Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 179 (2007) 50–63
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

Exploiting nested inequalities and surrogate constraints

Saïd Hanafi [a,*], Fred Glover [b]

[a] Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines, UMR CNRS 8530, Groupe Recherche Opérationnelle et Informatique, Université de Valenciennes et du Hainaut-Cambrésis, Le Mont Houy, 59313 Valenciennes Cedex, France
[b] Leeds School of Business, University of Colorado, Boulder, CO 80309-0419, United States

Received 7 September 2005; accepted 13 March 2006
Available online 13 June 2006

Abstract

The exploitation of nested inequalities and surrogate constraints as originally proposed in Glover [Glover, F., 1965. A multiphase-dual algorithm for the zero–one integer programming problem. Operations Research 13, 879–919; Glover, F., 1971. Flows in arborescences. Management Science 17, 568–586] has been specialized to multidimensional knapsack problems in Osorio et al. [Osorio, M.A., Glover, F., Hammer, P., 2002. Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions. Annals of Operations Research 117, 71–93]. We show how this specialized exploitation can be strengthened to give better results. This outcome results by a series of observations based on surrogate constraint duality and properties of nested inequalities. The consequences of these observations are illustrated by numerical examples to provide insights into uses of surrogate constraints and nested inequalities that can be useful in a variety of problem settings.

Keywords: Integer programming; Nested cuts; Multidimensional knapsack problem; Surrogate constraints

1. Introduction

A general *integer programming* (IP) problem consists of optimizing (minimizing or maximizing) a linear function subject to linear inequality and/or equality constraints, where all of the variables are required to be integral. An IP problem (which we assume is to be maximized) can be expressed as follows:

$$\begin{aligned} \text{(IP)}\quad \text{Maximize} \quad & x_0 = cx \\ \text{Subject to} \quad & A_i x \leqslant A_i^0 \quad \text{for } i \in M = \{1, 2, \ldots, m\}, \\ & 0 \leqslant x_j \leqslant U_j \quad \text{for } j \in N = \{1, 2, \ldots, n\}, \\ & x_j \text{ integer} \quad \text{for } j \in N. \end{aligned}$$

* Corresponding author.
E-mail addresses: Said.Hanafi@univ-valenciennes.fr, shanafi@univ-valenciennes.fr (S. Hanafi), fred.glover@colorado.edu (F. Glover).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2006.03.046

The variable x_0 identifies the objective function value of a feasible solution x defined by n decision variables x_j for $j \in N$. The vector $c \in R^n$ denotes the cost vector and the vector A^0 denotes the right-hand side of m linear constraints $A_i x \leqslant A_i^0$ for $i \in M$. No special structure is assumed for the input matrices $c(1 \times n)$, $A(m \times n), A^0(m \times 1)$, $b(n \times 1)$. The parameter U_j refer to an upper bound on the integer variable x_j.

Problem (IP) reduces to the *binary integer program* (01-IP) when all integer variables must equal 0 or 1 (i.e., $U_j = 1$, for all $j \in N$). The zero–one multidimensional knapsack (MDK) is also a subproblem of many general integer programs where the components of the data matrices c, A and A^0 are given non-negative integers. In the following, without loss of generality, we consider the case of the zero–one multidimensional knapsack. Letting e denote a vector with all components equal to 1, the zero–one multidimensional knapsack (MDK) problem can be expressed as follows:

$$\text{(MDK)} \quad \text{Maximize} \quad x_0 = cx \tag{1-a}$$

$$Ax \leqslant A^0, \tag{1-b}$$

$$0 \leqslant x \leqslant e, \tag{1-c}$$

$$x \in \{0,1\}^n. \tag{1-d}$$

The foregoing MDK formulation, where A and A^0 are non-negative, can model many combinatorial optimization problems, including capital budgeting, cargo loading, cutting-stock problems, and a variety of others (see Fréville, 2004, Fréville and Hanafi, 2005). MDK also arises as a subproblem in solving many other combinatorial optimization problems. Complexity results have not yet definitively identified the level of difficulty of these problems, but empirical findings suggest that the computational resources required to solve certain MDK problem instances can grow exponentially with the size of problem.

The exploitation of nested inequalities and surrogate constraints as originally proposed in Glover (1965, 1971) has been specialized to multidimensional knapsack problems in Osorio et al. (2002). In this paper, we show how this specialized exploitation can be strengthened to give better results. This outcome results by a series of observations based on surrogate constraint duality and properties of nested inequalities. The consequences of these observations are illustrated by numerical examples to provide insights into uses of surrogate constraints and nested inequalities that can be useful in a variety of problem settings. Recently, Osorio and Gómez (2004) proposed cutting analysis for MDK.

2. Mixed surrogate constraint

Bounding procedures that compute lower and upper bounds on the optimum x_0 value are useful for solving MDK. Upper bounds are provided by relaxation or duality techniques. Lower bounds are generally provided by heuristic and/or metaheuristic procedures using restriction techniques.

Most commercial Branch-and-Bound (B&B) procedures use the LP-relaxation to compute the bound function. Formally, the LP-relaxation of MDK, denoted by LP-MDK, where all variables are allowed to be continuous, can be defined as follows:

$$\text{LP-MDK} \quad \text{Maximize}\{x_0 = cx : Ax \leqslant A^0 \text{ and } 0 \leqslant x \leqslant e\}.$$

Bounds derived from other relaxations can sometimes be generated more readily than those obtained from LP relaxation, and in certain cases can be stronger than the LP bounds. In particular, Lagrangean relaxation, surrogate relaxation and composite relaxation, are often used to obtain such upper bounds. Lagrangean strategies have been shown to provide an effective tool for solving integer programming problems (see, for example, Geoffrion, 1974; Fischer, 1981). The Lagrangean relaxation absorbs a set of constraints into the objective function.

Surrogate constraint methods, which we focus on here, have been embedded in a variety of mathematical programming applications over the past thirty years. The surrogate relaxation, introduced by Glover (1965), replaces sum of the original constraints by a single new one, called a surrogate constraint (see also Glover (1968)). A surrogate relaxation $S(\mu)$ of MDK, where $\mu \in R^m$ is a vector of "mutipliers" satisfying $\mu \geqslant 0$, is defined as

$$S(\mu) \quad \max\{x_0 = cx : x \in \{0,1\}^n \text{ and } x \leqslant d^0\}, \tag{2}$$

where $d = \mu A$ and $d^0 = \mu A^0$.

We assume the surrogate constraint (2) does not include weighted combinations of the upper or lower bounds on the problem variables. The surrogate dual (S), defined as follows, yields the strongest surrogate constraint:

$$(\mathrm{S}) \quad \min\{S(\mu) : \mu \geqslant 0\}.$$

This dual in general yields stronger bounds for combinatorial optimization problems than the Lagrangian dual. The most widely used search methods for solving a surrogate dual problem are based on the properties of the corresponding relaxation function $S(\mu)$. Greenberg and Pierskalla (1970) showed that the surrogate function $S(\mu)$ is a quasi-convex function of the multiplier μ, and it is a discontinuous piecewise linear function for the MDK problem. This property assures that any local optimum for the surrogate function is also a global optimum.

In the following, the term *simple bounding constraint* refers to a constraint that imposes a lower or upper bound on a variable (such as $x_j \geqslant 0$ or $x_j \leqslant 1$). The term *component constraint* refers to a constraint that receives a non-zero weight in forming a surrogate constraint. An inequality or, more generally, a system of inequalities will be said to be *strengthened* (or *made stronger*) if the new system yields a set of feasible solutions contained within the set of feasible solutions to the original system.

The term x_o *constraint* (or *objective function constraint*) refers to a constraint of the form $x_o \geqslant x_o^* + \varepsilon$, where $x_o^* = cx^*$ is the x_o value for the best feasible solution x^* currently known, and ε is a chosen tolerance for approximating the inequality $x_o > x_o^*$ (which may permissibly equal the greatest common divisor of the c_j coefficients when c is an integer vector).

The term *mixed surrogate constraint* refers to a surrogate constraint created by combining a given surrogate constraint (2) (called the *component surrogate constraint*) with an objective function constraint. To create the mixed surrogate constraint, we write the associated objective function constraint as a "$\leqslant$" constraint to give it the same orientation as the surrogate constraint (2):

$$-cx \leqslant -cx^* - \varepsilon. \tag{3}$$

Consequently, by weighting (2) by α and (3) by β, the mixed surrogate constraint is

$$\pi x \leqslant \pi_o \tag{4}$$

with $\pi = \alpha d - \beta c$ and $\pi_o = \alpha d^0 - \beta\,(cx^* + \varepsilon)$.

We begin with an exceedingly straightforward observation that nevertheless has important consequences.

Observation 1. Surrogate constraints can be made stronger by excluding simple bounding constraints as component constraints.

This observation is an immediate consequence of the fact that the bounds on the variables are directly exploited by the methods that extract information from surrogate constraints, and hence folding such bounds into the constraints themselves creates an unnecessary degree of relaxation. Similarly, any constraints that are exploited in conjunction with surrogate constraints should not be included as component constraints. In the present context, therefore, Observation 1 can be extended to exclude nested inequalities as component constraints – except where a set of such inequalities is different from the one being exploited in connection with the surrogate constraint in a particular instance.

Moreover, note also that the surrogate relaxation that includes bounding constraints as component constraints is a surrogate relaxation of the one that excludes these bounding constraints. In general, suppose we define

$$\begin{aligned}
(\mathrm{P}) \quad & \max\{x_0 = cx : Ax \leqslant A^0,\ Bx \leqslant B^0,\ x \in X\},\\
S(u) \quad & \max\{x_0 = cx : uAx \leqslant uA^0,\ Bx \leqslant B^0,\ x \in X\},\\
S(v) \quad & \max\{x_0 = cx : Ax \leqslant A^0,\ vBx \leqslant vB^0,\ x \in X\},\\
S(u,v) \quad & \max\{x_0 = cx : uAx + vBx \leqslant uA^0 + vB^0,\ x \in X\}.
\end{aligned}$$

Then the problems $S(u)$, $S(v)$ and $S(u,v)$ are surrogate relaxations of P and $S(u,v)$ is a surrogate relaxation of the problems $S(u)$ and $S(v)$. Defining $S(u^*) = \min\{S(u) : u \geqslant 0\}$, $S(v^*) = \min\{S(v) : v \geqslant 0\}$ and $S = \min\{S(u,v) : u, v \geqslant 0\}$, then we have $S(u^*) \leqslant S(u^*, v)$ for all $v \geqslant 0$, $S(v^*) \leqslant S(u, v^*)$ for all $u \geqslant 0$, and $\max(S(u^*), S(v^*)) \leqslant S$.

Illustration of Observation 1. The LP relaxation of the surrogate problem $S(\mu)$ is

$$\text{LP-S}(\mu) \quad \max\{x_0 = cx : \mathrm{d}x \leqslant d^0 \text{ and } 0 \leqslant x \leqslant e\}.$$

We order the variables in descending *bang-per-buck* order, i.e., in descending order of the ratios of the objective function coefficients to the surrogate constraint coefficients. Then the solution to the LP relaxation of the surrogate problem occurs by sequentially setting the variables equal to 1, until reaching the point where the residual portion of the surrogate constraint RHS compels a fractional or 0 value to be assigned to the next variable (or where no more variables remain). More formally, the variables are ordered according the ratio $r_j = \frac{c_j}{d^j} \geqslant \frac{c_{j+1}}{d^{j+1}}$. An optimal solution $\bar{x}$ of the LP relaxation of the surrogate problem LP-S(μ) is obtained explicitly by

$$\bar{x}_j = 1 \quad \text{for } j = 1, \ldots, j^* - 1,$$

$$\bar{x}_{j^*} = \frac{d^0 - \sum_{k=1}^{j^*-1} d^k}{d^{j^*}}, \quad \bar{x}_j = 0 \quad \text{for } j = j^* + 1, \ldots, n,$$

$$\text{where } j^* = \min\left\{ j : \left(d^0 - \sum_{k=1}^{j} d^k \right) \geqslant 0 \right\}.$$

The resulting objective function value is $x_o = c\bar{x}$, giving an upper bound on the optimum x_o value for 0–1 solutions. In addition, suppose we have a feasible solution x^* to the original problem. The objective function value, cx^*, is a lower bound on the optimum x_o value. This solution is of course feasible for the surrogate constraint (2). To create the mixed surrogate constraint which combines (2) and (3), we choose the weight for (2) that is the same weight it receives in the LP dual solution to the surrogate relaxation (knapsack) problem $S(\mu)$. This weight is identified by pivoting on the variable in the surrogate constraint that received a fractional value in the LP solution. (In the absence of any variables with fractional values, the pivot can be on the last variable that receives a unit value or the first variable that receives a 0 value.) Let x_{j^*} be the variable giving the pivot element, and thus the dual weight is r_{j^*}. This weight is the bank-for-buck ratio for x_{j^*}, and it is also the multiple of (2) that would be subtracted from the objective function by a pivot operation to create the updated objective function. The coefficients of the resulting updated objective function are the negative of the reduced costs. Consequently, we weight (2) by r_{j^*} and add the result to (3) to create the mixed surrogate constraint $\pi x \leqslant \pi_o$ with

$$\pi = r_{j*}d - c \quad \text{and} \quad \pi_o = r_{j*}d^0 - cx^*. \tag{4'}$$

In fact, in the preceding calculation, if the surrogate constraint (2) had been obtained by weighting the original problem constraints by their associated dual values in the LP relaxation of this problem, then the surrogate constraint would already be a multiple of r_{j^*} times the version of the constraint depicted as (4). Then it would not be necessary to identify the dual weight for (2) by a pivot calculation, since the weight would automatically be 1 (i.e., the "dual LP form" of (2) would simply be added to (3) to give (4)).

By our preceding comments, the coefficients of the mixed surrogate constraint (4) are the same as the reduced costs in the LP solution. In accordance with the usual application of the bounded variable simplex method, a negative reduced cost identifies a variable that must be set equal to its upper bound to identify the LP solution. If, in contrast to the prescription of Observation 1, we had included weights for the simple bounding inequalities, the mixed surrogate constraint (4) would have 0 coefficients for each of the variables that appears with a negative reduced cost. Such an outcome creates a loss of useful information for bounding the variables, and also for generating nested inequality constraints from the surrogate constraint.

To put the mixed constraint (4) into the standard non-negative coefficient format, we set $y_j = 1 - x_j$ to complement the appropriate variables. More precisely, let π^-, π^+ denote the associated vectors defined by ${}^-\pi_j^+ = \max\{\pi_j, 0\}$, $\pi_j^- = \min\{\pi_j, 0\}$. The mixed constraint (4) can be disaggregated as follows:

$$\pi x = \pi^- x + \pi^+ x = \pi^-(e - y) + \pi^+ x \leqslant \pi_o.$$

We can also complement the variables even though it has a 0 coefficient, for example the variables that are set equal to 1 in the knapsack LP solution, giving

$$-\pi^{-}y + \pi^{+}x \leqslant \pi_{o} - \pi^{-}e. \tag{5}$$

This complementation does not uncover additional implications at this point, but it proves relevant to other more advanced analysis, as will subsequently be shown.

The mixed surrogate constraint (5) is the customary "variable fixing inequality" for zero–one problems. The variable x_j is fixed to 0 if the corresponding coefficient π_j^+ is greater than the value $\pi_o - \pi^- e$ and the variable x_j is fixed to 1 if the absolute value of the coefficient π_j^- is greater than the value $\pi_o - \pi^- e$. Evidently, the ability to use this inequality to fix x_j variables to 1 (by fixing the associated y_j variables to 0) would not be possible if the simple bounding constraints had been included as component constraints. Still more critically, Observation 1 affects the generation of nested inequalities – both by reference to the mixed surrogate constraint (5) and by reference to its component surrogate constraint (2). This has a bearing on our next observation.

Example A. Consider the following surrogate relaxation of a zero–one MDK:

$$\max \quad 40x_1 + 49x_2 + 24x_3 + 36x_4 + 40x_5 + 30x_6 + 32x_7 + 16x_8 + 27x_9 + 9x_{10} \tag{A1}$$

$$5x_1 + 7x_2 + 4x_3 + 6x_4 + 8x_5 + 6x_6 + 8x_7 + 4x_8 + 9x_9 + 3x_{10} \leqslant 33 \tag{A2}$$

$$x_j \in \{0,1\} \quad \text{for } j = 1,\ldots,10.$$

The LP surrogate solution in this case is

$$x_1 = x_2 = x_3 = x_4 = x_5 = 1, \quad x_6 = 1/2, \quad x_7 = x_8 = x_9 = x_{10} = 0.$$

The resulting objective function value is $x_o = 204$, giving an upper bound on the optimum x_o value for 0–1 solutions. In addition, suppose we have a feasible solution to the original problem given by

$$x_1 = x_2 = x_3 = x_4 = x_5 = x_{10} = 1, \quad \text{all other variables 0.}$$

The objective function value, $x_o = 198$, is a lower bound on the optimum x_o value, and the associated objective function constraint, to compel x_o to be better than 198, is given by

$$40x_1 + 49x_2 + 24x_3 + 36x_4 + 40x_5 + 30x_6 + 32x_7 + 16x_8 + 27x_9 + 9x_{10} \geqslant 199. \tag{A3}$$

We write the foregoing inequality as a "$\leqslant$ constraint" to give it the same orientation as the surrogate constraint (A2).

$$-40x_1 - 49x_2 - 24x_3 - 36x_4 - 40x_5 - 30x_6 - 32x_7 - 16x_8 - 27x_9 - 9x_{10} \leqslant -199. \tag{A3$'$}$$

The mixed surrogate constraint combines (A2) and (A3′).

The weight for (A2) is identified by pivoting on the variable in the surrogate constraint that received a fractional value in the LP solution. Thus, x_6 is the variable giving the pivot element, and the dual weight is 5. Consequently, we weight (A2) by 5 and add the result to (A3′) to create the mixed surrogate constraint:

$$-15x_1 - 14x_2 - 4x_3 - 6x_4 + 0x_5 + 0x_6 + 8x_7 + 4x_8 + 18x_9 + 6x_{10} \leqslant -34. \tag{A4}$$

To put (A4) into the standard non-negative coefficient format, we set $y_j = 1 - x_j$ to complement the appropriate variables, giving

$$15y_1 + 14y_2 + 4y_3 + 6y_4 + 0y_5 + 0x_6 + 8x_7 + 4x_8 + 18x_9 + 6x_{10} \leqslant 5. \tag{A5}$$

We have complemented x_5 even though it has a 0 coefficient because it is one of the variables set equal to 1 in the knapsack LP solution.

Example B. Consider the example of Osorio et al. with 15 variables and 4 knapsack constraints whose data are presented in Table 1.

The optimal value of the LP-relaxation of this problem is equal to 335.62 and an optimal dual vector is

$$u^*(\mathrm{LP}) = (335.62, 0.66, 0.52, 0.62, 2.78).$$

Table 1
Data set of Example B

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
c_j	36	83	59	71	43	67	23	52	93	25	67	89	60	47	64	A^0
A_1^j	7	19	30	22	30	44	11	21	35	14	29	18	3	36	42	87
A_2^j	3	5	7	35	24	31	25	37	35	25	40	21	7	17	22	75
A_3^j	20	33	17	45	12	21	20	2	7	17	21	11	11	9	21	65
A_4^j	15	17	9	11	5	5	12	21	17	10	5	13	9	7	13	55

An optimal solution of this LP-relaxation and an initial feasible solution, denoted by $\bar{x}$ and x^*, respectively, are given below with their associated cost:

$$\bar{x} = (0, 0.72, 0.49, 0, 0, 0, 0, 0, 0.89, 0, 0.22, 1, 1, 0, 0), \quad c\bar{x} = 335.62,$$
$$x^* = (0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0), \quad cx^* = 301.$$

The reduced cost vector π in the LP solution, which corresponds to the coefficients of the mixed surrogate constraint (4) is

$$\pi = (24.41, 0, 0, 20.47, 10.65, 5.11, 43.21, 40.89, 0, 35.73, 0, \mathbf{-23.13}, \mathbf{-22.44}, 10.62, 24.36).$$

If we had included weights for the simple bounding inequalities as in Osorio et al., the mixed surrogate constraint (4) would have 0 coefficients for each of the variables that appears with a negative reduced cost (in bold).

3. Valid inequalities

Valid inequalities are potentially useful in solving (mixed) integer programs, and are often derived from knapsack constraints. The well-known "covering inequalities," for example, which are based on simple knapsack constraint implications, have been used extensively in the literature. Knapsack constraints are also a key modeling structure in constraint programming. Crowder et al. (1983) used a thorough understanding of individual knapsacks to solve general integer programs.

In general, we may regard the knapsack problem as a special case of the MKP where $m = 1$. Let $N = \{1, \ldots, n\}$ and assume that the right-hand side a_0 and the vectors c and a are non-negative integer. The knapsack problem (KP) can be formulated as follows:

$$\text{(KP)} \quad \max\{x_0 = cx \text{ subject to } ax \leqslant a_0 \text{ and } x \in \{0,1\}^n\}.$$

We call a set C a cover or a dependent set with respect to N if $\sum_{j\in C} a_j > a_0$. A cover C is minimal if $\sum_{j\in S} a_j \leqslant a_0$ for all subsets $S \subset C$. If we choose all elements from the cover C, it is clear that the following knapsack cover inequality $\sum_{j\in C} x_j \leqslant |C| - 1$ is valid (Glover, 1971; Balas, 1975; Hammer et al., 1975; Wolsey, 1975).

It is easy to identify the rule to generate the upper bound on the sum of all variables, we simply sum the coefficients of the vector a, proceeding from the smallest a_j to the largest. Suppose the coefficients of the knapsack constraint $ax \leqslant a_0$ are already ordered that way, i.e.,

$$a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n. \tag{6-a}$$

Let $\delta_k = \sum_{j=1}^{k} a_j = \delta_{k-1} + a_k$, starting from $\delta_1 = a_1$. Then we keep adding coefficients until reaching a point where $\delta_k \leqslant a_o$ and $\delta_{k+1} > a_o$. This is exactly the same rule that would be used if all coefficients were non-negative, simply by complementing the variables, and evidently implies that the upper bound on the sum of all variables is given by

$$ex = \sum_{j\in N} x_j \leqslant k = \max\{j : \delta_j \leqslant a_o\}. \tag{6-b}$$

Cover Cut Procedure: //upper bound on sum of all variables

Input: knapsack constraint $ax \leqslant a_0$.
Output: cover constraint $ex \leqslant k$.
Step 1: Sort the coefficients of the knapsack constraint such that $a_j \leqslant a_{j+1}$ for $j = 1$ to $n - 1$.
Step 2: Let $\delta_0 = 0$ and for $j = 1$ to n do $\delta_j = \delta_{j-1} + a_j$. Generate the cut $ex \leqslant k$.

Consequently, in our Example A, where $N = \{1, \ldots, 10\}$, the value of k is 8, and hence the inequality bounding the sum of all variables is

$$ex \leqslant 8.$$

Another very straightforward observation is useful to illustrate connections between continuous and integer solutions that support the forgoing derivations.

Observation 2. The upper bound k on the sum of all variables is equal to the optimum value of the following knapsack problem:

$$\text{(KP)} \quad \max\{x_0 = ex \text{ subject to } ax \leqslant a_0 \text{ and } x \in \{0,1\}^n\}$$

and this value derives by rounding the LP solution to the continuous version of (KP).

Illustration of Observation 2. Consider the LP relaxation (LP-KP) obtained from (KP) by removing the integrality constraints on the variables:

$$\text{LP-KP} \quad \max\{x_0 = ex \text{ subject to } ax \leqslant a_0 \text{ and } 0 \leqslant x \leqslant e\}.$$

Assume the variables are ordered in descending order of the ratios of the objective function coefficients to the knapsack constraint coefficients, i.e., so that

$$\frac{1}{a^1} \geqslant \frac{1}{a^2} \geqslant \cdots \geqslant \frac{1}{a^n}. \tag{6-c}$$

Observe that the sort (6-c) is equivalent to the sort (6-a). Hence, an optimal solution of the problem LP-KP occurs by sequentially setting the variables equal to 1, until reaching the point where the residual portion of the knapsack constraint RHS compels a fractional or 0 value to be assigned to the next variable (or where no more variables remain). More formally, an optimal solution $\bar{x}$ of the LP relaxation LP-KP is obtained explicitly by

$$\bar{x}_j = 1 \quad \text{for } j = 1, \ldots, j^* - 1,$$
$$\bar{x}_{j*} = \frac{a_0 - \delta_{j*-1}}{a_{j*}}, \quad \bar{x}_j = 0 \quad \text{for } j = j^* + 1, \ldots, n,$$
$$\text{where } j^* = \max\{j : \delta_j \leqslant a_o\}.$$

The objective function value of the LP-relaxation LP-KP is a upper bound on the optimum value of the knapsack problem, i.e., $v(\text{KP}) \leqslant e\bar{x}$, where $v(\text{KP})$ is the optimal value of the knapsack problem (KP). Since all the objective function coefficients are integer, the following constraint is also valid:

$$v(\text{KP}) \leqslant \lfloor e\bar{x} \rfloor. \tag{6-d}$$

The optimum solution $\bar{x}$ of the LP relaxation problem LP-KP has at most one fractional variable $\bar{x}_{j*}$, so by setting this variable to zero, we obtain a feasible solution x^* of the knapsack problem (KP) such that $ex^* = j^* - 1$. It is clear that $\lfloor e\bar{x} \rfloor = ex^* = k$. Thus, from (6-d) we have $v(\text{KP}) = k$.

4. Additional valid inequalities

We now examine considerations that are no less fundamental, but that are perhaps less immediate.

Observation 3. Consider a system consisting of a set of problem constraints and a mixed surrogate constraint, together with its components, augmented by a set of nested inequalities generated from the mixed surrogate constraint. Then additional strengthening of the system can be obtained by incorporating two additional sets of nested inequalities generated by reference to the components of the mixed surrogate constraint (i.e., where one is derived from the component surrogate constraint and one is derived from the x_o constraint).

Observation 3 results from the fact that the two additional sets of nested inequalities can create nesting sequences that differ from each other and that also differ from the sequence produced by the mixed surrogate constraint. Moreover, the two nested inequality sets "pull in opposite directions." Thus, for example, in the multidimensional knapsack problem the objective function constraint generates "$\geqslant$" nested inequalities while the surrogate constraint generates "$\leqslant$" nested inequalities. The mixed surrogate constraint generates inequalities that are implicitly a mix of the implications of the other inequalities.

Illustration of Observation 3. The relevance of Observation 3 is quickly illustrated by the fact that the surrogate constraint (A2) and the objective function constraint (A3), respectively, imply $ex \leqslant 6$ and $ex \geqslant 6$, while the mixed constraint (A4) implies $3 \leqslant ex \leqslant 7$. Hence, the inequalities $ex \leqslant 6$ and $ex \geqslant 6$, members of the nested inequalities from each of the component constraints, dominate the associated inequality $3 \leqslant ex \leqslant 7$ obtained from the system for the mixed surrogate constraint. (This is true even though our illustration uses the stronger form of (A4) that results by applying Observation 1. If Observation 1 were not applied, (A4) would not have implied $ex \geqslant 3$.)

Moreover, if we had not been fortunate enough to know a very good feasible solution to the problem (which gives the good lower bound for x_o used in this example), the mixed constraint would be still weaker, while the surrogate constraint (A2) would be unaffected. For example, suppose the best feasible solution known was the one that sets x_1 to $x_5 = 1$, and the remaining variables to 0. (This is the one that results by rounding down the fractional variable in the LP solution.) Then the RHS for (A4) would be –26, and thus the mixed surrogate constraint would only yield $2 \leqslant ex \leqslant 8$, whereas the surrogate constraint (A2) and the objective function constraint (A3) would respectively yield $ex \leqslant 6$ and $ex \geqslant 4$. Given that the nested inequalities provide a primary source of improvement for solving hard problems, these differences are noteworthy.

Consider the two binary integer programs (BP^+) and (BP^-) which consist of maximizing and minimizing respectively the sum of the variables subject to two constraints, where one is the component surrogate constraint and one is the objective function constraint. The problems (BP^+) and (BP^-) are stated as follows:

$$(BP^+) \quad \max\{x_0 = ex : ax \leqslant a_0,\ cx \geqslant c_0,\ x \in \{0,1\}^n\},$$
$$(BP^-) \quad \min\{x_0 = ex : ax \leqslant a_0,\ cx \geqslant c_0,\ x \in \{0,1\}^n\}.$$

The mixed surrogate constraint, as previously indicated, is a surrogate constraint created by combining a given surrogate constraint with an objective function constraint. After rewriting the objective function constraint as a "$\leqslant$" constraint to give it the same orientation as the surrogate constraint, and after choosing non-negative weights α and β for the two constraints, we obtain the following surrogate relaxation problems:

$$(S^+(\alpha,\beta)) \quad \max\{x_0 = ex : \alpha ax - \beta cx \leqslant \alpha a_0 - \beta c_0,\ x \in \{0,1\}^n\},$$
$$(S^-(\alpha,\beta)) \quad \min\{x_0 = ex : \alpha ax - \beta cx \leqslant \alpha a_0 - \beta c_0,\ x \in \{0,1\}^n\}.$$

As the surrogate functions $v(S^+(\alpha,\beta))$ and $v(S^-(\alpha,\beta))$ are homogeneous functions over R^2_+, we can restrict the search domain over a compact set, for example, by using the norm L_1, the surrogate functions to be considered are $v(S^+(\alpha,(1-\alpha)))$ and $v(S^-(\alpha,(1-\alpha)))$ for $\alpha \in [0,1]$. Moreover, since the surrogate function $v(S^+(\alpha,\beta))$ is a quasi-convex function, thus for any $\alpha \in [0,1]$, we have $v(S^+(\alpha,(1-\alpha))) \leqslant \max\{v(S^+(1,0)), v(S^+(0,1))\}$ where

$$S^+(1,0)\ \max\{x_0 = ex : ax \leqslant a_0,\ x \in \{0,1\}^n\} \text{ and } S^+(0,1)\ \max\{x_0 = ex : cx \geqslant c_0,\ x \in \{0,1\}^n\}.$$

The surrogate function $v(S^-(\alpha,\beta))$ is a quasi-concave function, so we have

$$\min\{v(S^-(1,0)),\ v(S^-(0,1))\} \leqslant v(S^-(\alpha,(1-\alpha))) \quad \text{for any } \alpha \in [0,1].$$

In summary, for $\alpha \in [0,1]$, we have

$$\min\{v(S^-(1,0)),\ v(S^-(0,1))\} \leqslant v(S^-(\alpha,(1-\alpha))) \leqslant v(\mathrm{BP}^-) \leqslant ex \quad \text{and} \quad ex \leqslant v(\mathrm{BP}^+) \leqslant v(S^+(\alpha,(1-\alpha))) \leqslant \max\{v(S^+(1,0)),\ v(S^+(0,1))\}.$$

The above illustration shows the relevance of Observation 3. One way to improve the bounds on the sum of the variables is to solve the corresponding duals of the above relaxations. More precisely we have

$$v(S^-) \leqslant v(\mathrm{BP}^-) \leqslant ex \leqslant v(\mathrm{BP}^+) \leqslant v(S^+),$$

where

$$(S^+)\ \min\{v(S^+(\alpha,(1-\alpha))) : \alpha \in [0,1]\} \text{ and } (S^-)\ \min\{v(S^-(\alpha,(1-\alpha))) : \alpha \in [0,1]\}.$$

To solve these dual problems we can use one of the algorithms proposed by Glover (1965), Karwan and Rardin (1984), Fréville and Plateau (1993), and Hanafi (1993). For the multidimensional knapsack (MDK) problem where the right-hand sides a_0 and c_0 and the vectors a and c are non-negative, in spite of the trivial optimal solutions 0 and e for the surrogate problems $S^-(1,0)$ and $S^+(0,1)$, (i.e., $v(S^-(1,0)) = 0.0 = 0$ and $v(S^+(0,1)) = ee = n$), we do not necessarily have $v(\mathrm{BP}^+)$ equals to $v(S^+(1,0))$.

Example C. Consider the following surrogate relaxation of a zero–one MDK:

$$\begin{array}{lll} (\mathrm{BP}^+) & \max & x_1+x_2+x_3+x_4+x_5+x_6+x_7+x_8+x_9+x_{10} \\ & \text{s.t.} & x_1+2x_2+3x_3+4x_4+5x_5+6x_6+7x_7+8x_8+9x_9+10x_{10} \leqslant 12, \\ & & 2x_1+2x_2+2x_3+2x_4+10x_5+10x_6+10x_7+10x_8+10x_9+10x_{10} \geqslant 21, \\ & & x_j \in \{0,1\} \quad \text{for } j=1,\ldots,10. \end{array}$$

We have $v(\mathrm{BP}^+) = 3$, $v(S^+(1,0)) = 4$ and $v(S^+(0,1)) = 10$.

5. Nested valid inequalities

Valid inequalities are called *Nested Cuts* when two inequalities overlap in their unit coefficients only if the non-zero coefficients of one are contained in the other. More precisely, let N^k, $k = 1,\ldots,K$, denote a collection of distinct non-empty subsets of N, the subsets N^k are called nested sets if they satisfy the property

$$\text{For all } k,\ k' \in \{1,\ldots,K\}, \quad (k \neq k' \text{ and } N^k \cap N^{k'} \neq \emptyset) \ \Rightarrow\ (N^k \subset N^{k'} \text{ or } N^{k'} \subset N^k).$$

Let N be the index set of variables in the constraint $ax \leqslant a_0$. As noted, the cover cut procedure generates the valid inequality $\sum_{j\in N} x_j \leqslant \max\{j : \sum_{j\in N} a_j \leqslant a_o\}$. For each subset N' of N, we consider the constraint $a'x \leqslant a_0$ where the component $a'_j = a_j$, if j in N' and 0 otherwise. By using this constraint we can generate new valid inequalities corresponding to upper bounds on sums of variables in N'. The valid inequalities on partial sums of variables in N^k are called nested inequalities if the subsets N^k are nested subsets.

Let $X^k = (X^k_1, X^k_2, \ldots, X^k_n)$ denote a zero–one characteristic vector associated with the subset N^k, which is defined by $X^k_j = 1$ if j is in N^k, 0 otherwise. The nested property is equivalent to specifying that variables X^k satisfy

$$\text{For all } p,q \text{ in } N, \quad (p \neq q) \text{ and } X^p X^q \geqslant 1 \ \Rightarrow\ (X^p \geqslant X^q \text{ or } X^q \geqslant X^p).$$

5.1. Contiguous inequalities

The simple types of nested inequalities where each is strictly "contained in" the next member of the progression, are called *contiguous cuts*. Specifically, the contiguous cuts with associated subsets N^k, $k = 1,\ldots,K$, satisfy the property $N^1 \supset N^2 \cdots \supset N^k$.

Observation 4. It is possible to take account of dominance considerations by a simple check applied to consecutive contiguous cuts to reduce the collection of nested cuts generated.

Illustration of Observation 4. Let N be the index set of variables in the source constraint $ax \leqslant a_0$. Two sets N and N' are called adjacent sets if they differ only by a single element, i.e., $N' = N + \{j^0\}$. Define the vector a' so that $a'_j = a_j$ for $j \neq j^0$ and $a'_{j^0} = 0$, and consider the corresponding constraint $a'x \leqslant a_0$. Note that this latter constraint is a relaxation of the source constraint and the non-negativity constraint. According to Observation 2 if the coefficients are already ordered so that $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$, we have

$$\sum_{j\in N} x_j \leqslant k = \max\left\{j : \delta_j = \sum_{i=1}^{j} a_i \leqslant a_o\right\}, \tag{7-a}$$

$$\sum_{j\in N'} x_j \leqslant k' = \max\left\{j : \delta'_j = \sum_{i=1}^{j} a'_i \leqslant a_o\right\}. \tag{7-b}$$

It is easy to show that if $k < j^0$ then $\delta'_j = \delta_j$ for $j \leqslant k$, then the constraint (7-a) dominates the constraint (7-b). Otherwise (i.e., $j^0 \leqslant k$), if the condition $(\delta_{k+1} - a_{j^0} \leqslant a_0)$ is satisfied then we have $\delta'_{k+1} \leqslant a_o$ and $\delta'_{k+2} > a_o$ so the constraint (7-a) again dominates the constraint (7-b). In the case $(\delta_{k+1} - a_{j^0} > a_0)$ we have $\delta'_k \leqslant a_o$ and $\delta'_{k+1} > a_o$ which imply that $\sum_{j\in N'} x_j \leqslant k-1$. This latter constraint (7-b) combined with the upper bound on x_{j^0} imply the constraint (7-a). This proves that only one of two adjacent nested cuts need be kept.

Osorio et al. (2002) propose an algorithm as a special case of an approach of Glover (1971) for generating contiguous cuts $N^k = \{k, k+1, \ldots, n\}$ for a 0–1 inequality $ax \geqslant a_0$. It is assumed, that the coefficients are already ordered so that $a_1 \geqslant a_2 \geqslant \cdots \geqslant a_n$.

Contiguous nested cuts procedure

```
Let δ_0 = 0 and for j = 1 to n do δ_j = δ_{j−1} + a_j;
Let k = 1; k_last = 0;
For j = 1 to n do
   if (δ_n − a_0 < δ_j − δ_{k−1}){
      while (δ_n − a_0 < δ_j − δ_k)   k++;
      if (k > k_last) {
         generate the cut Σ_{i=1}^{j} x_i ⩾ k
         k_last = k;
      }
   }
```

Using the dominance between two consecutive contiguous cuts, we propose the following procedure. In this procedure we introduce a new variable called *j_last* to generate only the non-dominate cuts.

Improved contiguous nested cuts procedure

```
Let δ_0 = 0 and for j = 1 to n do δ_j = δ_{j−1} + a_j;
Let k = 1; k_last = 0; j_last = −1;
For j = 1 to n do
   if (δ_n − a_0 < δ_j − δ_{k−1}){
      while (δ_n − a_0 < δ_j − δ_k)   k++;
      if (k > k_last) {
         if (j_last + 1 < j) generate the cut Σ_{i=1}^{j} x_i ⩾ k;
         k_last = k;   j_last = j;
      }
   }
```

Example D. Consider the following knapsack constraint:

$$95x_1 + 92x_2 + 87x_3 + 80x_4 + 78x_5 + 72x_6 + 61x_7 + 54x_8 + 52x_9 + 30x_{10} \leqslant 467.$$

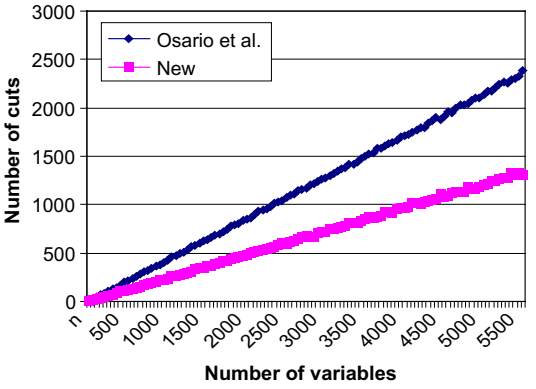

Fig. 1. Comparison of the two procedures for generating nested cuts.

The contiguous nested cuts procedure generates the following six cuts:

$$x1 + x2 + x3 \leqslant 1,$$

$$x1 + x2 + x3 + x4 \leqslant 2,$$

$$x1 + x2 + x3 + x4 + x5 \leqslant 3,$$

$$x1 + x2 + x3 + x4 + x5 + x6 + x7 \leqslant 4,$$

$$x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 \leqslant 5,$$

$$x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 \leqslant 6.$$

Our improved contiguous nested cuts procedure generates only two non-dominated cuts:

$$x1 + x2 + x3 \leqslant 1,$$

$$x1 + x2 + x3 + x4 + x5 + x6 + x7 \leqslant 4.$$

Fig. 1 shows the progression of the number of nested cuts generated by the two procedures as a function of the number of variables. The coefficients of the source constraint are generated randomly by taking $a_0 = \alpha ae$ with α close to 0.5.

5.2. Mixed nested inequalities

Observation 5. Different nested inequalities are produced by using different forms of the mixed surrogate constraint, where different sets of coefficients are selected to be negative. Moreover, the nested inequalities generated directly from the form of the mixed surrogate that does not complement the problem variables includes all of those generated in the Osorio et al. paper, plus additional nested inequalities, thus producing a system that dominates the system previously obtained. Finally, this expanded system can be generated with the same computer code used to generate the previous smaller system.

Observation 5 is important for the harder problems where the nested inequalities are the major contribution to improving the solution process.

Illustration of Observation 5. We show that the nested sum inequalities obtained from the mixed surrogate constraint in the form that has both negative and positive coefficients include all of those generated in Osorio et al., and also include others.

Write the mixed surrogate constraint that includes the negative coefficients in the form

$$\sum(\pi_j x_j : j \varepsilon N^*) + \sum(\pi_j x_j : j \varepsilon N - N^*) \leqslant \pi_o,$$

where N^* is the index set for the negative coefficients. The previous approach replaced the coefficients $\pi_j; j\varepsilon N^*$ with 0's to generate nested inequalities from the source inequality

$$\sum(0x_j : j\varepsilon N^*) + \sum(\pi_j x_j : j\varepsilon N - N^*) \leqslant \pi_o^*, \tag{8-a}$$

where $\pi_o^* = \pi_o - \sum(\pi_j : j\varepsilon N^*)$.

The first nested inequality from this $\leqslant$ source inequality is an "overall inequality"

$$\sum(x_j : j\varepsilon N) \leqslant \mathrm{RHS}(N).$$

The new nested inequalities that are omitted in Osorio et al. (2002) are those that involve partial sums over $j\varepsilon N^*$ of the following form:

$$\sum(x_j : j\varepsilon N^*(1)) \geqslant \mathrm{RHS}(1)$$
$$\sum(x_j : j\varepsilon N^*(2)) \geqslant \mathrm{RHS}(2)$$
$$\sum(x_j : j\varepsilon N^*(3)) \geqslant \mathrm{RHS}(3)$$
$$\sum(x_j : j\varepsilon N^*(4)) \geqslant \mathrm{RHS}(4), \text{ etc.}$$

Here, $N^*(1) = N^*$, and in turn $N^*(2)$ removes the index for the smallest absolute value coefficient associated with $N^*(1)$, then $N^*(3)$ removes the index for the smallest absolute value coefficient associated with $N^*(2)$, and so on.

It is easy to identify the rule to generate these nested inequalities directly, but they can also be generated using the rule already applied to generate nested inequalities from the $\leqslant$ source inequality, simply by complementing the variables. The first step begins with the source:

$$\sum(\pi_j x_j : j\varepsilon N) \leqslant \pi_o,$$

which is implied by the original mixed surrogate constraint. Then we complement the variables ($y_j = 1 - x_j$) for $j\varepsilon N^*$ to obtain the modified source

$$\sum(\pi_j^* y_j : j\varepsilon N^*) + \sum(\pi_j x_j : j\varepsilon N - N^*) \leqslant \pi_o^*, \tag{8-b}$$

where $\pi_j^* = -\pi_j > 0$ and, as before, $\pi_o^* = \pi_o - \sum(\pi_j : j\varepsilon N^*)$.

This inequality can also be obtained from the source inequality (8-a) used in Osorio et al. that drops the negative coefficients. Recall that this inequality is

$$\sum(0x_j : j\varepsilon N^*) + \sum(\pi_j x_j : j\varepsilon N - N^*) \leqslant \pi_o^{**}. \tag{8-c}$$

Hence, in Example A, where $\pi_o^* = \pi_o - \sum(\pi_j : j\varepsilon N^*). = -5 - (-11) = 6$, the inequality (8-a) is given by

$$0x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 2x_7 + 2x_8 + 3x_9 + 4x_{10} \leqslant 6. \tag{8-d}$$

It is easy to see that the upper bound on the sum of all variables is exactly the same as given above. In that the present case this inequality dominates all other nested inequalities from the source (8-a) used in Osorio et al. until reaching the subsets of variables whose coefficients are positive – i.e., in (8-d) it dominates all nested inequalities until reaching those whose index sets are $\{8,9,10\}$, $\{9,10\}$ and $\{10\}$. (It dominates the inequality over the indexes $\{7,8,9,10\}$ because this has the same right-hand side k as the bound on all the variables.) It is naturally important to include this inequality on the sum of all variables among the nested inequalities, although it is not in general true that the inequality will dominate a string of successive inequalities as in the present example.

5.2.1. Inequalities missing from the earlier implementation

To generate the $\geqslant$ inequalities that are missing from the Osorio et al. implementation, we start from the source inequality (8-d), and consider only the negative coefficients. Thus, (8-d) and (1-c) or (1-d) imply the following constraint:

$$\pi^{-}(e-x) \geqslant \pi^{-}e - \pi_{o}. \tag{8-e}$$

Clearly, this inequality is implied by (8-e), and it is the "missing part" of the Osorio et al. development.

The new inequalities that are also missing from the Osorio et al. implementation, can be obtained directly from the source inequality (8-d), where we consider negative and positive coefficients. Recall that both inequalities (8-c) and (8-e) are derived from (8-d), and that (8-d) is stronger than (8-c) or (8-e).

5.2.2. General nested cuts

Assume that the vectors π^{-} and π^{+} can be decomposed as follows: $\pi^{-} = \pi_1^{-} + \pi_2^{-}$ and $\pi^{+} = \pi_1^{+} + \pi_2^{+}$. Then the source inequality (4) can be rewritten as

$$\pi_1^{-}x + \pi_2^{-}x + \pi_1^{+}x + \pi_2^{+}x \leqslant \pi_{o}.$$

This latter constraint can in turn be rewritten as

$$-\pi_1^{-}(e-x) + \pi_2^{-}x + \pi_1^{+}x + \pi_2^{+}x \leqslant \pi_{o} - \pi_1^{-}e. \tag{8-f}$$

From the inequality (8-f) we can derive different new relaxations of this constraint combined with the original constraint such as (1-c) or (1-d). This combination can provide the following source constraints:

$$\pi_2^{-}x + \pi^{+}x \leqslant \pi_{o} - \pi_1^{-}e, \tag{8-g}$$
$$\pi_2^{-}x + \pi_1^{+}x \leqslant \pi_{o} - \pi_1^{-}e, \tag{8-h}$$
$$\pi^{-}x + \pi_1^{+}x \leqslant \pi_{o}. \tag{8-i}$$

Remarks

(1) In the constraints (8-g:i) we can interchange π_1^{-} with π_2^{-} and/or π_1^{+} with π_2^{+}.
(2) Osario et al. considered only the case (8-g) with $\pi_2^{-} = 0$.

Example B. To give a numerical example, we start with the mixed inequality, in the form of (4):

$$-4x_1 - 3x_2 - 2x_3 - 2x_4 + 0x_5 + 0x_6 + 2x_7 + 2x_8 + 3x_9 + 4x_{10} \leqslant -5. \tag{B1}$$

The inequality (8-c), which drops the negative coefficients, is given by

$$0x_1 + 0x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 2x_7 + 2x_8 + 3x_9 + 4x_{10} \leqslant 6. \tag{B2}$$

It is easy to see that the $\leqslant$ nested inequalities that have already been generated from the source (B2) in the Osorio et al. implementation, are

$$x_9 + x_{10} \leqslant 1, \tag{B3a}$$
$$x_7 + x_8 + x_9 + x_{10} \leqslant 2. \tag{B3b}$$

The "missing part" of the Osorio et al. development are the $\geqslant$ nested inequalities derived from (8-d), which corresponds to the following inequality:

$$4x_1 + 3x_2 + 2x_3 + 2x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} \geqslant 5. \tag{B4}$$

Using the preceding procedure with the source constraint (B4) gives rise to the inequalities

$$x_1 + x_2 \geqslant 1, \tag{B5a}$$
$$x_1 + x_2 + x_3 + x_4 \geqslant 2. \tag{B5b}$$

The new $\leqslant$ and $\geqslant$ nested inequalities are derived directly from the source (B1) by complementing the variables with negative coefficients, to give

$$x_1 \geqslant x_{10}, \tag{B6a}$$
$$x_1 \geqslant x_9 + x_{10}, \tag{B6b}$$
$$x_1 + x_2 + x_4 \geqslant 1 + x_9 + x_{10}, \tag{B6c}$$
$$x_1 + x_2 + x_3 + x_4 \geqslant 2 + x_9 + x_{10}, \tag{B6d}$$
$$x_1 + x_2 + x_3 + x_4 \geqslant 1 + x_7 + x_8 + x_9 + x_{10}. \tag{B6e}$$

Note that the inequalities (B3a) and (B5b) are implied by the inequalities (B6b) and (B6d), respectively. We also observe that the nested inequalities (B6) can dominate both of the nested constraints (B3) and (B5) if all the coefficients of the source constraint (4) are different, since, after complementation, several variables in the transformed source constraint have the same coefficient. To illustrate, in order to use the procedure directly, we transform the source constraint (B1′) into a $\geqslant$ constraint with only positive coefficients as follows:

$$4x_1 + 3x_2 + 2x_3 + 2x_4 + 0x_5 + 0x_6 + 2(1 - x_7) + 2(1 - x_8) + 3(1 - x_9) + 4(1 - x_{10}) \geqslant 16. \tag{B1′}$$

Considering all the orderings of the variables having the same coefficients, we can also generate the new nested constraints:

$$x_1 + x_2 \geqslant 1 + x_{10}, \tag{B6f}$$
$$x_1 + x_2 + x_3 \geqslant 1 + x_9 + x_{10}, \tag{B6g}$$
$$x_1 + x_2 + x_3 + (1 - x_7) \geqslant 1 + x_9 + x_{10}, \tag{B6h}$$
$$x_1 + x_2 + x_3 + (1 - -x_8) \geqslant 1 + x_9 + x_{10}, \tag{B6i}$$
$$x_1 + x_2 + x_3 + (1 - x_7) \geqslant 1 + x_7 + x_8 + x_9 + x_{10}, \tag{B6j}$$
$$x_1 + x_2 + x_3 + (1 - x_8) \geqslant 1 + x_7 + x_8 + x_9 + x_{10}. \tag{B6k}$$

The collection of the nested constraints (B6) dominates the nested constraints (B3) and (B5).

References

Balas, E., 1975. Facets of the knapsack polytope. Mathematical Programming 8, 146–164.

Crowder, H., Johnson, E., Padberg, M., 1983. Solving large-scale zero–one linear programming problems. Operations Research 31, 803–834.

Fischer, M.L., 1981. The Lagrangean relaxation method for solving integer programming problems. Management Sciences 27, 1–18.

Fréville, A., 2004. The multidimensional 0–1 knapsack problem: An overview, invited review. European Journal of Operational Research 155, 1–21.

Fréville, A., Hanafi, S., 2005. The multidimensional 0–1 knapsack problem – Bounds and computational aspects. Annals of Operations Research 139 (1), 195–227 (33).

Fréville, A., Plateau, G., 1993. An exact search for the solution of the surrogate dual of the 0–1 bidimensional knapsack problem. European Journal of Operational Research 68, 413–421.

Geoffrion, A.M., 1974. The Lagrangean relaxation for integer programming. Mathematical Programming Study 2, 82–114.

Glover, F., 1965. A multiphase-dual algorithm for the zero–one integer programming problem. Operations Research 13, 879–919.

Glover, F., 1968. Surrogate constraints. Operations Research 16, 741–749.

Glover, F., 1971. Flows in arborescences. Management Science 17, 568–586.

Greenberg, H., Pierskalla, W., 1970. Surrogate mathematical programs. Operations Research 18, 924–939.

Hammer, P.L., Johnson, E.L., Peled, U.N., 1975. Facets of regular 0–1 polytopes. Mathematical Programming 8, 179–206.

Hanafi, S., 1993. Contribution à la résolution de problèmes duaux de grande taille en optimisation combinatoire, PhD thesis, University of Valenciennes, France.

Karwan, M.H., Rardin, R.L., 1984. Surrogate dual multiplier search procedures in integer programming. Operations Research 32, 52–69.

Osorio, M.A., Gómez, E., 2004. Cutting analysis for MKP. In: Ricardo Baeza-Yates, J., Luis Marroquín, M., Edgar Chávez, C. (Eds.), Proceedings of the Fifth Mexican International Conference on Computer Science. IEEE Computer Society, Silver Spring, MD, ISBN 0-7695-2160-6, pp. 298–303.

Osorio, M.A., Glover, F., Hammer, P., 2002. Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions. Annals of Operations Research 117, 71–93.

Wolsey, L.A., 1975. Faces for a linear inequality in 0–1 variables. Mathematical Programming 8, 165–178.