Computational Optimization and Applications 8, 151–172 (1997)

Generating Cuts from Surrogate Constraint Analysis for Zero-One and Multiple Choice Programming

FRED GLOVER
Graduate School of Business, CB 419, University of Colorado, Boulder, CO 80309

HANIF D. SHERALI
Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0118

YOUNGHO LEE
US WEST Advanced Technologies, Applied Research, 4001 Discovery Drive, Boulder, Colorado 80303

Received May 19, 1995; Revised March 28, 1996; Accepted April 8, 1996

Abstract. This paper presents a new surrogate constraint analysis that gives rise to a family of strong valid inequalities called surrogate-knapsack (S-K) cuts. The analytical procedure presented provides a strong S-K cut subject to constraining the values of selected cut coefficients, including the right-hand side. Our approach is applicable to both zero-one integer problems and problems having multiple choice (generalized upper bound) constraints. We also develop a strengthening process that further tightens the S-K cut obtained via the surrogate analysis. Building on this, we develop a polynomial-time separation procedure that successfully generates an S-K cut that renders a given non-integer extreme point infeasible. We show how sequential lifting processes can be viewed in our framework, and demonstrate that our approach can obtain facets that are not available to standard lifting methods. We also provide a related analysis for generating "fast cuts". Finally, we present computational results of the new S-K cuts for solving 0-1 integer programming problems. Our outcomes disclose that the new cuts are capable of reducing the duality gap between optimal continuous and integer feasible solutions more effectively than standard lifted cover inequalities, as used in modern codes such as the CPLEX mixed 0-1 integer programming solver.

Keywords: surrogate-knapsack cuts, fractional surrogate constraint cuts, Chvátal Gomory cuts, knapsack polytope, separation procedure, liftings

1. Introduction

Consider the zero-one knapsack polytope KP defined as follows:

$$\mathrm{KP} \equiv Conv\left\{x \in (0,1)^n : \sum_{j\in N} a_j x_j \le a_0\right\}, \tag{1}$$

where $N = \{1, \ldots, n\}$ and $0 < a_j \le a_0\ \forall j \in N$.

The knapsack polytope has been extensively investigated for the last 30 years. The main focus of interest has been to study the polyhedral structure of the polytope (for example, see [1, 2, 18, 24, 26], to name a few). Another research direction in regard to the knapsack polytope has been in the context of generating tight relaxations via the process of surrogating the original constraints into a single knapsack constraint. The works of Glover [9, 11], Karwan and Rardin [21], and Greenberg and Pierskalla [16, 17] are examples of this approach. As demonstrated in the results of Crowder et al. [6] and Hoffman and Padberg [20], even a partial knowledge of the polyhedral structure of the knapsack polytope associated with individual problem constraints can significantly enhance the overall performance of branch-and-cut algorithms. This is particularly true if the problem is sparse, a property exhibited by many real world zero-one integer programming problems. Consequently, most mixed integer programming solvers implement several cut generation features based on the polyhedral structure of the knapsack polytope (for example, see [5]).

In this paper, we present a new cut generation scheme based on a surrogate constraint analysis that creates an appropriate nonnegative linear combination of the knapsack constraint with bounding inequalities of the form $x_j \leq 1$, $\forall j \in N$, and then derives a special class of the Gomory all-integer cuts [14]. By coordinating the determination of surrogate constraint parameters with the associated cutting plane coefficients, we extend the work of Glover [10, 11] which shows how to generate coordinated parameters for combining a cutting plane with its source constraint, to yield stronger "fractional cuts" for eliminating non-integer linear programming vertices. Our analysis also includes strengthened cuts in relation to the multiple choice (generalized upper bound) knapsack polytope GUBKP defined by

$$\text{GUBKP} \equiv Conv\left\{x \in (0,1)^n : \sum_{i \in M} \sum_{j \in N_i} a_j x_j \leq a_0, \sum_{j \in N_i} x_j \leq 1 \ \forall i \in M\right\}, \tag{2}$$

where $0 < a_j \leq a_0$ $\forall j \in N$, $M = \{1, \ldots, m\}$, and where $\bigcup_{i \in M} N_i = N \equiv \{1, \ldots, n\}$, with $N_i \cap N_j = \phi$ for $i, j \in M, i \neq j$.

This paper is organized as follows. In Section 2 below, we present a new family of strong valid inequalities, called surrogate-knapsack (S-K) cuts, based on a surrogate constraint analysis. This analytical procedure generates a strong surrogate-knapsack cut satisfying prescribed bounds on the coefficients of the cut. We also develop a strengthening process to further tighten this cut. Subsequently, in Section 3, we develop a separation procedure for identifying a violated surrogate-knapsack cut that deletes a given fractional linear programming solution. Moreover, we show that this separation procedure can be implemented in polynomial-time. In Section 4, we present another strengthening procedure to derive a particular class of the surrogate-knapsack cut, and exhibit how the standard sequential lifting process can be viewed in this framework. We also show that our analysis yields facets that cannot be obtained by customary lifting approaches. Section 5 provides a related analysis for generating "fast cuts". Section 6 presents some preliminary computational results on the new S-K cut for solving 0-1 integer programming problems. Section 7 concludes the paper.

2. Surrogate-knapsack cuts

We begin by considering the knapsack constraint given in (1), and later show how our results extend to the GUB knapsack polytope of (2). For a given set $J \subseteq N$, we surrogate the bounding constraints $x_j \leq 1,\ j \in J$ with the knapsack constraint $\sum_{j\in N} a_j x_j \leq a_0$ using nonnegative surrogate multipliers $u_j,\ j \in J$, and u_0, respectively. By summing and rounding down the coefficients of this surrogate constraint, we can generate a "surrogate-knapsack cut" as an instance of those characterized in [9], where in this case, the cut derives from the formulation of Gomory [14]. (These classes of cuts are also referred to as Chvátal-Gomory cuts, in recognition of an exploration of their properties by Chvátal [4]—see, for example, [23].) The resulting cut is of the form

$$\sum_{j\in J} \lfloor u_0 a_j + u_j \rfloor x_j + \sum_{j\in \bar{J}} \lfloor u_0 a_j \rfloor x_j \leq \left\lfloor u_0 a_0 + \sum_{j\in J} u_j \right\rfloor, \tag{3}$$

where $\bar{J} \equiv N - J$ and $\lfloor\cdot\rfloor$ denotes the rounded-down value. Dietrich and Escudero [8] show that some of the well known cover, clique and coefficient reduction cuts can be derived as certain simple instances of this type of cut. We will, however, be concerned with the development of a surrogate constraint analysis that generates a new class of strong cuts having somewhat broader properties. Moreover, this surrogate constraint analysis will enable us to prescribe a separation scheme for identifying a cut of the type (3) that deletes a given fractional linear programming solution, and for developing new types of facets.

To begin, suppose that we are interested in constructing a cut of the type (3) that has coefficients greater than or equal to some prescribed integers $b_j \geq 1, \forall j \in J$, while having as small a right-hand side as possible. Hence, we would like to solve the following "surrogate constraint" problem SC.

$$\begin{aligned} \text{SC: Minimize} \quad & u_0 a_0 + \sum_{j\in J} u_j \\ \text{subject to} \quad & u_0 a_j + u_j \geq b_j \quad \forall j \in J \\ & u_0 \geq 0, u_j \geq 0 \quad \forall j \in J. \end{aligned} \tag{4}$$

We can project Problem SC onto the space of the variable u_0 by noting that given u_0, we can construct an optimal solution by selecting

$$u_j = \max\{0, b_j - u_0 a_j\} \quad \forall j \in J.$$

The dual to SC is given by the following bounded variable knapsack problem.

$$\begin{aligned} \text{DSC: Maximize} \quad & \sum_{j\in J} b_j x_j \\ \text{subject to} \quad & \sum_{j\in J} a_j x_j \leq a_0 \\ & 0 \leq x_j \leq 1 \quad \forall j \in J. \end{aligned}$$

Now, let us assume that

$$\sum_{j\in J} a_j > a_0 \tag{5}$$

or else $u_0 = 0$ solves SC yielding an inconsequential cut (3). Then, assuming for convenience that

$$J \equiv \{1, \ldots, r\} \quad \text{and that} \quad \frac{b_1}{a_1} \leq \frac{b_2}{a_2} \leq \cdots \leq \frac{b_r}{a_r}, \tag{6}$$

let us find the largest index p, $1 \leq p < r$, such that

$$a_0 - \sum_{j=p}^{r} a_j < 0. \tag{7}$$

It follows then that an optimal basic feasible solution to DSC has basis $B = \{a_p\}$, yielding an optimal dual solution $u_0^* = b_p/a_p$. Hence, from SC, we then have $u_j^* = \max\{0, b_j - u_0 a_j\}$ for all $j \in J$. The optimum solution to Problem SC is therefore given by

$$u_0^* = \frac{b_p}{a_p}, \quad u_j^* = b_j - \frac{b_p}{a_p} a_j \quad \forall j \in J^*, \quad \text{and} \quad u_j^* = 0 \quad \forall j \in J - J^*, \tag{8}$$

where $J^* = \{j \in J : \frac{b_j}{a_j} \geq \frac{b_p}{a_p}\}$.

Consequently, defining $\bar{J}^* = N - J^*$, the cut (3) would be given by

$$\sum_{j \in J^*} b_j x_j + \sum_{j \in \bar{J}^*} \lfloor u_0^* a_j \rfloor x_j \leq \left\lfloor u_0^* \left(a_0 - \sum_{j \in J^*} a_j \right) + \sum_{j \in J^*} b_j \right\rfloor \equiv b_0^*. \tag{9}$$

Now, let us strengthen (9) further by manipulating u_0^* to $u_0^{**} \geq u_0^*$ such that the right-hand side in (3) remains the same as that derived in (9), while possibly improving (increasing) the coefficients on the left-hand side of (3). Denote the right-hand side in (9) as b_0^*, and choose $0 < \epsilon < 1$, as a small positive quantity (to be prescribed later). We now wish to find the surrogate coefficients in (3) by solving the following problem.

$$\begin{aligned} \text{SCR: Maximize} \quad & u_0 \\ \text{subject to} \quad & u_0 a_0 + \sum_{j \in J} u_j = b_0^* + 1 - \epsilon \\ & u_0 a_j + u_j \geq b_j, \quad \forall j \in J \\ & u_0 \geq 0, u_j \geq 0 \quad \forall j \in J. \end{aligned}$$

Remark 1. If the expression in $\lfloor \cdot \rfloor$ on the right-hand side of (9) equals $b_0^* + f_0$, where $0 \leq f_0 < 1$, then by selecting $b_0^* + 1 - \epsilon \geq b_0^* + f_0$, i.e., $0 < \epsilon \leq 1 - f_0$, we are assured that the optimum value in Problem SCR will be at least u_0^*. We therefore prescribe $\epsilon = \min\{0.01, 1 - f_0\}$.

Problem SCR can be restated as follows.

$$\begin{aligned} \text{SCR: Maximize} \quad & u_0 \\ \text{subject to} \quad & u_0 a_0 + \sum_{j \in J} u_j = (b_0^* + 1 - \epsilon) \equiv b_0^{**} && (10) \\ & u_0 \geq \frac{b_j}{a_j} - \frac{u_j}{a_j}, \quad \forall j \in J && (11) \\ & u_0 \geq 0, \; u_j \geq 0 \quad \forall j \in J. \end{aligned}$$

Note that an upper bound on Problem SCR is

$$u_0 = \frac{b_0^{**}}{a_0} \text{ which is feasible iff } u_j = 0 \quad \forall j \in J \text{ is feasible, i.e., } \frac{b_0^{**}}{a_0} \geq \frac{b_j}{a_j} \; \forall j \in J. \tag{12}$$

If (12) is infeasible, then the optimum value is less than b_0^{**}/a_0, and at an extreme point optimum to the linear program SCR, it is easy to see that for each $j \in J$, either $u_j = 0$ or the corresponding constraint in (11) is binding. Moreover, noting (6), we have that at optimality, there exists a $J^{**} = \{q, \ldots, r\} \subseteq J$ such that

$$u_0 = \frac{b_j}{a_j} - \frac{u_j}{a_j} \quad \forall j \in J^{**}, \quad \text{and} \quad u_0 \geq \frac{b_j}{a_j} \quad \forall j \in J - J^{**}. \tag{13}$$

Equation (10) would then determine the optimum $u_0 \equiv u_0^{**}$ via $a_0u_0 + \sum_{j \in J^{**}} (b_j - u_0 a_j) = b_0^{**}$, as

$$u_0^{**} = \frac{b_0^{**} - \sum_{j \in J^{**}} b_j}{a_0 - \sum_{j \in J^{**}} a_j}. \tag{14}$$

Therefore, from (13), we would get the corresponding optimal values of u_j as

$$u_j^{**} = \begin{cases} b_j - a_j u_0^{**} & \text{for } j \in J^{**} \\ 0 & \text{for } j \in J - J^{**} \end{cases} \tag{15}$$

where J^{**} is determined via the largest index $1 \leq q \leq r$ for which u_0^{**} of (14) satisfies (13), or equivalently, either $q = 1$, or

$$u_0^{**} \geq \frac{b_{q-1}}{a_{q-1}} \quad \text{if } q \geq 2. \tag{16}$$

Hence, (14) and (15) would then determine the desired cut as

$$\sum_{j \in J^{**}} b_j x_j + \sum_{j \in \overline{J^{**}}} \lfloor u_0^{**} a_j \rfloor x_j \leq b_0^{*}, \tag{17}$$

where $\overline{J^{**}} \equiv N - J^{**}$.

Example 1. Consider the following knapsack polytope.

$$\text{KP} \equiv Conv\{x \in (0, 1)^4 : 25x_1 + 11x_2 + 11x_3 + 10x_4 \leq 44\}$$

with $J = \{1, 2, 3, 4\}$, and $b_j = 1$, $\forall j \in J$, so that (5) and (6) hold. From (7), we first find that $p = 1$, and so from (8), we get $u_0^* = 1/25$, $J^* = \{1, 2, 3, 4\}$ with $u_1^* = 0$, $u_2^* = 14/25$, $u_3^* = 14/25$, and $u_4^* = 15/25$, which yields $b_0^* = \lfloor -13/25 + 4 \rfloor = 3$ as the right-hand side of (9). Hence, the cut (9) is given by

$$x_1 + x_2 + x_3 + x_4 \leq 3. \tag{18}$$

To strengthen this to the form (17), we first test (12). Here, if $u_0 \equiv b_0^{**}/a_0 = (b_0^*+1-\epsilon)/a_0 = 3.99/44$, where $\epsilon = 0.01$, we get $u_0 < b_4/a_4 = 0.1$. Hence, we apply (14), (15) and (16) to obtain the following trials.

(i) $J^{**} = \{4\}$ and $u_0^{**} = 2.99/34 = 0.0879 < 1/11$; hence, (16) is violated.
(ii) $J^{**} = \{3, 4\}$ and $u_0^{**} = 1.99/23 = 0.0865 < 1/11$; hence, (16) is violated.
(iii) $J^{**} = \{2, 3, 4\}$ and $u_0^{**} = 0.99/12 = 0.0825 \geq 1/25$; hence, (16) holds true.

So, at optimality, $u_0^{**} = 0.0825$, and the set $J^{**} = \{2, 3, 4\}$, $\overline{J^{**}} = \{1\}$. This produces the cut (17) as $2x_1 + x_2 + x_3 + x_4 \leq 3$, noting that $\lfloor 0.0825(25) \rfloor = 2$.

Remark 2 (Extension to the GUB knapsack polytope). We now see how our preceding analysis also applies to the GUB knapsack problem defined in (2). Here, we choose the set $J \subseteq N$ to contain the index of at most a single variable from each GUB set N_i for $i \in M$. That is, $|N_i \cap J| \leq 1$, $\forall i \in M$. Letting $M_J = \{i \in M : |N_i \cap J| = 1\}$, we apply the surrogate multiplier u_i, for each $i \in M_J$, to the corresponding GUB inequality $\sum_{j \in N_i} x_j \leq 1$. Then, it is readily shown that our analysis goes through as before, with suitable refinement of notation, except that we now strengthen additional cut coefficients.

3. A separation procedure for the surrogate-knapsack cuts

Now, suppose that we are given a solution $\hat{x}$ to the linear programming relaxation that happens to be fractional, and that we are interested in generating, if possible, a (tight) constraint of the type (3) that deletes $\hat{x}$. Preferably, we would like this valid inequality to be a facet of KP. Toward this end, let $J = \{j \in N : \hat{x}_j > 0\}$ (or some suitable subset thereof), and let us first try to generate a valid inequality of type (3) that has coefficients greater than or equal to 1 for all $j \in J$, and that has the smallest right-hand side value. Hence, by Eq. (18), we can select surrogate multipliers as

$$u_0^* = \frac{1}{a_p}, \quad u_j^* = 1 - \frac{a_j}{a_p} \quad \forall j \in J^*, \quad \text{and} \quad u_j^* = 0 \quad \forall j \in \overline{J^*},$$

where $J^* \equiv \{p, \ldots, r\} \subseteq J$ is defined by selecting p via (7). This yields a surrogate-knapsack cut of the type (3) or (9) as follows, where we have used (7) to derive the right-hand side.

$$\sum_{j \in J^*} x_j + \sum_{j \in \overline{J^*}} \left\lfloor \frac{a_j}{a_p} \right\rfloor x_j \leq |J^*| - 1. \tag{19}$$

Note from (7) that $\sum_{j \in J^*} x_j \leq |J^*| - 1$ is a valid cover inequality and so, (19) is a strengthening thereof.

Now, we can formulate a separation problem to generate a cut of the type (3) whose right-hand side is less than or equal to $|J^*| - 1$ as in (19), but which maximizes the left hand side of (3), so as to possibly delete $\hat{x}$. This can be stated as follows.

$$\text{SEP0: Maximize} \quad \sum_{j\in J} \lfloor u_0 a_j + u_j \rfloor \hat{x}_j \tag{20}$$
$$\text{subject to} \quad u_0 a_0 + \sum_{j\in J} u_j = |J^*| - \epsilon$$
$$u_0 \geq 0, \quad u_j \geq 0 \quad \forall j \in J,$$

where $0 < \epsilon < 1$ is to be specified later, but can now be viewed as simply requiring the right-hand side of (3) to be given by that in (19). Since $\lfloor u_0 a_j + u_j \rfloor \; \forall j \in J$ are integers, problem SEP0 can be rewritten as

$$\text{SEP0: Maximize} \quad \sum_{j\in J} y_j \hat{x}_j$$
$$\text{subject to} \quad u_0 a_0 + \sum_{j\in J} u_j = |J^*| - \epsilon$$
$$u_0 a_j + u_j \geq y_j \quad \forall j \in J$$
$$u_0 \geq 0, \quad u_j \geq 0, \quad y_j \text{ integer} \quad \forall j \in J.$$

Ideally, we would like to be able to solve SEP0 efficiently (in polynomial-time). Barring this possibility, we instead consider the following related problem. This problem is equivalent to including $y_j \geq 1 \; \forall j \in J$ and relaxing integrality on y in the foregoing restatement of SEP0.

$$\text{SEP1: Maximize} \quad \sum_{j\in J} (u_0 a_j + u_j)\hat{x}_j \equiv \nu_0 u_0 + \sum_{j\in J} u_j \hat{x}_j \tag{21}$$
$$\text{subject to} \quad u_0 a_0 + \sum_{j\in J} u_j = |J^*| - \epsilon$$
$$u_0 a_j + u_j \geq 1 \quad \forall j \in J \tag{22}$$
$$u_0 \geq 0, \quad u_j \geq 0 \quad \forall j \in J,$$

where $\nu_0 \equiv \sum_{j\in J} a_j \hat{x}_j$.

Note that besides removing the rounding operators from (20), we have also added the constraints (22) in SEP1. Without (22), the solution to SEP1 would simply put $u_0 = 0$ and $u_j = 0, \; \forall j \in J - j_1$, where $j_1 = \arg\max\{\hat{x}_j : j \in J\}$ if $\hat{x}_{j_1} \geq \nu_0/a_0$ (as would be the case if any $\hat{x}_j = 1$, for example) and lead to the trivial facet constraint $x_{j_1} \leq 1$. Also, note that the linear program SEP1 is specially structured and admits an efficient solution scheme. Toward this end, consider a projection of problem SEP1 onto the space of the single variable u_0. This yields,

$$\max_{u_0 \geq 0} \; \nu_0 u_0 + \max \quad \sum_{j\in J} u_j \hat{x}_j$$
$$\text{subject to} \quad \sum_{j\in J} u_j = |J^*| - \epsilon - u_0 a_0 \tag{23}$$
$$-u_j \leq u_0 a_j - 1 \quad \forall j \in J \tag{24}$$
$$u_j \geq 0 \quad \forall j \in J.$$

Defining π_0 and π_j, $j \in J$, as the dual variables associated with (23) and (24), respectively, this problem becomes

$$\max_{u_0 \geq 0} \left[v_0 u_0 + \min_{\pi_0, \pi} \left\{ (|J^*| - \epsilon - u_0 a_0)\pi_0 + \sum_{j \in J} (u_0 a_j - 1)\pi_j : 0 \leq \pi_j \leq \pi_0 - \hat{x}_j \; \forall j \in J \right\} \right]. \tag{25}$$

Let $\hat{x}_{j_1} = \max\{\hat{x}_j : j \in J\}$. Then, noting the structure of the inner minimization problem in (25), we can restate (25) as follows:

$$\max_{u_0 \geq 0} \left[v_0 u_0 + \min_{\pi_0 \geq \hat{x}_{j_1}} \left\{ (|J^*| - \epsilon - u_0 a_0)\pi_0 + \sum_{j \in J} (\pi_0 - \hat{x}_j) \min\{u_0 a_j - 1, 0\} \right\} \right]. \tag{26}$$

Observe that for the inner minimization problem over π_0 to be bounded in (26), the coefficient of π_0 must be nonnegative, i.e., we must have

$$\phi(u_0) \equiv (|J^*| - \epsilon - u_0 a_0) + \sum_{j \in J} \min\{u_0 a_j - 1, 0\} \geq 0. \tag{27}$$

Then, whenever (27) holds true, the inner minimization problem in (26) is solved by $\pi_0 = \hat{x}_{j_1}$. Hence, (26) is equivalent to the following problem.

$$\begin{aligned} &\text{Maximize} \quad f(u_0) \equiv \left[v_0 u_0 + (|J^*| - \epsilon - u_0 a_0)\hat{x}_{j_1} + \sum_{j \in J} (\hat{x}_{j_1} - \hat{x}_j) \min\{u_0 a_j - 1, 0\} \right] \\ &\text{subject to} \quad \phi(u_0) \geq 0, \quad u_0 \geq 0. \end{aligned} \tag{28}$$

We now solve (28) to recover an optimum to SEP1, noting that this problem seeks the maximum of a piecewise linear concave univariate function over a bounded interval $[u_{01}, u_{02}]$ defined by (27) and $u_0 \geq 0$. In the algorithm given below, we first find u_{02} using Newton's method to determine the roots of (27). Note that at any given value of u_0, defining

$$J^+(u_0) = \left\{ j \in J : \frac{1}{a_j} > u_0 \right\} \quad \text{and} \quad J^-(u_0) = \left\{ j \in J : \frac{1}{a_j} \geq u_0 \right\}, \tag{29}$$

we have that the right-hand and left-hand derivatives $f^+(u_0)$ and $f^-(u_0)$, respectively, at u_0 are given by

$$f^+(u_0) = v_0 - a_0\hat{x}_{j_1} + \sum_{j \in J^+(u_0)} \theta_j \quad \text{and} \quad f^-(u_0) = v_0 - a_0\hat{x}_{j_1} + \sum_{j \in J^-(u_0)} \theta_j, \tag{30}$$

where $\theta_j \equiv (\hat{x}_j - \hat{x}_j)a_j \; \forall j \in J$.

Hence, if $f^-(u_{02}) \geq 0$, then we have $u_0^* = u_{02}$ at optimality in (28). Else, $u_0^* < u_{02}$. We next find the distinct values $v_1 < v_2 < \cdots < v_t$ taken by $\frac{1}{a_j}$, $j \in J$, that are less than u_{02}, and determine the largest index $1 \leq k \leq t$ for which $f^-(v_k) \geq 0$. (This is easily accomplished via (29) and (30).) If either k is nonexistent or $\phi(v_k)$ found from (27) is negative, then $u_0^* = u_{01}$. Otherwise, $u_0^* = v_k$. In case we conclude that $u_0^* = u_{01}$, this value is computed similar to u_{02} by using Newton's method to find the roots of (27). The procedure is stated formally below.

Algorithm to solve (28)

Step 1 (Find u_{02} using Newton's method).

(a) Let $u_0 = (|J^*| - \epsilon)/a_0$.

(b) If $\phi(u_0) \geq 0$, let $u_{02} = u_0$ and proceed to Step 1(c). Else, compute the left-hand derivative δ of ϕ at u_0 as $\delta = -a_0 + \sum_{j \in J^-(u_0)} a_j$. Update $u_0 \leftarrow u_0 - \phi(u_0)/\delta$ and repeat Step 1(b).

(c) If $f^-(u_{02}) \geq 0$, go to Step 4. Else, proceed to Step 2.

Step 2 (Determine the best breakpoint of (28)). Let $v_1 < v_2 < \cdots < v_t$ be the distinct values taken by $1/a_j$ for $j \in J$ that are strictly less than u_{02}. Find the largest index k in $\{1, \ldots, t\}$ for which $f^-(v_k) \geq 0$ via (29). If k is nonexistent, proceed to Step 3. Else, determine $\phi(v_k)$. If $\phi(v_k) < 0$, then proceed to Step 3, and otherwise, put $u_0 = v_k$ and go to Step 4.

Step 3 (Find u_{01} using Newton's method)

(a) Put $u_0 = 0$.

(b) If $\phi(u_0) \geq 0$, let $u_{01} = u_0$ and go to Step 4. Else, compute the right-hand derivative δ of ϕ at u_0 as $\delta = -a_0 + \sum_{j \in J^+(u_0)} a_j$. Update $u_0 \leftarrow u_0 - \phi(u_0)/\delta$ and repeat Step 3(b).

Step 4 (Optimum for SEP1). Put $u_0^* = u_0$, $u_j^* = 1 - u_0^* a_j$ $\forall j \in J^+(u_0^*) - j_1$, $u_j^* = 0$ $\forall j \in J - J^+(u_0^*) - j_1$, and $u_{j_1}^* = |J^*| - \epsilon - u_0^* a_0 - \sum_{j \in J^+(u_0^*) - j_1} u_j^* \equiv \phi(u_0^*)$. Use (u_0^*, u^*) in (3) to derive the required cut.

Remark 3. Note that $u_0 = \frac{1}{a_p}$ yields a feasible solution for Problem SEP1, and hence, it satisfies (27). Therefore, Steps 1 and 3 of the above algorithm are well defined.

The algorithm described above can be used for a variety of different cut generation schemes that are different in their choice(s) of the objective function in (21). Remarks 4 and 5 address some such cases.

Remark 4. After having derived (19), we might choose to simply maximize u_0 so as to leave the right-hand side in (19) unchanged, but to possibly improve the coefficients on the left-hand side of (19). We can accordingly construct the problem SEP1, replacing v_0 by 1 and $\hat{x}_j$ by 0 for all $j \in J$, and solve this resulting problem using the foregoing algorithm. Note that by (29) and (30), Step 1(c) of this procedure would necessarily detect $f^-(u_{02}) = 1 > 0$, and hence terminate the procedure at Step 4. (That is, u_{02} would be the desired maximum value.)

Remark 5. Note that in Problem SC and SEP1, we can replace the minimum coefficient values of one in constraints (22) by some general integer values $b_j \geq 1 \ \forall j \in J$. In Problem SC, scaling each constraint (22) by its corresponding b_j, and denoting

$$a'_j = \frac{a_j}{b_j} \quad \forall j \in J, \quad \text{and} \quad u'_j = \frac{u_j}{b_j} \quad \forall j \in J,$$

we can solve the following problem using a scheme identical to that used for solving Problem SC.

$$\begin{aligned} \text{Maximize} \quad & a_0 u_0 + \sum_{j \in J} b_j u'_j \\ \text{subject to} \quad & u_0 a'_j + u'_j \geq 1 \quad \forall j \in J \\ & u_0 \geq 0, u'_j \geq 0 \quad \forall j \in J. \end{aligned}$$

Likewise, for Problem SEP1, the following modifications would occur. In Eqs. (27), (28), and (29), the term 1 would be replaced by b_j, $\forall j \in J$. (Equation (30) remains unchanged.) With this revision, the algorithm would follow identically, except that $\frac{1}{a_j}$ should be replaced by $\frac{b_j}{a_j}$ at Step 2, and 1 should be replaced by b_j in defining u_j^* at Step 4 for each $j \in J^+(u_0^*) - j_1$.

Remark 6. We now prescribe a value for ϵ. Let the optimal objective value for Problem SC that resulted in the cut (19) be given by $b_0^* + f_0$, where $b_0^* \equiv |J^*| - 1$ is integer valued and $0 \leq f_0 < 1$ is the fractional part of this value. If we select $b_0^* + 1 - \epsilon \geq b_0^* + f_0$, i.e., $\epsilon \leq 1 - f_0$, we are assured that the maximum value u_{02} for u_0 in the above procedure will be at least that obtained at optimality for Problem SC. Hence, we can select

$$\epsilon = \min\{0.01, 1 - f_0\}.$$

Example 2. Consider the knapsack polytope KP given in Example 1. Note that KP can be tightened by using the standard coefficient reduction scheme (for example, see [23]) to yield the following. (We remark here that Dietrich and Eseudero [8] show that this kind of coefficient reduction can also be obtained via a surrogate procedure.)

$$\text{KP} \equiv Conv\{x \in (0, 1)^4 : 13x_1 + 11x_2 + 11x_3 + 10x_4 \leq 32\}.$$

Suppose that the linear programming solution that we are trying to delete is $\hat{x} = (1, 1/2, 1/2, 1/2)$. With $J = \{1, 2, 3, 4\}$, and $b_j = 1 \ \forall j \in J$, we get via (7) and (8) that $p = 1$ and we get $u_0^* = 1/13$, and $J^* = \{1, 2, 3, 4\}$. This gives (19) as

$$x_1 + x_2 + x_3 + x_4 \leq 3 \tag{31}$$

which is a cover inequality (but not minimal; it is implied by the minimal cover inequality $(x_1 + x_2 + x_3 \leq 2)$. On the other hand, solving Problem SEP1 via the prescribed separation algorithm, we obtain the following, using $\epsilon = 0.01$. Note that $v_0 = 29$ and $\hat{x}_{j_1} = 1$, with $j_1 \equiv 1$.

Step 1. $u_0 = 3.99/32$ gives $\phi(u_0) = 0$. Hence, $u_{02} = 3.99/32$ and $J^-(u_{02}) = \phi$, so that, $f^-(u_{02}) = 29 - 32 < 0$.

Step 2. $\{v_1, \dots v_t\} \equiv \{1/a_j,\ j \in J\}$. Also, $\{a_j(\hat{x}_{j_1} - \hat{x}_j)\} = \{0, 5.5, 5.5, 5\} \equiv \{\theta_j\}$. Hence, $f^-(v_k) = (29 - 32) + \sum_{j\,:\,1/a_j \geq v_k} \theta_j$. This yields $k := 4$, $v_k = 0.1$, and $\phi(0.1) = 3.99 - 3.2 = 0.79$.

Step 4. Hence, $u_0^* = 0.1$, $u_j^* = 0$ for $j = 2, 3, 4$, and $u_1^* = 3.99 - 3.2 = 0.79$. This yields the cut (3) given by

$$2x_1 + x_2 + x_3 + x_4 \leq 3. \tag{32}$$

Note that (32) deletes $\hat{x}$ and is a facet for the knapsack polytope KP. In fact, the separation problem of Crowder et al. [6] yields the minimal cover inequality $x_1 + x_2 + x_3 \leq 2$, which is itself a facet, but which does not delete $\hat{x}$. Also, note that the cut (32) is not available by the usual (simultaneous or sequential) lifting of any *minimal cover inequality* of KP.

Remark 7. We remark here that the cut (32) is available by special lifting procedures of the minimal cover inequality described as follows.

Note that the knapsack polytope KP in Example 2 is a full dimensional polytope. Since $C = \{1, 2, 3\}$ is a minimal cover set, $x_1 + x_2 + x_3 \leq 2$ is a valid inequality for KP. Considering a partition (C_1, C_2) of C, where $C_1 = \{1\}$ and $C_2 = \{2, 3\}$, we have that $x_2 + x_3 \leq 1$ is a facet for KP(1, 4) = KP $\cap \{x_1 = 1,\ x_4 = 0\}$. Applying the simultaneous lifting procedure of Zemel [27], we can identify facets for KP of the type

$$\alpha_1(1 - x_1) + x_2 + x_3 + \alpha_4 x_4 \leq 1,$$

where the cut coefficients (α_1, α_4) are extreme points of the polyhedral cone

$$\Lambda = \{\alpha_1 \leq -1,\ \alpha_1 + \alpha_4 \leq -1,\ \alpha_4 \leq 1\}.$$

This polyhedral cone Λ has two extreme points given by $(\alpha_1, \alpha_4) = (-2, 1)$ and $(-1, 0)$. The first extreme point generates the facet $2x_1 + x_2 + x_3 + x_4 \leq 3$, while the second one yields $x_1 + x_2 + x_3 \leq 2$. Note that the first facet is (32), which we obtained by applying the surrogate-knapsack cut procedure in Example 2.

As Hoffman [19] points out, observe that finding the first extreme point $(\alpha_1, \alpha_4) = (-2, 1)$ can be viewed as a sequential lifting procedure of the partitioned minimal cover inequality. Specifically, if we project out variables x_1 at 1 and x_4 at 0, and generate a minimal cover over the fractional variables, we obtain the inequality $x_2 + x_3 \leq 1$. By lifting back variable x_4, we get $x_2 + x_3 + x_4 \leq 1$, and by lifting back variable x_1, we get $2x_1 + x_2 + x_3 + x_4 \leq 3$.

However, note that our method may find a facet of the underlying problem polytope, which is not available by any lifting procedure applied to minimal cover inequalities. For example, suppose that the original problem constraints included the restrictions $5x_1 + 5x_2 + 5x_3 + 5x_4 \leq 14$, $x_1 - x_2 + x_3 \leq 2$, and $x_1 \leq 1$. Note that the LP solution of Example 2 satisfies each of these constraints. The first constraint is equivalent to $x_1 + x_2 + x_3 + x_4 \leq 2$, and the other constraints are already facetial. Hence, the facet $2x_1 + x_2 + x_3 + x_4 \leq 3$ is not available from any single constraint, although the facet representing the first constraint cuts off the LP solution. However, using surrogate multipliers of 2, 1, and 2 with respect to the above three constraints, respectively, yields the knapsack constraint of Example 2, for which the mentioned facet is derived. Therefore, although this facet is not available from the individual constraints, regardless of any lifting strategy, it is indeed available via the particular surrogate knapsack polytope using our method.

Remark 8. Note that if we solve the problem of Remark 4, for KP given by Example 2, we obtain $u_0^* = 3.99/32$, $u_j^* = 0\ \forall j \in J$, and the cut (3) is given by (31) itself, which does not delete $\hat{x}$, and is in fact, dominated by the minimal cover inequality $x_1 + x_2 + x_3 \leq 2$. However, in Example 1, using the original non-coefficient-reduced knapsack constraint, we did obtain (32) via the procedure suggested in Remark 4. Hence, it is possible that a weaker knapsack constraint can generate a tighter cut via this procedure.

Remark 9. In concluding this section, we remark here that there are several possible ways to tighten the S-K cut (3) after having the derived the surrogate multipliers u_0 and u_j, $j \in J$. We mention one such technique here, and computationally test this scheme later in Section 6.

Suppose that the surrogate constraint derived is of the form $\sum_{j \in N} \alpha_j x_j \leq \alpha_0$, so that (3) is given by $\sum_{j \in N} \lfloor \alpha_j \rfloor x_j \leq \lfloor \alpha_0 \rfloor$. Instead of deriving (3) by rounding down all the coefficients on the left-hand side of the surrogate constraint, let us partition N into sets J_1 and J_2 where the coefficients in J_1 are rounded down but those in J_2 are rounded up. Define f_j, $j \in N$ and f_0 according to

$$\alpha_j = \lfloor \alpha_j \rfloor + f_j \quad \forall j \in N, \quad \alpha_0 = \lfloor \alpha_0 \rfloor + f_0$$

and suppose that the set J_2 satisfies

$$0 < f_j < 1, \quad \forall j \in J_2, \quad \text{and} \quad f_0 + \sum_{j \in J_2} (1 - f_j) < 1.$$

Using this condition, and writing the surrogate constraint as

$$\sum_{j \in J_1} \lfloor \alpha_j \rfloor x_j + \sum_{j \in J_2} \lceil \alpha_j \rceil x_j - \lfloor \alpha_0 \rfloor \leq f_0 + \sum_{j \in J_2} (1 - f_j) x_j - \sum_{j \in J_1} f_j x_j,$$

we observe that the right-hand side is less than 1 for any binary x. Hence, we can derive a Gomory cut of the form,

$$\sum_{j \in J_1} \lfloor \alpha_j \rfloor x_j + \sum_{j \in J_2} \lceil \alpha_j \rceil x_j \leq \lfloor \alpha_0 \rfloor,$$

which is a strengthened form of (3). (Extensions of this analysis to general bounded integer variables are also evident.)

As an illustration, consider KP of Example 2. Using $u_0^* = 1/10$ and $u_j^* = 0\ \forall j \in J$, (3) yields the inequality (31) that is dominated by the minimal cover inequality $x_1 + x_2 + x_3 \leq 2$.

However, noting that $f_0 + (1 - f_1) = 0.2 + 0.7 = 0.9 < 1$, the strengthened version of (3) derived as above is given by $2x_1 + x_2 + x_3 + x_4 \leq 3$, which is a facet of KP.

4. Strengthening the surrogate-knapsack cut: relationship with a sequential lifting process

In the previous section, after formulating the separation Problem SEP0, we chose to solve its variant given by SEP1. Instead of maximizing the weighted sum (21), examining (20), we might have ordered the terms in (20) in descending values of $\hat{x}_j$, $j \in J$, and then *sequentially* maximized each term or coefficient $(u_0 a_j + u_j)$, $j \in J$, in this rank order. For any such term $(u_0 a_k + u_k)$, say, the objective function in (21) would be as if $v_0 = a_k$, $\hat{x}_k = 1$, and $\hat{x}_j = 0$ for $j \neq k$, but in the constraints (22), we would now replace the right-hand side of 1 by the previously determined maximum coefficient for the terms already considered thus far in this sequential process. Theorem 1 below addresses the relationship between this sequential process and the usual sequential lifting process (for example, see [23]).

Theorem 1. *Suppose that* (19) *is of the type* $\sum_{j \in J^*} x_j \leq |J^*| - 1$ *and is a minimal cover inequality. Consider the foregoing sequential process for determining the cut coefficients. Here, given coefficients* $\pi_j \geq 0$, $j \in L \supseteq J^*$ *up to some stage, we select* $t \in \bar{L}$ *(the complement of* L*) and solve Problem* SEP1S *given below, letting* π_t *equal the rounded-down optimal objective value.*

$$\begin{aligned}
\text{SEP1S: } \textit{Maximize} \quad & (u_0 a_t + u_t) \\
\textit{subject to} \quad & u_0 a_0 + \sum_{j \in L - \{t\}} u_j = |J^*| - \epsilon && (33) \\
& u_0 a_j + u_j \geq \pi_j \quad \forall j \in L && (34) \\
& u_0 \geq 0, \quad u_j \geq 0 \quad \forall j \in L + \{t\}.
\end{aligned}$$

Let $0 < \epsilon \leq \min_{j \in N}\{1/a_j\}$. *If for each stage, Problem* SEP1S *yields* $u_t > 0$ *at optimality, and furthermore, if* $\pi_t \leq 1$ *for each* t *except perhaps for the final* t *considered, then the resulting cut is a facet of KP.*

Proof: Consider Problem SEP1S and suppose that $\pi_j \in \{0, 1\}$ $\forall j \in L$. Furthermore, assume that $\sum_{j \in L} \pi_j x_j \leq |J^*| - 1$ is a valid inequality that is a facet of the restricted knapsack polytope, $\text{KP}(L) \equiv \text{KP} \cap \{x_j = 0, \ j \in \bar{L}\}$, in which the variables in the complement $\bar{L}$ of L have been fixed at zero, as determined by the sequential lifting of the given minimal cover inequality using the procedure of Balas and Zemel [2]. Note that this is true for $L \equiv J^*$. Inductively, we will prove the assertion of the theorem by demonstrating that the coefficient π_t determined via Problem SEP1S would then coincide with the coefficient π_t^*, say, that would be obtained via Balas and Zemel's sequential lifting process.

Toward this end, consider the dual of Problem SEP1S. This is given as follows, where x_0 and x_j, $j \in L$, are the dual variables associated with the constraints (33) and (34), respectively.

$$\begin{aligned} \text{Minimize} \quad & (|J^*| - \epsilon)x_0 - \sum_{j \in L} \pi_j x_j \\ \text{subject to} \quad & a_0 x_0 - \sum_{j \in L} a_j x_j \geq a_t \\ & 0 \leq x_j \leq x_0 \quad \forall j \in L, \quad x_0 \geq 1. \end{aligned} \tag{35}$$

Since $u_t > 0$ at optimality by our hypothesis, the related complementary slackness condition asserts that $x_0 \equiv 1$, and so, we have,

$$\begin{aligned} \pi_t = \text{ rounded-down value of } \{|J^*| - \epsilon - \max \quad & \sum_{j \in L} \pi_j x_j \\ \text{subject to} \quad & \sum_{j \in L} a_j x_j \leq a_0 - a_t \\ & 0 \leq x_j \leq 1 \quad \forall j \in L. \end{aligned} \tag{36}$$

Note that from Balas and Zemel's procedure, we would have obtained

$$\pi_t^L = (|J^*| - 1) - \max \quad \sum_{j \in L} \pi_j x_j \tag{37}$$

$$\text{subject to} \quad \sum_{j \in L} a_j x_j \leq a_0 - a_t \tag{38}$$

$$x_j \in (0, 1) \quad \forall j \in L.$$

Let ν(LP) and ν(IP) respectively denote the optimal values of the maximization problems in (36) and (37). Noting that $\pi_j \in \{0, 1\}$ $\forall j \in L$, ν(IP) is determined by considering $L' = \{j \in L : \pi_j = 1\} \supseteq J^*$, ordering the indices in L' in ascending order of a_j, and then by setting $x_j = 1$ for $j \in L'$ in this order, while the sum of the corresponding a_j remains less than or equal to $(a_0 - a_t)$. But ν(LP) is determined in the same manner, except that after setting the final variable at unity as for ν(IP), we also set the next variable x_p, say, at $x_p = s/a_p$, where $0 \leq s \leq a_p - 1$ is the slack in (38) at optimality. Hence, we have,

$$\nu(\text{LP}) = \nu(\text{IP}) + \frac{s}{a_p} \leq \nu(\text{IP}) + \frac{a_p - 1}{a_p} \leq \nu(\text{IP}) + (1 - \epsilon). \tag{39}$$

Therefore, from (39), we obtain $|J^*| - \epsilon - \nu(\text{LP}) \geq |J^*| - 1 - \nu(\text{IP}) \equiv \pi_t^L$, and so,

$$\pi_t = \lfloor |J^*| - \epsilon - \nu(\text{LP}) \rfloor \geq \pi_t^L. \tag{40}$$

But since $\sum_{j \in L} \pi_j x_j + \pi_t^L x_t \leq |J^*| - 1$ is a facet of $\text{KP}(L+t) \equiv \text{KP} \cap \{x_j = 0,\ j \in \bar{L} - \{t\}\}$ by Balas and Zemel's procedure, and since $\sum_{j \in L} \pi_j x_j + \pi_t x_t \leq |J^*| - 1$ is valid for KP via Problem SEP1S, and hence, it is valid for KP$(L + t)$, we must have $\pi_t \leq \pi_t^L$. This together with (40) establishes that $\pi_t = \pi_t^L$, and hence, the proof is complete. □

Example 3. Suppose that after deriving (31) as in Example 2, we now attempt to lift the coefficients in the order $\{1, 2, 3, 4\}$, of decreasing $\hat{x}_j$ values. For maximizing $(a_1 u_0 + u_1)$, taking $v_0 = a_1 = 13$, and $\hat{x}_1 = 1$, $\hat{x}_j = 0\ \forall j \neq 1$, we get the following steps of the algorithm.

Step 1. As in Example 2.
Step 2. $\{v_1, \ldots, v_r\} \equiv \{\frac{1}{a_j},\ j \in J\}$ and $\theta_j \equiv \{0, 11, 11, 10\}$. Hence, $f^-(v_k) = (13 - 32) + \sum_{j:1/a_j \geq v_k} \theta_j$, which yields, $k = 3$, $v_k = 1/11$ and $\phi(v_k) = 3.99 - 32/11 + (10/11 - 1) = 0.99$.
Step 4. Hence, $u_0^* = 1/11$, $u_4^* = 1 - 10/11$, $u_2^* = u_3^* = 0$, and $u_1^* = \phi(u_0^*) = 0.99$. This yields the cut (3) given by

$$2x_1 + x_2 + x_3 + x_4 \leq 3,$$

which is the same as (32). Since this is already facetial, we know that a further sequential maximization of the other coefficients will not yield any further improvements in this cut.

Remark 10. Note that in contrast with the sequential lifting process, each stage of the surrogate-knapsack cut process embodied by solving SEP1S, determines surrogate coefficients that might yield nonzero coefficients for several $j \in \bar{L}$, besides just for t. Hence, while the sequential lifting process is able to consider the derivation of π_t^* under the consideration of $x_t = 1$, the analogous step of using $x_t = 1$ with an unconstrained multiplier u_t in SEP1S would not necessarily yield a valid surrogate-knapsack cut. Moreover, comparing (37) with (35) and (36), the case of π_t being possibly smaller than π_t^L is evident. Hence, although the conditions specified in Theorem 1 under which a facet is obtained are somewhat restrictive, the analysis provides some insight into the connection between the sequentially generated surrogate-knapsack cut and the sequential lifting process. Also, note that in SEP1S, it is entirely possible (in fact likely) that $u_t > 0$ even when $\pi_t = 0$. Furthermore, it is also possible that $u_t = 0$ while $\pi_t \geq 1$. In the latter case, we lose connection with the sequential lifting process. However, if π_t turns out to equal π_t^L in case $u_t = 0$, as well, then so long as $\pi_t \in \{0, 1\}$, except possibly for the final t considered, we will obtain a facet of KP.

5. Fractional surrogate constraint cuts

In this section, we consider a related simple and fast way to generate a cut that deletes a fractional linear programming vertex solution using surrogate constraint analysis. For this, we use the cutting plane formulation of Glover [10]. Assume that the current LP tableau (in the updated basis representation) contains the equation

$$y + \sum_{j \in N_b} a_j x_j = a_0 \tag{41}$$

where y is a basic integer variable, and the x_j, for $j \in N_b$, are current nonbasic integer variables, which must satisfy $x_j \leq 1$ for $j \in N_b' \subseteq N_b$, i.e., N_b' represents the index set of binary nonbasic variables. We also suppose that a_0 is not integer, as must occur (for some such equation) if the LP extreme point is not integer feasible. All variables are assumed nonnegative. From this we may then derive the cutting plane

$$s + \sum_{j\in N_b} (pa_j - \lceil ha_j \rceil) x_j = pa_0 - \lceil ha_0 \rceil, \tag{42}$$

where $\lceil\cdot\rceil$ denotes the rounded-up value, s is a nonnegative integer variable, p is an integer parameter such that pa_0 is non-integer, and h is a parameter chosen to satisfy two conditions: $p - 1 < h \le p$, and $h \ge (\lceil pa_0 \rceil - 1)/a_0$. These respectively assure that $p = \lceil h \rceil$ and that the right-hand side of (42) is negative, where the latter condition also ensures that the cut eliminates the current LP vertex. As shown in Glover [10] the cutting plane representation (42) makes it possible to generate all of the standard "fractional cuts" of Gomory [13] and also to generate additional types of fractional cuts that are stronger. We now show how to extend this cut using surrogate constraint analysis to obtain a still stronger cut. Define $f_j = \lceil ha_j \rceil - ha_j$ for $j \in N_b$ and $j = 0$. Then for $h = p$ the Eq. (42) yields the customary fractional cut

$$s + \sum_{j\in N_b} -f_j x_j = -f_0.$$

The following observations do not require $h = p$, and hence we can combine them with earlier results for obtaining values of h that may yield stronger cuts. However, we also obtain stronger cuts for the case $h = p$. We proceed as in the derivation of (42), first multiplying the source Eq. (41) by the parameter h to yield a modified source equation:

$$hy + \sum_{j\in N_b} ha_j x_j = ha_0.$$

Now, we additionally generate a surrogate constraint using this along with the inequalities $-u_j x_j \ge -u_j$ for $j \in N_b'$, where $u_j \ge 0$ is a surrogate multiplier associated with $-x_j \ge -1$, for each $j \in N_b'$. For convenience, we take $u_j \equiv 0$ for $j \in N_b - N_b'$, and obtain the surrogate constraint

$$hy + \sum_{j\in N_b} (ha_j - u_j) x_j \ge ha_0 - u_0,$$

where $u_0 = \sum_{j\in N_b} u_j$. A Gomory all-integer cut from this inequality can be written as

$$\lceil h \rceil y + \sum_{j\in N_b} \lceil ha_j - u_j \rceil x_j \ge \lceil ha_0 - u_0 \rceil,$$

and by introducing a nonnegative, integer valued slack variable s we can write the foregoing inequality as

$$s - \lceil h \rceil y - \sum_{j\in N_b} \lceil ha_j - u_j \rceil x_j = -\lceil ha_0 - u_0 \rceil.$$

We now multiply the source Eq. (41) by p and add the result to the preceding cut. By choosing h to yield $\lceil h \rceil = p$, as indicated earlier, this gives

$$s + \sum_{j \in N_b} (pa_j - \lceil ha_j - u_j \rceil) x_j = pa_0 - \lceil ha_0 - u_0 \rceil \tag{43}$$

which implies that

$$\sum_{j \in N_b} (pa_j - \lceil ha_j - u_j \rceil) x_j \leq pa_0 - \lceil ha_0 - u_0 \rceil .$$

We observe that the foregoing surrogate constraint cut reduces to the cut (42) when $u_j = 0 \, \forall j$. Furthermore, if u_j can be made sufficiently large to yield $\lceil ha_j - u_j \rceil < \lceil ha_j \rceil$, while keeping u_0 small enough to assure that $\lceil ha_0 - u_0 \rceil = \lceil ha_0 \rceil$, then the surrogate constraint cut (43) will be strictly stronger than (42). Our goal is to specify the conditions under which this occurs, and to identify the form of the surrogate constraint cut more precisely under these conditions. For this, in direct analogy to the definition of f_j, define $g_j = \lceil ha_j \rceil - pa_j$, $\forall j \in N_b \cup \{0\}$ so that the cut (42) can be written in the form

$$s - \sum_{j \in N_b} g_j x_j = -g_0 . \tag{44}$$

Theorem 2. *Define $r_j = 1 - (\lceil ha_j \rceil - ha_j) \; \forall j \in N_b \cup \{0\}$, and let N_b'' be any subset of N_b' such that $\sum_{j \in N_b''} r_j < r_0$. Then we can create a surrogate constraint cut* (43) *that strictly dominates* (42) *(if N_b'' is nonempty) having the form*

$$s + \sum_{j \in N_b''} (1 - g_j) x_j - \sum_{j \in N_b - N_b''} g_j x_j = -g_0 . \tag{45}$$

Proof: To retain the right-hand side of (43) the same as that in (42), we require $\lceil ha_0 - u_0 \rceil = \lceil ha_0 \rceil$. This implies that $u_0 < 1 - (\lceil ha_0 \rceil - ha_0)$, or equivalently, $u_0 < r_0$. To increase the coefficient of x_j in the cut, replacing $-g_j$ by $1 - g_j$ for $j \in N_b''$, we require $\lceil ha_j - u_j \rceil < \lceil ha_j \rceil$, which implies that $u_j \geq 1 - (\lceil ha_j \rceil - ha_j)$, or equivalently, $u_j \geq r_j$. (This inequality also holds if we increase the coefficient by more than 1, but the next step shows that this is impossible.) Consequently, from $u_0 = \sum_{j \in N_b''} u_j$, with $u_j = 0$ for $j \in N_b - N_b''$, we see that the stipulated conditions can hold if and only if $\sum_{j \in N_b''} r_j < r_0$, and the proof of the theorem is complete. □

Example 4. We give an example of the preceding theorem for the case where we choose $h = p$, so that (42), and hence (44), is a standard fractional cut. In particular, suppose that this fractional cut is the following:

$$s - (5/8)x_1 - (7/8)x_2 - (1/8)x_3 = -1/8.$$

Assume that all x_j variables are 0-1 variables. Then we have $r_1 = 3/8$, $r_2 = 1/8$, $r_3 = 7/8$ and $r_0 = 7/8$. By the condition $\sum_{j \in N_b''} r_j < r_0$, we can select $N_b'' = \{1, 2\}$, which gives the surrogate cut (45) in the form

$$s + (3/8)x_1 + (1/8)x_2 - (1/8)x_3 = -1/8.$$

(As before, the s variables in these cuts are not the same.) It is interesting to note that the possibility to form strengthened surrogate constraint cuts of this type rely on the right-hand side of the standard cut being relatively small in absolute value, which normally is associated with a weaker cut. Here, for example, if we suppose that the original fractional cut is obtained for $h = p = 1$, then a stronger fractional cut can be obtained by setting $p = h = 7$ (or, due to periodicity, setting $p = h = -1$), to yield

$$s - (3/8)x_1 - (1/8)x_2 - (7/8)x_3 = -7/8.$$

This cut indeed strictly dominates the first fractional cut shown above. Nevertheless, this cut in turn is also strongly dominated by the surrogate constraint cut (derived for the same p and h values that produced the "weaker" fractional cut).

In general, it can be seen that the potential strengthening attained by the fractional surrogate constraint cut (45) occurs over limited sets of coefficients, but can be quite significant along the dimensions where these coefficients are increased. Thus, practically speaking, it appears relevant to generate a collection of such cuts to make different subsets of cut coefficients positive, thus driving the LP solution away from regions into which it may otherwise be drawn by the customary cutting planes. The fact that the cuts (45) can be generated virtually as quickly as the standard fractional cuts, without requiring a more involved separation procedure to be applied, is a convenient feature.

We can directly extend this analysis to obtain additional strengthened cuts for problems with GUB constraints.

Corollary 3. *Consider the GUB knapsack polytope defined in* (2). *Define r_j as in Theorem* 2, *and refine the definition of N_b'' to require that N_b'' contains at most one index $j(i) \in N_i$ for each $i \in M$. Finally, define $N'' = \{h \in N_i : i \in M_{N_b''} \text{ and } r_h \leq r_{j(i)}\}$, where $M_{N_b''} = \{i \in M : |N_i \cap N_b''| = 1\}$. Then, the following (fractional) surrogate constraint cut strictly dominates the cut* (45) *of Theorem* 2 *if N'' strictly contains N_b''*

$$s + \sum_{j\in N''}(1 - g_j)x_j - \sum_{j\in N-N''} g_j x_j = -g_0. \tag{46}$$

Proof: The corollary is readily verified by adapting the proof of Theorem 2 to include consideration of the GUB constraints. □

The ability to make still larger subsets of coefficients positive in these extended cuts, coupled with the ability to generate these cuts very quickly, is attractive for providing alternatives of greater strength for GUB constrained problems.

6. Computational results

In this section, we report our computational experience using a set of ten test 0-1 integer programming problems taken from MIPLIB [3]. We selected these test problems because they represent real-world problems having known optimal solutions and because they comprise a standard test bed of integer programs. A summary of these test problems is given in Table 1. Six of these problems are part of the test set used by Crowder et al. [6], referred to as the P-problems. In Table 1, ν_{LP} represents the optimal objective function value of the linear programming relaxation at the root node, and ν_{IP} represents the optimal objective value of the integer program.

It is important to note that the primary purpose of this computational testing is *not* to attempt to outperform the well established branch-and-cut codes (for example, CPLEX [5] and MINTO, [25]), since these codes owe their performance to a variety of enhanced techniques other than cutting planes. The goal of this computational study is to determine the strength of the surrogate-knapsack cuts at the root node, by comparison to alternative cutting approaches that are incorporated in state-of-the-art methods, and thus to identify the potential merit of our results independent of the use of supplemental strategies as embedded in various preprocessing and auxiliary analysis techniques.

There are many algorithmic and implementation choices for selecting critical algorithm parameters including the selection of J^* and ϵ in the separation problem. Rather than seek the most refined choices possible, we provide a default algorithm, called Algorithm S-K Cut, which represents a set of strategies that we consider to be reasonable. We then present computational results that compare the performance of this default algorithm using our S-K cuts to the performance of the cutting plane process embedded in the CPLEX 3.0 MIP solver. As will be seen, the default algorithm using the S-K cuts dominates the lifted cover cutting planes used by the CPLEX 3.0 MIP solver.

For a given fractional solution $\hat{x}$, the default algorithm attempts to generate violated S-K cut inequalities. For each nontrivial knapsack constraint of the form $\sum_{j \in N} a_j x_j \leq a_0$, where $a_j \geq 0\ \forall j \in N$, $a_0 > 0$, and not all coefficients a_j are equal to 0 or 1, we perform the

Table 1. Test problem summary.

Problem	Variables	Rows	ν_{LP}	ν_{IP}
bm23	27	20	20.6	34
lseu	89	28	834.7	1120
mod008	319	6	290.9	307
p0033	33	16	2520.6	3089
p0201	201	134	6875.0	7615
p0282	282	242	176867.5	258411
p0291	291	253	1705.1	5223.7
p0548	548	177	315.3	8691
p2756	2756	756	2688.7	3124
sentoy	60	30	−7839.3	−7772

following steps to detect a violated S-K cut. If a violated S-K cut is found, then we augment the LP with this cut and repeat the cut generation procedure. Otherwise, we terminate the algorithm.

Algorithm S-K cut

Step 1 (Initialization). Sort the variables x_j in order of non-increasing values of $a_j \hat{x}_j$. Proceeding in this order, select J as a cover set such that the total sum of a_j for j in J just exceeds a_0. If no such set is found, then stop; the particular knapsack constraint is trivially redundant. Otherwise, proceed to Step 2.

Step 2 (Separation problem). Solve Problem SC of Section 2 with $b_j = 1$ $j \in J$. From (8) and (9), this yields $J^* \equiv J$, $u_0^* = 1/a_p$ where $a_p = \max_{j \in J}\{a_j\}$, and that the optimal objective function value of Problem SC is given by $b_0^* + f_0$, where $b_0^* = |J^*| - 1$ and $f_0 = 1 - (\sum_{j \in J^*} a_j - a_0)u_0^*$. Following Remark 6, select $\epsilon = 1 - f_0$, and solve Problem SEP1 of Section 3 using the algorithm described therein. This yields surrogate multipliers u_0 and u_j, $j \in J$, for deriving an S-K cut (3).

Step 3 (Improved S-K cut). Use Remark 9 of Section 3 to derive a strengthened version of (3) by selecting J_2 in nondecreasing order of $(1 - f_j)$, so long as $f_0 + \sum_{j \in J_2}(1 - f_j) < 1$ holds true. Let this strengthened S-K cut be given by $\sum_{j \in J_1} \lfloor \alpha_j \rfloor x_j + \sum_{j \in J_2} \lceil \alpha_j \rceil x_j \leq \lfloor \alpha_0 \rfloor$ as in Remark 9. If $\sum_{j \in J_1} \lfloor \alpha_j \rfloor \hat{x}_j + \sum_{j \in J_2} \lceil \alpha_j \rceil \hat{x}_j > \lfloor \alpha_0 \rfloor$, then augment the current LP relaxation with the violated S-K cut. Otherwise, stop.

The computational results obtained using these S-K cuts on the ten test problems are shown in Table 2. All runs were performed on a SUN SPARC 10 workstation using CPLEX 3.0 [5] as an LP solver. We compare these results with the solver CPLEX 3.0 MIP that uses the classical lifted cover (LC) cuts for tightening the LP value at the root node. In Table 2, observe that the proposed algorithm using the S-K cut provides a tighter

Table 2. Computational results.

	S-K cuts			LC cuts		
Problem	Z_{root}	Time (sec)	Cuts	Z_{root}	Time (sec)	Cuts
bm23	22.7	0.1	9	22.5	0.1	1
lseu	1001.2	0.3	14	999.5	0.2	13
mod008	291.7	0.6	5	291.3	0.2	5
p0033	2902.6	0.1	15	2916.2	0.2	13
p0201	7075.0	0.8	3	7075.0	0.9	2
p0282	252356.0	2.5	89	180999.7	1.2	58
p0291	5009.2	1.0	28	1873.8	1.3	25
p0548	3883.7	8.1	158	4052.9	2.5	138
p2756	2701.8	16.4	75	2701.7	10.5	68
sentoy	−7837.7	0.2	5	−7832.5	0.3	5

LP objective function value (Z_{root}) than the same procedure using LC cuts for six test problems, and generates the same tight objective value as the LC cuts approach for the two other problems. Hence, the use of S-K cuts appears to be as effective as generating facets of the individual knapsack polyhedra defined by the problem constraints. In a sequel to this paper, we examine the effect of these and other S-K cuts in greater detail, by conducting an intensive computational study for solving large-scale 0-1 IP problems within the framework of a branch and cut procedure that incorporates various enhanced solution techniques including preprocessing, variable fixing, and heuristics.

Remark 11. Note that our procedure for solving the separation problem SEP1 of Section 3 is more efficient than using a standard LP code. For example, our method consumed 1.6 seconds (total CPU time for the separation procedure) for solving the model SEP1 of Problem p0548, while the CPLEX LP solver requires 50.5 seconds. Similar computational advantages of our approach over the use of CPLEX were observed for solving the separation models of the other test problems.

7. Conclusion

Surrogate constraint analysis provides a convenient tool for producing cutting planes that are stronger than a variety of those previously introduced in the literature. The basic tools, including the idea of generating cutting planes as part of a recursive surrogate constraint definition, derive from approaches of the 1960s to generate stronger cutting planes—in particular, by forming linear combinations including classical Gomory cuts, and identifying specific parameters for the resulting representation that are guaranteed to yield cuts having specific properties.

The present paper shows how to extend this work to yield new and more general cutting planes. Likewise, we show how to strengthen these cuts subject to constraining the values of selected coefficients. In addition, we give a separation procedure to remove non-integer linear programming vertices, and demonstrate that these new cuts include facets that are not available to standard lifting procedures.

An interesting potential application of our results, reinforcing the theme by which they are derived, is to embed them in recursive cut generation methods to obtain "best cutting plane parameters" at each stage. In addition, the fact that we can *design* the cuts to be stronger relative to selected variables invites the generation of cuts that yield complementary strengths over different dimensions as a way of taking advantage of problems having special structures.

Acknowledgments

This work has been supported in part by the National Science and Engineering Council of Canada under Grants 5-83998 and 5-84181, and by the National Science Foundation under Grant DMI-9521398.

References

1. E. Balas, "Facets of the knapsack polytope," Mathematical Programming, vol. 8, pp. 146–164, 1975.
2. E. Balas and E. Zemel, "Facets of the knapsack polytope from minimal covers," SIAM Journal on Applied Mathematics, vol. 34, pp. 119–148, 1978.
3. R.E. Bixby, E.A. Boyd, S.S. Dadmehr, and R.R. Indovina, "The MIPLIB mixed integer library," Technical Report R92-36, Rice University, 1992.
4. V. Chvátal, "Edmonds polytopes and a hierarchy of combinatorial problems," Discrete Mathematics, vol. 4, pp. 305–357, 1973.
5. CPLEX Version 3.0, Using the CPLEX Callable Library, CPLEX Optimization, Inc., 1995.
6. H. Crowder, E L. Johnson, and M. Padberg, "Solving large-scale zero-one linear programming problems," Operations Research, vol. 31, pp. 803–834, 1983.
7. B.L. Dietrich, L.F. Escudero, and F. Chance, "Efficient reformulation for 0–1 programs, methods and computational results," Discrete Applied Mathematics, vol. 42, pp. 147–175, 1993.
8. B.L. Dietrich and L.F. Escudero, "Obtaining clique, cover and coefficient reduction inequalities as Chvátal-Gomory inequalities and Gomory fractional cuts," European Journal of Operational Research, vol. 73, pp. 539–546, 1994.
9. F. Glover, "A multi-phase-dual algorithm for the zero-one integer programming problem," Operations Research, vol. 13, pp. 879–919, 1965.
10. F. Glover, "Generalized cuts in diophantine programming," Management Science, vol. 13, pp. 254–268, 1966.
11. F. Glover, "Surrogate constraints," Operations Research, vol. 16, pp. 741–749, 1968.
12. F. Glover, "Surrogate constraint duality in mathematical programming," Operations Research, vol. 23, pp. 434–451, 1975.
13. R.E. Gomory, "Outline of an algorithm for integer solutions to linear programs," Bulletin of the American Mathematical Society, vol. 64, pp. 275–278, 1958.
14. R.E. Gomory, "Solving linear programming problems in integers," in Combinatorial Analysis, R.E. Bellman and M. Hall, Jr. (Eds.), American Mathematical Society, pp. 211–216, 1960.
15. R.E. Gomory, "An algorithm for integer solutions to linear programs," in Recent Advances in Mathematical Programming, R. Graves and P. Wolfe (Eds.), McGraw-Hill, pp. 269–362.
16. H.J. Greenberg and W.P. Pierskalla, "Surrogate mathematical programs," Operations Research, vol. 18, pp. 924–939, 1970.
17. H.J. Greenberg and W.P. Pierskalla, "Quasi conjugate functions and surrogate duality," Cahiers du Centre dÉtudes de Recherche Operationell, vol. 15, pp. 230–334, 1973.
18. P.L. Hammer, E.L Johnson, and U.N. Peled, "Facets of regular 0-1 polytopes," Mathematical Programming, vol. 8, pp. 179–206, 1975.
19. K. Hoffman, Informal Communication, May 1996.
20. K. Hoffman and M. Padberg, "Improving LP-representations of zero-one linear programs for branch-and-cut," ORSA Journal on Computing, vol. 3, pp. 121–134, 1991.
21. M.H. Karwan and R.L. Rardin, "Some relationships between Lagrangian and surrogate duality in integer programming," Mathematical Programming, vol. 17, pp. 320–324, 1979.
22. R.K. Martin and L. Schrage, "Subset coefficient reduction cuts for 0-1 mixed integer programming," Operations Research, vol. 33, pp. 505–526, 1985.
23. G.L. Nemhauser and L.A. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons: New York, 1988.
24. M. Padberg, "$(1, k)$ configuration and facets for packing problem," Mathematical Programming, vol. 18, pp. 94–99, 1980.
25. M.W.P. Savelsbergh and G.L. Nemhauser, Mixed INTeger Optimizer (MINTO) Version 2.0, 1994.
26. L.A. Wolsey, "Faces for a linear inequality in 0-1 variables," Mathematical Programming, vol. 8, pp. 165–178, 1975.
27. E. Zemel, "Lifting the facets of zero-one polytopes," Mathematical Programming, vol. 15, pp. 268–277, 1978.