$, we can determine the lifting coefficients in LCIs that are valid for (3) by solving the sequence of problems

$$\begin{aligned}
&\alpha_{k_i} \leqslant r - 1 - \max\left\{\sum_{j\in C} x_j + \sum_{j=1}^{i-1} \alpha_{k_j} x_{k_j}\right\},\\
&\sum_{j\in C} a_j x_j + \sum_{j=1}^{i-1} a_{k_j} x_{k_j} \leqslant b - a_{k_i},\\
&\sum_{j\in S_l} x_j \leqslant 1 \quad \forall l \in L, \qquad\qquad (4)\\
&\sum_{j\in S_l} x_j \leqslant 0 \quad \text{if } x_{k_i} \in S_l,\\
&x_j \in \{0,1\} \quad \forall\ j \in C,\\
&x_{k_j} \in \{0,1\} \quad \text{for } j = 1,\ldots,i-1,
\end{aligned}$$

for $i = 1, \ldots, n - r$. As for the case without GUB constraints, the LCI defines a facet if and only if equality holds for $i = 1, \ldots, n - r$.

We call a set $S \subseteq \bar{C}$ an *independent* set if for all nonempty $Q \subseteq S$ such that each member of Q is in a different GUB

$$\sum_{i\in Q} a_i > \mu_{\beta(Q)-|C_2(Q)|} + \sum_{j\in C_2(Q)} a_j - \lambda. \qquad (5)$$

Otherwise, S is called a *dependent* set. We use this terminology because the sets S form an *independence system*. An independence system is characterized by two properties: (1) the empty set must be independent, and (2) any subset of an independent set must be independent.

In the special case when no GUB constraints are present, we say that $S \subseteq \bar{C}$ is an *independent set* if for all nonempty $Q \subseteq S$,

$$\sum_{i\in Q} a_i > \mu_{\beta(Q)} - \lambda. \qquad (6)$$

Note that the right hand side of (5) reduces to the right hand side of (6) when $C_2(Q) = \emptyset$.

To gain some insight into the significance of the independence condition, consider the case when $C_2(Q) = \emptyset$. If Q is independent, the quantity $b - \mu_{\beta(Q)} + \lambda$ is an upper bound on the amount of room left in the knapsack after all the members of Q are added. The largest possible value of the sum of the lifting coefficients on the members of Q is obtained by lifting the members of Q first. The sum equals $r-1$ minus the maximum number of members of C that can be added to this knapsack. Consider possible solutions to the lifting problem with $x_i = 1$ for all $i \in Q$. It is easy to see that a solution that maximizes $\sum_{j\in C} x_j$ can be constructed by adding members of C in nondecreasing order by weight until there is not enough room remaining for the next item. If the amount of room remaining in the knapsack after adding all the members of Q is strictly less than $b - \mu_{\beta(Q)} + \lambda$, then at most $r - \beta(Q) - 1$ members of C can be added since the sum of the weights of the smallest $r - \beta(Q)$ members of C is $\mu_r - \mu_{\beta(Q)} = b - \mu_{\beta(Q)} + \lambda$. Thus, at most $r - \beta(Q) - 1$ members of C will fit in the knapsack, and the sum of the lifting coefficients of the members of Q is less than or equal to $\beta(Q)$. If $\sum_{i\in Q} a_i \leqslant \mu_{\beta(Q)} - \lambda$, $r - \beta(Q)$ members of C will fit in the knapsack, and the sum of the lifting coefficients must be less than or equal to $\beta(Q) - 1$.

Theorem 1.

$$\sum_{j\in C} x_j + \sum_{j\in \bar{C}} \alpha_j x_j \leqslant r-1$$

defines a facet of the 0–1 knapsack problem with nonoverlapping GUB constraints if and only if $\alpha_j = \beta_j + 1$ *for all* j *in a maximal independent set* S *and* $\alpha_j = \beta_j$ *for all* $j \in \bar{C} \setminus S$.

Proof. Define

$$G_{C\setminus C_1(Q)}(v) = \max \sum_{j\in C\setminus C_1(Q)} x_j,$$

$$\sum_{j\in C\setminus C_1(Q)} a_j x_j \leqslant v$$

$$x_j \in \{0,1\} \quad \forall j \in C \setminus C_1(Q).$$

Since $C_1(Q)$ is the set of members of C that appear in a GUB constraint with a member of Q, $G_{C\setminus C_1(Q)}(v)$ is the maximum number of members j of C that can be placed in a knapsack of size v if all members of C that are in a GUB with some $k \in Q$ are forbidden. Note that

$$G_{C\setminus C_1(Q)}\left(b - \mu_{\beta(Q)-|C_2(Q)|} - \sum_{j\in C_2(Q)} a_j + \lambda\right) = r - \beta(Q). \tag{7}$$

To see why this is true, observe that

$$b - \mu_{\beta(Q)-|C_2(Q)|} - \sum_{j\in C_2(Q)} a_j + \lambda = \mu_r - \mu_{\beta(Q)-|C_2(Q)|} - \sum_{j\in C_2(Q)} a_j.$$

The members of $C_2(Q)$ are members of $C_1(Q)$ with small enough weights that they would have been used in the greedy solution to the knapsack if they were not forbidden. Thus, $\mu_r - \mu_{\beta(Q)-|C_2(Q)|} - \sum_{j\in C_2(Q)} a_j$ is equal to the sum of the weights of the $r - \beta(Q)$ smallest members of C that can be placed in a knapsack with the members of Q. Also note that

$$G_{C\setminus C_1(Q)}\left(b - \mu_{\beta(Q)-|C_2(Q)|} - \sum_{j\in C_2(Q)} a_j + \lambda - 1\right) = r - \beta(Q) - 1 \tag{8}$$

since $b - \mu_{\beta(Q)-|C_2(Q)|} - \sum_{j\in C_2(Q)} a_j + \lambda - 1$ is less than the sum of the weights of the $r - \beta(Q)$ smallest members of C that can be placed in a knapsack with the members of Q.

For the LCI to be valid, the inequality

$$\sum_{i\in Q}\alpha_i + G_{C\setminus C_1(Q)}\left(b-\sum_{i\in Q}a_i\right)\leqslant r-1$$

must hold if it is feasible for all $k\in Q$ to have $x_k=1$ simultaneously (no members of Q are contained in the same GUB). We will derive the bound implied by this inequality for S dependent and S independent.

Suppose $S\subseteq\bar{C}$ is dependent. Then by (5) there exists $Q\subseteq S$ with

$$\sum_{i\in Q}a_i\leqslant \mu_{\beta(Q)-|C_2(Q)|}+\sum_{j\in C_2(Q)}a_j-\lambda$$

and no two members of Q are members of the same S_l. Since

$$b-\sum_{i\in Q}a_i\geqslant b-\mu_{\beta(Q)-|C_2(Q)|}-\sum_{j\in C_2(Q)}a_j+\lambda,$$

we have

$$\begin{aligned}G_{C\setminus C_1(Q)}\left(b-\sum_{i\in Q}a_i\right)&\geqslant G_{C\setminus C_1(Q)}\left(b-\mu_{\beta(Q)-|C_2(Q)|}-\sum_{j\in C_2(Q)}a_j+\lambda\right)\\&=r-\beta(Q),\end{aligned}$$

where the equality follows from (7). So

$$\sum_{i\in Q}\alpha_i\leqslant r-1-G_{C\setminus C_1(Q)}\left(b-\sum_{i\in Q}a_i\right)\leqslant r-1-r+\beta(Q)=\beta(Q)-1.$$

Thus, at least one of the members of the set Q must have $\alpha_i=\beta_i$.

Suppose S is independent, then by (5) for all $Q\subseteq S$,

$$\sum_{i\in Q}a_i>\mu_{\beta(Q)-|C_2(Q)|}+\sum_{j\in C_2(Q)}a_j-\lambda$$

or, if this inequality is violated for some Q then at least two members of Q are contained in the same GUB. First, suppose no two members of Q are contained in the same GUB.

Since

$$b-\sum_{i\in Q}a_i\leqslant b-\mu_{\beta(Q)-|C_2(Q)|}-\sum_{j\in C_2(Q)}a_j+\lambda-1,$$

$$\begin{aligned}G_{C\setminus C_1(Q)}\left(b-\sum_{i\in Q}a_i\right)&\leqslant G_{C\setminus C_1(Q)}\left(b-\mu_{\beta(Q)-|C_2(Q)|}-\sum_{j\in C_2(Q)}a_j+\lambda-1\right)\\&=r-\beta(Q)-1,\end{aligned}$$

where the equality follows from (8). Let

$$\sum_{i\in Q}\alpha_i=\beta(Q).$$

Then

$$\sum_{i\in Q}\alpha_i + G_{C\setminus C_1(Q)}\left(b-\sum_{i\in Q}a_i\right) \leqslant \beta(Q)+(r-\beta(Q)-1)\leqslant r-1.$$

The result for Q independent with no two members of Q in the same GUB hinges on the observation that if all $k \in Q$ have $x_k = 1$ it is not possible to add enough members of C to the knapsack to violate $\sum_{j\in C} x_j + \sum_{k\in Q}\alpha_k x_k \leqslant r-1$ even if $\alpha_k = \beta_k + 1$ for all $k \in Q$. If the independence of Q results from two or more members of Q being contained in the same GUB rather than (5), then it is infeasible for all $k \in Q$ to have $x_k = 1$ simultaneously due to the GUB constraints. Therefore, the bound on the lifting coefficients of the members of Q is dictated by the weights of subsets of Q with no two members in the same GUB. Since Q is independent, all such subsets satisfy (5) and it is therefore possible to allow $\alpha_k = \beta_k + 1$ for all $k \in Q$. Thus, all the members of S may have $\alpha_i = \beta_i + 1$. □

The analogous result for the 0–1 knapsack problem without GUB constraints follows immediately.

Corollary 2.

$$\sum_{j\in C} x_j + \sum_{j\in\bar{C}}\alpha_j x_j \leqslant r-1$$

defines a facet of the 0–1 knapsack polytope if and only if $\alpha_j = \beta_j + 1$ *for all* j *in a maximal independent set* S *and* $\alpha_j = \beta_j$ *for all* $j \in \bar{C}\setminus S$.

3. An Example

We illustrate the application of Theorem 1 and Corollary 2 with an example. First we consider a 0–1 knapsack problem with no additional GUB constraints.

Example 1

$$37x_1 + 25x_2 + 23x_3 + 15x_4 + 14x_5 + 12x_6 + 11x_7 + 8x_8 + 7x_9 + 3x_{10} \leqslant 39,$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10} \in \{0,1\}.$$

Let $C = \{4,5,8,9\}$. Then $\mu_0 = 0$, $\mu_1 = 15$, $\mu_2 = 29$, $\mu_3 = 37$, $\mu_4 = 44$, and $\lambda = 5$. First, we will identify all independent sets S with $|S| = 1$. There are three such sets, $\{2\},\{6\}$, and $\{7\}$, with $\beta(\{2\}) = 2$, and $\beta(\{6\}) = \beta(\{7\}) = 1$. The lifting coefficients of all other members of $\bar{C}$ are completely determined. Thus, we obtain three valid B–Z LCIs by setting $\alpha_j = \beta_j + 1$ for $j = 2, 6$, or 7 and letting all other variables have $\alpha_k = \beta_k$.

To check whether two of these variables may have $\alpha_j = \beta_j + 1$ simultaneously, we check the independence condition on each pair. For the set $\{2,6\}$, we have $\beta(\{2,6\}) = 3$, $a_2 + a_6 = 37$, and $\mu_3 - \lambda = 32$. Thus, the set $\{2,6\}$ is independent. Similarly, the set $\{2,7\}$ is independent. For the set $\{6,7\}$, we have $\beta(\{6,7\}) = 2$, $a_6 + a_7 = 23$, and $\mu_2 - \lambda = 24$. Thus, the set $\{6,7\}$ is dependent. The two maximal independent sets $\{2,6\}$ and $\{2,7\}$ give rise to the two facet defining inequalities

$$3x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 + x_8 + x_9 \leqslant 3,$$

$$3x_1 + 2x_2 + x_3 + x_4 + x_5 + x_7 + x_8 + x_9 \leqslant 3,$$

which clearly dominate the three B–Z inequalities.

Now suppose the GUB constraints

$$x_6 + x_7 \leqslant 1, \qquad x_9 + x_{10} \leqslant 1$$

are added to the above knapsack problem. All other variables are in GUB sets S_l of size one. As before, the sets $\{2\}, \{6\}, \{7\}, \{2,6\}$, and $\{2,7\}$ are independent. Since x_6 and x_7 appear in the same GUB constraint, the sets $\{6,7\}$ and $\{2,6,7\}$ are also independent. The only other variable that may have a change in its lifting coefficient is x_{10} since it is the only other member of $\bar{C}$ appearing in a nontrival GUB constraint. Checking the independence of $\{10\}$, we find $\beta(\{10\}) = 1$ and $C_1(\{10\}) = C_2(\{10\}) = \{9\}$ since $a_9 \leqslant a_{j_1}$. Checking the independence condition (5), yields $\mu_0 + a_9 - \lambda = 2$. Since $a_{10} = 3 > 2$ we conclude that $\{10\}$ is independent. To see if $\{10\}$ is a maximal independent set, we will try to combine it with each of the singleton independent sets. For $\{2,10\}$, we have $a_2 + a_{10} = 28$, $\beta(\{2,10\}) = 3$, $C_1(\{2,10\}) = C_2(\{2,10\}) = \{9\}$, $\mu_2 + a_9 - \lambda = 31$, so the set $\{2,10\}$ is dependent. Similarly, we conclude that $\{6,10\}$ and $\{7,10\}$ are dependent. Therefore, we have two maximal independent sets, $\{2,6,7\}$ and $\{10\}$ which provide the two facet-defining LCIs

$$3x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 \leqslant 3,$$
$$3x_1 + x_2 + x_3 + x_4 + x_5 + x_8 + x_9 + x_{10} \leqslant 3.$$

These inequalities are stronger than the LCIs obtained if the GUB constraints are ignored.

4. Selecting LCIs

Given a fractional solution x^* and a cover C, if we set all the lifting coefficients equal to β_j the violation of the LCI is $\sum_{j\in C} x_j^* + \sum_{j\in\bar{C}} \beta_j x_j^* - |C| + 1$. Identifying an independent set S and setting $\alpha_j = \beta_j + 1$ for all $j \in S$ we increase the violation by $\sum_{j\in S} x_j^*$. Thus our objective is to find a maximum weight independent set S with respect to the weights x_j^*. It can be shown that this problem is NP-Hard by reduction from 0–1 knapsack [3].

We propose the following heuristic to solve the maximum weight independent set problem.

Step 1. Initialize S to be empty. Construct a graph $G = (V, E)$ with $V = \{j \in \bar{C} : \alpha_j \text{ is not fixed }\}$ and $E = \{(i,j) : i,j \in V \text{ and } \{i,j\} \text{ is a dependent set}\}$. Let $N(j)$ be the set of all nodes that are adjacent to j.

Step 2. Choose a $j \in V$ with maximum $x_j^* - \sum_{k\in N(j)} x_k^*$.

Step 3. If $|S| \leqslant 1$, go to Step 4. If $|S| > 1$, for each $Q \subset S$ with $|Q| > 1$ check condition (5) for $Q \cup \{j\}$. Once a dependent set is found, stop checking subsets and delete j from G leaving S unchanged; go to Step 5. If all of these sets satisfy (5), go to Step 4.

Step 4. Add j to S and delete j and all the members of $N(j)$ from G.

Step 5. If G is empty, stop. Otherwise, go to Step 2.

Example 2

Consider the example with GUB constraints given in the previous section. Suppose we are given the fractional solution $x_1^* = 0$, $x_2^* = 0.5$, $x_3^* = 0$, $x_4^* = 0.5$, $x_5^* = 0$, $x_6^* = x_7^* = x_8^* = x_9^* = 0.5$, $x_{10}^* = 0$, and the cover $C = \{4,5,7,8\}$. We wish to generate a violated lifted cover inequality using this cover. From our previous work on the example, we see that G has four nodes, one for each of the items 2, 6, 7, and 10. Adding an edge between each pair of items that are pairwise dependent and placing the weights x^* on the nodes gives G as shown in Fig. 1.

The heuristic identifies the independent set $\{2,6,7\}$ with weight 1.5 and the inequality

$$3x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 \leqslant 3,$$

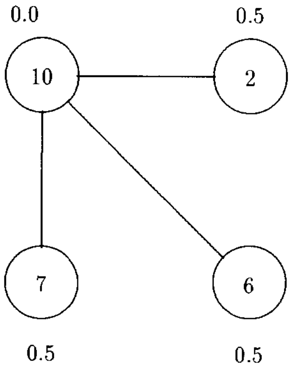

Fig. 1. Pairwise independence graph.

which is violated by x^*. It is important to note that this fractional solution does not violate any lifted cover inequality that can be derived from C without considering the GUB constraints.

References

[1] E. Balas, "Facets of the knapsack polytope", *Math. Prog.* **8**, 146-164 (1975).
[2] E. Balas and E. Zemel, "Facets of the knapsack polytope from minimal covers", *SIAM J. Appl. Math.* **34**, 119-148 (1978).
[3] Z. Gu, G.L. Nemhauser and M.N.P. Savelsberg, "Lifted cover inequalities for 0-1 integer programs II: Complexity", Report 94-10, Computational Optimization Center, Georgia Institute of Technology, 1994.
[4] D. Hartvigsen and E. Zemel, "The complexity of lifted inequalities for the knapsack problem", *Discr. Appl. Math.* **39**, 113-123 (1992).
[5] G.L. Nemhauser, M.W.P. Savelsbergh and G.C. Sigismondi, "MINTO, a Mixed INTeger Optimizer", *Oper. Res. Lett.* **15**, 47-58 (1994).
[6] IBM Corporation, *Optimization Subroutine Library, Guide and Reference*, 1990.
[7] L.A. Wolsey, "Valid inequalities for mixed integer programs with generalised and variable upper bound constraints", *Discr. Appl. Math.* **25**, 251-261 (1990).
[8] E. Zemel, "Easily computable facets of the knapsack polytope", *Math. Oper. Res.* **14**, 760-764 (1989).