Math. Program., Ser. B (2015) 154:55–80
DOI 10.1007/s10107-015-0902-5

FULL LENGTH PAPER

Sequence independent lifting for mixed knapsack problems with GUB constraints

Alejandro Angulo[1] · **Daniel Espinoza**[1] · **Rodrigo Palma**[2]

Received: 7 May 2014 / Accepted: 19 March 2015 / Published online: 12 April 2015

Abstract In this paper, we consider the semi-continuous knapsack problem with generalized upper bound constraints on binary variables. We prove that generalized flow cover inequalities are valid in this setting and, under mild assumptions, are facet-defining inequalities for the entire problem. We then focus on simultaneous lifting of pairs of variables. The associated lifting problem naturally induces multidimensional lifting functions, and we prove that a simple relaxation in a restricted domain is a superadditive function. Furthermore, we also prove that this approximation is, under extra requirements, the optimal lifting function. We then analyze the separation problem in two phases. First, finding a seed inequality, and second, select the inequality to be added. In the first step we evaluate both exact and heuristic methods. The second step is necessary because the proposed lifting procedure is simultaneous; from where our class of lifted inequalities might contain an exponential number of these. We choose a strategy of maximizing the resulting violation. Finally, we test this class of inequalities using instances arising from electrical planning problems. Our tests show that the proposed class of inequalities is strong in the sense that the addition of these inequalities closes, on average, 57.70 % of the root integrality gap and 97.70 % of the relative gap while adding less than three cuts on average.

Keywords Knapsack problem · sequence-independent multidimensional lifting · Generalized upper bounds

This research was funded by FONDECYT Grant 1110024 and Millennium Nucleus Information and Coordination in Networks ICM/FIC RC13003.

✉ Daniel Espinoza
daespino@dii.uchile.cl

1 Department of Industrial Engineering, Universidad de Chile, Santiago, Chile

2 Department of Electrical Engineering, Universidad de Chile, Santiago, Chile

Mathematics Subject Classification 90C11 · 90C57 · 65K05 · 49M30

1 Introduction

Binary knapsack programs are a common model for choosing between discrete alternatives. If the choice is continuous but limited; the resulting model is called a classical single node flow set, as studied in [17,19]. If the choice is semi-continuous; we must consider mixed-binary knapsack programs. This problem is known as the semi-continuous knapsack problem (SCKP). If binary variables are subjected to independent clique constraints we have what we call a semi-continuous knapsack problem with generalized upper bound constraints (SCKPGUB). This kind of model is common for representing (possibly non-linear) functions (with only one GUB constraint), or for studying the combined non-linear output of several machines.

This is the case in production scheduling problems, such as in electricity generation [6,9,18,26], where the cost function of each generating unit can be highly non-linear and even discontinuous. Furthermore, if the original problem has integer variables; these functions are usually approximated by a piece-wise linear function.

In this setting, sequential lifting is too limited in the sense that whenever we have a constraint $y \leq x$ for $x \in \{0, 1\}$ and $y \in [0, 1]$, lifting must be carried out first on the integer variable and then on the continuous variable. This precludes finding some facet-defining inequalities for the complete problem; making simultaneous lifting essential in this setting. For a basic reference on simultaneous lifting, see [13,27], and see [8,11,15] for some experiments and results regarding sequential and simultaneous lifting.

In this paper, we use generalized flow cover (GFC) inequalities [21]; show that they are valid in our setting and, under mild assumptions, they induce facets or high dimensional faces of the original problem. We then propose a valid sequence-independent multidimensional lifting scheme to obtain valid inequalities for SCKPGUB. We show that the proposed lifting function is superadditive on a restricted set of *feasible* right-hand sides, and show that this condition is sufficient to obtain sequence-independent lifting. Finally, we also provide sufficient conditions for this lifting to be maximal.

The paper is organized as follows. Section 2 covers some known facts about semi-continuous knapsack problems; including a class of valid inequalities and basic results for the semi-continuous knapsack problem with GUB constraints. Section 3 deals with multidimensional lifting for SCKPGUB; specifically on how to obtain valid sequence-independent lifting functions for them. We also propose simple algorithms to solve the separation problem both for the seed inequality and for the selection of maximally violated lifted inequalities. In Sect. 4 we propose a heuristic separation algorithm that provides good numerical results and shows that the lifting step is crucial. We generate our test-instances from problems in electricity generation where some of the parameters are randomly perturbed. Finally, we present our conclusions and explore further research questions in Sect. 5.

2 Definitions and basic polyhedral results

In this section we introduce most of the notation used throughout the paper. This includes the precise definition of the polytope with which we will work. We also present some previously known results and state some basic polyhedral results. Finally, we prove that our seed inequality, under mild hypothesis, is also a facet.

2.1 The problem

We consider the semi-continuous knapsack problem with generalized upper bound constraints given by

$$X_G = \left\{ \begin{array}{rl} (x, y) \in \{\{0,1\} \times [0,1]\}^M : & \\ \sum_{k \in M} (a_k x_k + m_k y_k) \leq b & \\ y_k \leq x_k & \forall k \in M \\ \sum_{k \in M_g} x_k \leq 1 & \forall g \in G \end{array} \right\}, \tag{1}$$

where G is the set of GUB constraints, for each $g \in G, n_g \in \mathbb{N}$ is the number of elements in the GUB constraint indexed by $g, n = \sum_{g \in G} n_g, M = \{(g, j) : g \in G, j \in 1, \ldots, n_g\}$, and $M_g = \{(g', j') \in M : g' = g\}$. This implies that each $k \in M$ is an ordered pair (g, j); we will use both notations interchangeably. Also, to simplify notation, when we have a vector $\mu \in \mathbb{R}^r$ (for some $r \in \mathbb{N}$) and $R \subseteq \{1, \ldots, r\}$, we will denote $\mu(R) := \sum \mu_i : i \in R$. Note that $a_k x_k + m_k y_k$ is a model for a semi-continuous variable with values in $\{0\} \cup [a_k, a_k + m_k]$. Although a_k and m_k could in general be negative, we will focus on the case where both quantities are non-negative, i.e., $a_k, m_k \geq 0, \forall k \in M$. Note that unlike in the classical knapsack case, this assumption is restrictive, but we choose it nonetheless because it follows what happens in many applications. The first constraint is the semi-continuous knapsack constraint, the second constraint ensures semi-continuity, and the third constraint imposes a generalized upper bound condition among disjoint sets of binary variables.

2.2 Literature review

Several special subsets of this structure have been studied before. For example, the classical binary knapsack problem (KP) was studied by Balas and Jeroslow [4] in a theoretical study where canonical cuts on the unit hypercube were introduced. Based on this work, in 1975, Wolsey [24] and Balas [3] presented facet-defining inequalities for the KP by using the notion of cover for the first time. Hammer, Johnson and Peled [12] also studied facets of regular 0–1 polytopes, which include knapsack problems. This study also characterizes every non-trivial facet with 0–1 coefficients. In 1978, Balas and Zemel [5] extended previous work by applying lifting procedures to valid inequalities obtained from minimal covers. In 1980, Padberg [20] presented $(1, k)$-configurations as a generalization of minimal cover inequalities. Johnson and Padberg [14] studied the inclusion of GUB constraints in the KP (KPGUB); they also showed how to transform a general instance of the problem into one with only non-negative coefficients. In 1988, Wolsey [23] defined some valid inequalities for the KPGUB and proved that they are facet-defining under certain conditions. Sherali and Lee [22] applied sequential and simultaneous lifting to valid inequalities for KPGUB deduced from minimal covers.

Another special case is when $a_k = 0$ and $|M_g| = 1$. This case is called single-node flow sets (SNFS), and their study has been extended from the work of Gu et al. [10] from lifting procedures applied to this set. In 2007, Louveaux and Wolsey [16] provided a survey of strong valid inequalities for knapsack and single-node flow sets.

As can be seen, the application of lifting procedures is a fundamental part of cut generation techniques for many specific sets. In 1977, Wolsey [25] presented the first work in this area and used the concept of superadditivity. In 2000, Gu et al. [11] generalized it and defined sequence-independent lifting of general mixed integer programs. In 2004, Atamtürk [2] presented similar results.

The above research can be seen as concerned with one-dimensional lifting, since all of these studies consider the perturbation of only one constraint. Applications of multidimensional lifting are scarce, with work by Zeng [28] and Zeng and Richard [29,30] as the most relevant. They defined a general framework to derive multidimensional and superadditive lifting functions and applied it to the precedence-constrained knapsack problem and the single-node flow set. They showed that the traditional concept of superadditivity used by Gu et al. [11] can be restricted depending on the problem at hand. We provide a simple proof of this result in the context of SCKPGUB.

2.3 Polyhedral results

2.3.1 Basic results for SCKPGUB

We will henceforth assume that $\bar{a} := \max\{a_k : k \in M\} < b$. With this in place, Proposition 1 follows:

Proposition 1 *1.* *X_G is full-dimensional.*
2. *Inequality $y_k \geq 0$ is facet-defining for $X_G, \forall k \in M$.*
3. *If $a_k + m_k \leq b$, the inequality $y_k \leq x_k$ is facet-defining for $X_G, \forall k \in M$.*
4. *If $\overline{a}_g := \max\limits_{k \in M \setminus M_g} \{a_k\} + \min\limits_{k \in M_g} \{a_k\} < b$, then $\sum_{k \in M_g} x_k \leq 1$ is facet defining for X_G, for $g \in G$.*

Proof The basic idea of the proof is to find the appropriate number of feasible affinely-independent points satisfying each inequality at equality. For details see "Proof of Proposition 1". □

2.3.2 Generalized flow cover inequalities for SCKP

Consider the set

$$X = \left\{ \begin{array}{l} (x, y) \in \{0,1\}^n \times [0,1]^n : \\ \qquad \sum_{j \in M} (a_k x_k + m_k y_k) \leq b \\ \qquad\qquad\qquad y_k \leq x_k \qquad \forall k \in M \end{array} \right\}. \tag{2}$$

Van Roy and Wolsey [21] studied (a generalization of) this polyhedron and proposed a family of valid inequalities that they called generalized flow cover (GFC) inequalities. In our setting, we restate this family of inequalities as follows:

Given X as defined in (2), we call a pair (C, C_U), with $C \subset M$ and $C_U \subset C$, satisfying $\Gamma := a(C) + m(C_U) - b > 0$ and $m(C_U) > 0$, a *generalized cover*. Then the inequality

$$\sum_{j\in C} \min\left\{1, \frac{\xi_j}{\Gamma}\right\}(x_j - 1) + \sum_{j\in C_U} \frac{m_j}{\Gamma}\left(y_j - x_j\right) \leq -1, \qquad (3)$$

where $\xi_j = a_j$ for $j \in C \backslash C_U$ and $\xi_j = a_j + m_j$ for $j \in C_U$, is valid for X.

Theorem 1 gives sufficient conditions for (3) to be facet-defining for X.

Theorem 1 *Let (C, C_U) be a generalized flow cover satisfying $\sum_{j\in C_U:\xi_j>\Gamma} m_j > \Gamma$. Then, inequality (3) is facet-defining for $X_o := X \cap \{x_i = 0, \forall i \notin C\}$.*

Proof Van Roy and Wolsey [21] proved the validity of inequality (3). We prove that (3) is facet-defining for $X_o := X \cap \{x_i = 0, i \notin C\}$ by constructing a set of $2s$ affinely independent points in X_o satisfying it at equality. For details see "Proof of Theorem 1". □

Note that X is a face of X_G where we choose at most one element from every GUB constraint to be active. Given this, Theorem 1 ensures that whenever $\sum_{j\in C_U:\xi_j>\Gamma} m_j > \Gamma$, the resulting flow cover inequality is facet-defining for this face of X_G. Moreover, since X is also a relaxation of X_G, (3) defines valid inequalities for X_G.

3 Multidimensional lifting for SCKPGUB

In this section we deal with the problem of lifting our seed inequality, defined in (3). To achieve this, we first define what is a *valid lifting function* in a setting that allows simultaneous and multidimensional lifting in general, and apply it to our particular set. We also identify simple conditions under which optimal lifting coefficients are zero; and introduce a *superadditive* approximation for maximal lifting functions. Since the full separation of the seed inequality is $\mathcal{NP}$-hard, we propose a heuristic algorithm to find seed inequalities, and describe a simple algorithm to apply our sequence-independent lifting function.

3.1 Valid lifting functions

In this section we study the following problem: given a polytope

$$P = \left\{\begin{array}{ll} x \in \mathbb{R}^n : & Ax \leq b, \\ & 0 \leq x \leq u, \\ & x_i \in \mathbb{Z}, \forall i \in I \end{array}\right\},$$

where $I \subseteq \{1, \ldots, n\}$; a set $N \subseteq \{1, \ldots, n\}$, $N^c = \{1, \ldots, n\} \setminus N$; and an inequality $ax \leq b_o$ valid for P, satisfying $a_i = 0$, $\forall i \in N$; we want to find $\alpha \in \mathbb{R}^n$ satisfying $\alpha_i = 0$, $\forall i \notin N$ such that $ax + \alpha x \leq b_o$ is a valid inequality (and hopefully tight) for P, i.e.

$$ax + \alpha x \leq b_o, \ \forall x \in P$$

Taking advantage of the condition that $a_i \cdot \alpha_i = 0$, $\forall i \in \{1, \ldots, n\}$, we can think that $x \in \mathbb{R}^n$ has two independent components $x = (x_{N^c}, x_N) = (v, w)$ and that $P = \{(v, w) \in \mathbb{R}^n : A_1 v + A_2 w \leq \bar{b}, v \in V, \ w \in W\}$, where V, W describe the corresponding box-constraints, integrality requirements, and inequalities involving variables indexed by N^c and N respectively. Abusing notation, we can re-state our problem as finding α such that

$$av + \alpha w \leq b_o, \quad \forall A_1 v \leq \bar{b} - A_2 w, \ v \in V, \ w \in W.$$

Here, we can make a second observation; the interaction between v and w is only through the common inequalities $A_1 v \leq \bar{b} - A_2 w$; and there, the interaction is only through values of $A_2 w : w \in W$ for which there exists $v \in V$ satisfying $A_1 v + A_2 w \leq \bar{b}$. Moreover, the problem of ensuring the condition for all feasible points in P is an optimization problem in a different space. Formally, we state this problem as follows:

$$\overbrace{\begin{array}{rl} \max & \alpha w \\ s.t. & A_2 w = z \\ & w \in W \end{array}}^{h_\alpha(z)} \leq \overbrace{\begin{array}{rl} \min & b_o - ax \\ s.t. & A_1 v \leq \bar{b} - z \\ & v \in V \end{array}}^{f(z)} \quad \forall z \in Z,$$

where $Z = \{z : \exists v \in V, \ w \in W, \ z = A_2 w, \ A_1 v + z \leq \bar{b}\}$. With these definitions, our problem can be simply stated as finding α such that

$$h_\alpha(z) \leq f(z), \ \forall z \in Z.$$

Note that in the literature, usually $Z \subseteq \mathbb{R}$; however, in our set, $Z \subseteq \mathbb{R} \times \{0, 1\}^G$, and the elements in Z will have a continuous component z and a binary vector component $\boldsymbol{v} \in \{0, 1\}^G$. In this sense, our lifting is multidimensional. Also, since at every step N will have two variables, we will be doing simultaneous lifting.

To apply these ideas iteratively, in this section, we re-write (3) as

$$\sum_{i \in C} \gamma_j x_j + \sum_{j \in C_U} \frac{m_j}{\Gamma} \left(y_j - x_j \right) \leq \gamma(C) - 1, \tag{4}$$

where $\gamma_j = \min\{1, \frac{\xi_j}{\Gamma}\}$. As noted before, given a *generalized flow cover* C, C_U in X_G satisfying $|C \cap \{k : k \in M_g\}| \leq 1, \forall g \in G$; (4) is valid for X_G and, if $m(C_U^+) > \Gamma$, where $C_U^+ = \{k \in C_U : a_k + m_k > \Gamma\}$, (4) is a facet-defining inequality for $X_G \cap \{x_i =$

$0, \forall i \notin C\}$. Following Gu et al. [11], we consider the problem of sequentially lifting pairs of variables $(x_k, y_k),\ k \notin C$ to obtain

$$\sum_{i\in C}\gamma_j x_j + \sum_{j\in C_U}\frac{m_j}{\Gamma}\left(y_j - x_j\right) + \sum_{k\notin C}(\alpha_k x_k + \beta_k y_k) \le \gamma(C) - 1. \tag{5}$$

For simplicity we index pairs $\{k_i\}_{i=1}^{n-|C|} = \{(g_i, j_i)\}_{i=1}^{n-|C|} = M\backslash C$ and assume that the first $i-1$ pairs of variables have been lifted. With this, the i-th lifting function reduces to

$$\begin{aligned} h_{k_i}(z, \boldsymbol{v}) = \max\ & \alpha_{k_i} x_{k_i} + \beta_{k_i} y_{k_i} \\ s.t.\ & a_{k_i} x_{k_i} + m_{k_i} y_{k_i} = z \\ & x_{k_i} = \boldsymbol{v_{g_i}} \\ & 0 \le y_{k_i} \le x_{k_i} \\ & x_{k_i} \in \{0, 1\}, \end{aligned} \tag{6}$$

and $f_{k_i}(z, \boldsymbol{v})$ reduces to

$$\begin{aligned} f_{k_i}(z, \boldsymbol{v}) = \min & \sum_{i\in C}\gamma_j\left(1 - x_j\right) - \sum_{j\in C_U}\frac{m_j}{\Gamma}\left(y_j - x_j\right) \\ & - \sum_{k\in K^i}(\alpha_k x_k + \beta_k y_k) - 1 \\ & s.t. \sum_{k\in C\cup K^i}(a_k x_k + m_k y_k) \le b - z \\ & \sum_{k\in M_g(C\cup K^i)} x_k \le 1 - \boldsymbol{v}_g, \quad \forall g \in G \\ & 0 \le y_k \le x_k,\ x_k \in \{0, 1\} \quad \forall k \in C \cup K^i, \end{aligned} \tag{7}$$

where $K^i = \{k_1, \dots, k_{i-1}\}, z \in [0, b]$, and $\boldsymbol{v}$ has the dimension of the right-hand sides of X_G for GUB constraints; and define $\boldsymbol{v_{g'}} = \delta_{g',g_i}$ for $g \in G$; where $\delta_{a,b} = 1$ if $a = b$, and zero otherwise, i.e., $\boldsymbol{v} = e_{g_i}$. Our objective is to find $\alpha_{k_i}, \beta_{k_i}$ that ensure that $h_{k_i}(z, u\boldsymbol{v}) \le f_{k_i}(z, u\boldsymbol{v})$ for all $(z, u) \in \{(0, 0)\} \cup \{([a_{k_i}, a_{k_i} + m_{k_i}], 1)\}$ and $\boldsymbol{v} = e_{g_i}$. This implies that we are not interested in h_{k_i}, f_{k_i} for all possible $(z, \boldsymbol{v}) \in \mathbb{R} \times \{0, 1\}^{G\cup M}$, but only in the true domain of feasible points of X_G.

Although the lifting is multidimensional, there are only two degrees of freedom at each step in the functions, namely z and $u \in \{0, 1\}$. The analysis of h_{k_i} is easy because for the case where $m_{k_i} > 0$ the optimal value of $h_{k_i}(z, \boldsymbol{v})$ is

$$h_{k_i}(z, \boldsymbol{v}) = \begin{cases} 0 & u = 0, z = 0 \\ \tilde{\alpha} + \tilde{\beta} z & u = 1, a_{k_i} \le z \le a_{k_i} + m_{k_i} \end{cases}$$

where $\tilde{\alpha} = \alpha_{k_i} - \frac{a_{k_i}}{m_{k_i}}\beta_{k_i}$ and $\tilde{\beta} = \frac{1}{m_{k_i}}\beta_{k_i}$.

For the case where $m_{k_i} = 0$, the optimal value of the function is

$$h_{k_i}(z, \boldsymbol{v}) = \begin{cases} 0 & u = 0, z = 0 \\ \tilde{\alpha} & u = 1, z = a_{k_i} \end{cases}$$

where $\tilde{\alpha} = \alpha_{k_i}$ and $\tilde{\beta} = \beta_{k_i} = 0$. We call $\tilde{\alpha}$ and $\tilde{\beta}$ *normalized* lifting coefficients.

To study f_{k_i}, we start with a simple case in the following proposition:

Proposition 2 *Let* $D = \{(g, j) \in M \backslash C : \exists (g, j') \in C,\ \xi_{gj'} \geq \Gamma,\ a_{gj} + m_{gj} \leq \xi_{gj'} - \Gamma\}$. *Then, for all* $k \in D$, *the maximal lifting coefficients* (α_k, β_k) *are* $(0, 0)$.

Proof Since $f_{k_i} \leq f_{k_{i+1}}$; we obtain the best possible lifting coefficients for these variables when we lift them first. We will prove that even in this best case these coefficients are zero. So, we assume that the first elements to lift from the seed inequality are from D. Let k be an element in D. It is known that $f_k(z, \boldsymbol{v}) \geq 0$ is a monotonic, non-decreasing function, and that $f(0, \mathbf{0}) = 0$. This implies that it is enough to find a feasible point for $z = a_k + m_k$ with an objective value equal to zero to prove our result. Let $k_o \in C$ satisfying $\xi_{k_o} > \Gamma$ and $a_k + m_k \leq \xi_{k_o} - \Gamma$. Then, setting $(x, y) = (\mathbf{1}_C - e_{k_o}, \mathbf{1}_C - e_{k_o})$, we obtain $f_k(z, \boldsymbol{v}) \leq 0$ for $z \in [a_k, a_k + m_k]$, $\boldsymbol{v} = e_{g_k}$. □

Note that Theorem 1 ensures that there is nothing to be gained from lifting x and/or y in C; whereas Proposition 2 ensures the same for variables in D. The following proposition will allow us to assume that it is enough to consider the case when $D = \emptyset$ and where $m(C_L) = 0$, where $C_L := C \backslash C_U$.

Proposition 3 *If* $k \in M \backslash (C \cup D)$, *then for every optimal solution of the problem* $f_k(z, \boldsymbol{v})$, *it is always possible to find an optimal solution* x^*, y^* *satisfying* $y_k^* = 0$ *for* $k \in C_L$ *and* $x_k^* = y_k^* = 0$ *for* $k \in D$.

Proof Let y^*, x^* be an optimal solution to $f_k(z, \boldsymbol{v})$. Note that if x^*, y^* is valid for (7), then for any $j \in M$, changing any x_j^*, y_j^* to zero maintain feasibility. Thus, we only need to prove that by making this change for $j \in D$ and $y_j^*,\ j \in C_L$, the objective function will not deteriorate.

For $j \in D$, the optimal lifting coefficients $(\alpha_j, \beta_j) = (0, 0)$. Then, any valid lifting coefficient pairs $\alpha_j x_j^* + \beta_j y_j^* \leq 0$. Replacing x_j^*, y_j^* with $(0, 0)$ will then not deteriorate the objective function; thus proving that there exists an optimal solution with these variables set to zero.

For $j \in C_L$ the argument is similar: if $k = 1$, note that the objective function has a zero coefficient for y_j, from where $\beta_j \leq 0$. Using the fact that the lifting functions will be decreasing and that the coefficients in the objective function accompanying y_j, for all $j \in C_L$, are non-positive; we conclude that we can always find an optimal solution with $y_j^* = 0$ for $j \in C_L$. □

These two propositions allow us to work with the assumption that $D = \emptyset$ and that $m(C_L) = 0$. This is because Proposition 2 ensures that the best possible lifting coefficients are zero; while Proposition 3 ensures that, if $D \neq \emptyset$, there exists an optimal solution for f_{k_i} with $x_i, y_i = 0$ for all $i \in D$.

3.2 A lower bound for lifting functions

Given $\boldsymbol{v} \in \{0,1\}^G$, and defining $C_{\boldsymbol{v}} = \{(g,j) \in C : \boldsymbol{v}_g = 1\}$, we can re-write the first lifting function $f(z_1, \boldsymbol{v})$ as

$$\begin{aligned} f_1(z,\boldsymbol{v}) = \gamma(C) - 1 - \max & \sum_{k \in C \setminus C_{\boldsymbol{v}}} \left(\gamma_k x_k + \frac{m_k}{\Gamma}(y_k - x_k) \right) \\ s.t. \quad & \sum_{k \in C \setminus C_{\boldsymbol{v}}} (\xi_k x_k + m_k(y_k - x_k)) \leq b - z \\ & 0 \leq y_k \leq x_k,\ x_k \in \{0,1\}\ \forall k \in C \setminus C_{\boldsymbol{v}}. \end{aligned} \tag{8}$$

In general, $f_1(z, \boldsymbol{v})$ is a complex function, and a simpler functional form $\tilde{f}$ is needed satisfying $\tilde{f} \leq f_1$. We propose the following relaxation of f_1: first note that $b = \xi(C) - \Gamma$; replace x by $1 - x$ and y by $x - y$; and we obtain the following equivalent form for f_1:

$$\begin{aligned} f_1(z,\boldsymbol{v}) = \gamma(C_v) - 1 + \min & \sum_{k \in C \setminus C_{\boldsymbol{v}}} \left(\gamma_k x_k + \frac{m_k}{\Gamma} y_k \right) \\ s.t. \quad & \sum_{k \in C \setminus C_{\boldsymbol{v}}} (\xi_k x_k + m_k y_k) \geq z + \Gamma - \xi(C_v) \\ & 0 \leq y_k \leq x_k,\ x_k \in \{0,1\}\ \forall k \in C \setminus C_{\boldsymbol{v}}. \end{aligned}$$

Now we define $C^+ = \{k \in C : \xi_k > \Gamma\}$, $s = \sum_{k \in (C \setminus C^+) \setminus C_{\boldsymbol{v}}} (\xi_k x_k + m_k y_k) + \sum_{k \in C^+ \setminus C_{\boldsymbol{v}}} m_k y_k$, discard the inequality $y_k \leq x_k$ and the integrality condition of x_k for $k \in \left(C \setminus C^+\right) \setminus C_{\boldsymbol{v}}$ to obtain the following relaxation:

$$\begin{aligned} \tilde{f}(z,\boldsymbol{v}) = \gamma(C_{\boldsymbol{v}}) - 1 + \min & \sum_{k \in C^+ \setminus C_{\boldsymbol{v}}} x_k + \frac{s}{\Gamma} \\ s.t. \quad & \sum_{k \in C^+ \setminus C_{\boldsymbol{v}}} \xi_k x_k + s \geq z + \Gamma - \xi(C_{\boldsymbol{v}}) \\ & 0 \leq s,\ x_k \in \{0,1\},\ \forall k \in C^+ \setminus C_{\boldsymbol{v}}. \end{aligned} \tag{9}$$

Note that, under mild conditions,[1] $f_1(z, \boldsymbol{v})$ is equivalent to $\tilde{f}$.

Now, we will prove that $\tilde{f}$ has a closed form, and then we will prove that it is also superadditive in an appropriated domain.

Proposition 4 *By renaming $C^+ \setminus C_{\boldsymbol{v}} = \{1, \ldots, r_{\boldsymbol{v}}\}$ while ensuring that $\xi_h^{\boldsymbol{v}} \geq \xi_{h+1}^{\boldsymbol{v}}$, defining $\Lambda_h^{\boldsymbol{v}} = \xi(C_{\boldsymbol{v}}) + \sum_{i<h} \xi_i^{\boldsymbol{v}}$, and defining $H(z) = 0$ if $z \leq 0$ and $H(z) = 1$ if $z > 0$, we have*

$$\tilde{f}(z,\boldsymbol{v}) = \gamma(C_{\boldsymbol{v}}) - 1 + \frac{s^*}{\Gamma} + \sum_{h=1}^{r_{\boldsymbol{v}}} H\left(z - s^* - \Lambda_h^{\boldsymbol{v}} + \Gamma\right), \tag{10}$$

[1] A sufficient condition is that $m_k \geq \Gamma$ for the two smallest ξ_k coefficients in C.

where

$$s^* = \left(z - \Lambda^{\boldsymbol{v}}_{r_{\boldsymbol{v}}+1} + \Gamma\right) H\left(z - \Lambda^{\boldsymbol{v}}_{r_{\boldsymbol{v}}+1} + \Gamma\right) + \sum_{h=1}^{r_{\boldsymbol{v}}} \left(z - \Lambda^{\boldsymbol{v}}_h\right)\left(H(z - \Lambda^{\boldsymbol{v}}_h) - H(\Gamma + \Lambda^{\boldsymbol{v}}_h - z)\right).$$

Moreover, the optimal solution for x *is given by* $x^*_h = H(z - s^* - \Lambda^{\boldsymbol{v}}_h + \Gamma)$, $\forall h = 1, \ldots, r_{\boldsymbol{v}}$.

Proof Using the definition of $\tilde{f}(z, \boldsymbol{v})$ given in (9), if we consider $z \geq \Lambda^{\boldsymbol{v}}_{r_{\boldsymbol{v}}+1} - \Gamma$; the solution of the continuous relaxation is also integer-feasible (with $x^*_i = 1$ for all $i \in C^+ \backslash C_{\boldsymbol{v}}$); from where we have that $s^* = z - \Lambda^{\boldsymbol{v}}_{r_{\boldsymbol{v}}+1} + \Gamma$ and $z - s^* - \Lambda^{\boldsymbol{v}}_h + \Gamma = \Lambda^{\boldsymbol{v}}_{r_{\boldsymbol{v}}+1} - \Lambda^{\boldsymbol{v}}_h > 0$; thus proving our result for this case.

If we now consider $z < \Lambda^{\boldsymbol{v}}_{r_{\boldsymbol{v}}+1} - \Gamma$ and restrict ourselves to solutions where $s = 0$; the resulting problem has an optimal solution given by

$$x_h = H(z + \Gamma - \Lambda^{\boldsymbol{v}}_i), \quad \forall h = 1, \ldots, r_{\boldsymbol{v}},$$

whence (10) is directly obtained.

To compute $\tilde{f}(z, \boldsymbol{v})$, consider first a simpler optimization problem:

$$q(z) = \min_{s \geq 0} \left\{ g(z, s) := \frac{s}{b} + aH(z - s) \right\} \geq 0.$$

Note that if $s \geq ab$, then $g(z, s) \geq g(z, 0) = aH(z)$, thus proving that we can restrict ourselves to $s \in [0, ab]$. We now compute $q(z)$ by identifying three cases:

Case 1 If $z \leq 0$, we have

$$\left.\frac{s}{b} + aH(z - s)\right|_{s=0} = 0 < \left.\frac{s}{b} + aH(z - s)\right|_{0<s\leq ba} \quad \Rightarrow \quad q(z) = 0, \quad s^* = 0$$

Case 2 If $z > ba$, we have

$$\left.\frac{s}{b} + aH(z - s)\right|_{s=0} = a < \left.\frac{s}{b} + aH(z - s)\right|_{0<s\leq ba} \quad \Rightarrow \quad q(z) = a, \quad s^* = 0$$

because $z - s > 0$.

Case 3 If $0 < z \leq ba$, we can select $s^* = z$ and we have

$$\left.\frac{s}{b} + aH(z - s)\right|_{s=0} = a \geq \left.\frac{s}{b} + aH(z - s)\right|_{s=s^*} \quad \Rightarrow \quad q(z) = \frac{z}{b}, \quad s^* = z.$$

We can now write

$$\tilde{f}(z, \boldsymbol{v}) = \gamma(C_{\boldsymbol{v}}) - 1 + \min_{s \geq 0} \left\{ \frac{s}{\Gamma} + \sum_{h=1}^{r_{\boldsymbol{v}}} H\left(z - s - \Lambda^{\boldsymbol{v}}_h + \Gamma\right) \right\}, \tag{11}$$

Note that

$$\tilde{f}(z, \boldsymbol{v}) \geq \gamma(C_{\boldsymbol{v}}) - 1 + \min_{h=1}^{r_{\boldsymbol{v}}} \left\{ \min_{s \geq 0} \left\{ \frac{s}{\Gamma} + H(z - s - \Lambda_h^{\boldsymbol{v}} + \Gamma) \right\} \right\}. \tag{12}$$

Using the optimal solution and values computed for $q(z)$; we see that the optimal solution $\{s_h^*\}_{h=1}^{r_{\boldsymbol{v}}}$ for the lower bound (12) is either the all-zero vector, or it is the case that exactly one component, say h^*, is non-zero and is in the range $]0, \Gamma]$. In the first case, setting $s = 0$ in (11), we attain the lower bound, and thus solve the problem. In the second case, we have $0 \leq s^* = z - \Lambda_{h^*}^{\boldsymbol{v}} \leq \Gamma$. Since $\xi_h > \Gamma$, for $h \neq h^*$, $H(z - \Lambda_h^{\boldsymbol{v}} + \Gamma) = H(z - s_{h^*}^* - \Lambda_h^{\boldsymbol{v}} + \Gamma)$, thus proving that setting $s = s_h^*$ in (11) is a feasible solution that attains the lower bound. □

Theorem 2 *The function $\tilde{f}(z, \boldsymbol{v})$ is superadditive for $(z, \boldsymbol{v}) \in [0, +\infty) \times \{0, 1\}^G$.*

Proof Note that $\tilde{f}(z, \mathbf{0})$ is exactly the superadditive function defined in [10]. Namely,

$$\tilde{f}(z, \mathbf{0}) = \begin{cases} -1 & z \leq -\Gamma \\ (z - \Lambda_i)/\Gamma + i - 1 & \Lambda_i - \Gamma \leq z \leq \Lambda_i, \quad \forall i = 1, \ldots, r - 1 \\ i - 1 & \Lambda_i \leq z \leq \Lambda_{i+1} - \Gamma, \quad \forall i = 1, \ldots, r \\ (z - \Lambda_r)/\Gamma + r - 1 & \Lambda_r - \Gamma \leq z \leq b \end{cases}$$

We propose a different proof for this more general case.

We start with $\tilde{f}(z_1, \mathbf{0}) + \tilde{f}(z_2, \mathbf{0}) \leq \tilde{f}(z_1 + z_2, \mathbf{0})$:

Let $\{x_h^o\}_{h=1}^{r_{\boldsymbol{v}}}, s^o$ be the optimal solution of $\tilde{f}(z_1 + z_2, \mathbf{0})$, defined as in Proposition 4, and $\{x_h^1\}_{h=1}^{r_{\boldsymbol{v}}}, s^1$ be the optimal solution of $\tilde{f}(z_1, \mathbf{0})$. By construction, $x^o \geq x^1$, and assume that $h_o^*, h_1^* \in \{0, \ldots, r_{\boldsymbol{v}}\}$ are the last active elements in x^o, x^1 respectively.

We prove this by analyzing two cases:

Case a $\xi \cdot x^1 + s^1 = \Gamma + z_1$:

Define $x_h^2 = x_{h+h_1^*}^o$ for $h \leq h_o^* - h_1^*$, $x_h^2 = 0$ for $h > h_o^* - h_1^*$ and $0 \leq s^2 = \Gamma + s^o - s^1$. By construction and Proposition 4, we have $\tilde{f}(z_1 + z_2, \mathbf{0}) = \tilde{f}(z_1, \mathbf{0}) + \mathbb{I} \cdot x^2 + s^2/\Gamma - 1$, where $\mathbb{I}$ is the vector of all ones of the appropriate dimension. Thus, it is enough to prove that (x^2, s^2) is feasible for (9). However, by hypothesis, we have $\Gamma + z_2 \leq \xi(x^o - x^1) + (s^o - s^1) + \Gamma \leq \xi x^2 + s^2$, thus proving our result.

Case b $\xi \cdot x^1 + s^1 > \Gamma + z_1$:

In this case, by optimality, $s^1 = 0$ and $\xi x^1 = \Lambda_{h_1^*+1}^0$. Define $x_h^2 = x_{h+h_1^*-1}^o$ for $h \leq 1 + h_o^* - h_1^*$, $x_h^2 = 0$ for $h \geq 2 + h_o^* - h_1^*$ and $s^2 = s^o$. By construction, we have $\tilde{f}(z_1 + z_2, \mathbf{0}) = \tilde{f}(z_1, \mathbf{0}) + \mathbb{I} \cdot x^2 + s^2/\Gamma - 1$. Thus, is enough to prove that (x^2, s^2) is feasible for (9). However by hypothesis, we have $z_1 \geq \Lambda_{h_1^*}^0 - \Gamma$, whence $\Gamma + z_2 \leq \xi x^o + s^o - z_1 \leq \xi(x^o - x^1) + \xi_{h_1^*} + s^2 \leq \xi x^2 + s^2$, thus proving our result. This concludes Case 1.

The cases $\tilde{f}(z_1 + z_2, e_i)$ and $\tilde{f}(z_1 + z_2, e_i + e_j)$ are analogous. □

Corollary 1 *If, for each pair of variables* (x_k, y_k)*, where* $k = (g, j)$*, we choose lifting coefficients* (α_k, β_k) *such that* $h_k(z, u) \le \tilde{f}(z, u\boldsymbol{e_g})$ *for* $(z, u) \in \{([a_k, a_k + m_k], 1), (0, 0)\}$*; then the lifting process is sequence-independent.*

Proof We only need to prove validity at any intermediate step r, and call L^r the set of variables (not in C) lifted at step r. That is, we need to prove that

$$\begin{aligned}
\max \ & \sum_{k\in C} \left(\gamma_k x_k + \frac{m_k}{\Gamma}(y_k - x_k)\right) + \sum_{k\in L^r}^{r} (\alpha_r x_r + \beta_r y_r) \\
s.t. \ & \sum_{k\in C\cup L^r} \left(\xi_j x_j + m_j \left(y_j - x_j\right)\right) \le b \\
& \sum_{k\in M_g^r} x_k \le 1 && \forall g \in G \\
& 0 \le y_k \le x_k,\ x_k \in \{0, 1\} && \forall k \in C \cup L^r,
\end{aligned} \tag{13}$$

where $M_g^r = \{(g', j) \in C \cup L^r : g' = g\}$ is less than or equal to $\gamma(C) - 1$. Note that by hypothesis, $\alpha_k x_k + \beta_k y_k = h(\xi_k x_k + m_k(y_k - x_k), x_k) \le \tilde{f}(\xi_k x_k + m_k(y_k - x_k), x_k e_{g(k)})$, $\forall k \in L^r$. Using this, Eq. (13) can be re-written as

$$\begin{aligned}
\max \ & \sum_{k\in C} \left(\gamma_k x_k + \frac{m_k}{\Gamma}(y_k - x_k)\right) \\
& + \sum_{k\in L^r}^{r} \tilde{f}\left(\xi_k x_k + m_k(y_k - x_k), x_k g(k)\right) \\
s.t. \ & \sum_{k\in C\cup L^r} \left(\xi_j x_j + m_j \left(y_j - x_j\right)\right) \le b \\
& \sum_{k\in M_g^r} x_k \le 1 && \forall g \in G \\
& 0 \le y_k \le x_k,\ x_k \in \{0, 1\} && \forall k \in C \cup L^r,
\end{aligned} \tag{14}$$

where $g((g', j')) = g'$ for $k = (g' j')$. By defining $z_g = \sum_{k\in L^r} (\xi_k x_k + m_k (y_k - x_k))$, $\delta_g = \sum_{k\in L^r \cap M_g} x_k$, and $M_g = \{(g', j') \in M : g' = g\}$, we bound (14) above by the following expression:

$$\begin{aligned}
\max \ & \sum_{k\in C} \left(\gamma_k x_k + \frac{m_k}{\Gamma}(y_k - x_k)\right) \\
& + \sum_{g\in G}^{r} \tilde{f}\left(z_g, \delta_g e_g\right) \\
s.t. \ & \sum_{k\in C} \left(\xi_j x_j + m_j \left(y_j - x_j\right)\right) + \sum_{g\in G} z_g \le b \\
& \sum_{k\in M_g\cap C} x_k \le 1 - \delta_g && \forall g \in G \\
& 0 \le z_g \le b\delta_g,\ \delta_g \in \{0, 1\} && \forall g \in G \\
& 0 \le y_k \le x_k,\ x_k \in \{0, 1\} && \forall k \in C.
\end{aligned} \tag{15}$$

Note that by the definition of C, we have $|C \cap M_g| \in \{0, 1\}, \forall g \in G$. We set $G^o = \{g \in G : |C \cap M_g| = 0\}$, define $z_o = \sum_{g \in G^o} z_g$, and identify C with $G \backslash G^o$. With this, Eq. (15) is equivalent to

$$
\begin{aligned}
\max \ & \sum_{g \in G \backslash G^o} \left(\gamma_g x_g + m_g \left(y_g - x_g\right)\right) \\
& + \tilde{f}(z_o, 0) + \tilde{f}(z_g, \delta_g e_g) \\
s.t. \ & z_o + \sum_{g \in G \backslash G^o} \left(\xi_g x_g + m_g \left(y_g - x_g\right)\right) + z_g \le b \\
& x_g \le 1 - \delta_g && \forall g \in G \backslash G^o \\
& 0 \le z_g \le b\delta_g, \ \delta_g \in \{0, 1\} && \forall g \in G \backslash G^o \\
& 0 \le y_g \le x_g, \ x_g \in \{0, 1\} && \forall g \in G \backslash G^o \\
& 0 \le z_o \le b.
\end{aligned}
\tag{16}
$$

Using that $\delta \in \{0, 1\}^{G \backslash G^o}$ and the definition of $\tilde{f}(z, \delta)$; we can bound (16) as

$$
\begin{aligned}
\max \ & \gamma(\delta) - 1 - \tilde{f}\left(\sum_{g \in \{o\} \cup G \backslash G^o} z_g, \delta\right) \\
& + \tilde{f}(z_o, 0) + \sum_{g \in G \backslash G^o} \tilde{f}(z_g, \delta_g) \\
s.t. \ & \sum_{g \in \{o\} \cup G \backslash G^o} z_g \le b \\
& 0 \le z_g \le b\delta_g, \ \delta_g \in \{0, 1\} \qquad \forall g \in \{o\} \cup G \backslash G^o.
\end{aligned}
\tag{17}
$$

Finally, by using the superadditivity of $\tilde{f}$, we bound (17) by

$$
\max \left\{ \gamma(\delta) - 1 : \delta \in \{0, 1\}^{G \backslash G^o} \right\} = \Gamma(C) - 1. \tag{18}
$$

which proves our result. □

3.3 Algorithmic separation

The foregoing results show that we can use $\tilde{f}(z, \boldsymbol{v})$ to find valid lifting coefficients for GFC inequalities for SCKPGUB; and thus obtain strong inequalities for X_G.

In this subsection we deal with the separation problem of such lifted inequalities. More precisely, given x^* as a fractional solution in the linear programming (LP) relaxation of (1), we try to find a violated lifted constraint. We address this problem in two stages: we first show how to lift a candidate inequality, and then propose a heuristic to identify a candidate seed inequality. Finally, we only keep strictly violated inequalities.

3.3.1 Lifting GUB-constrained flow cover inequalities

It is important to note that for each pair of lifted variables $y_k, x_k, k \in M \backslash C$; there are several maximal pairs of coefficients α_k, β_k satisfying $h_k(z, x_k) \le \tilde{f}(z, x_k e_{g(k)})$.

Algorithm 1 Finding the lower envelope of $\tilde{f}(\cdot, \boldsymbol{v})$

Require: Breakpoints of $\tilde{f}(\cdot, \boldsymbol{v})$ and $B = \{z_i\}_{i=1}^m$ where $z_i \le z_{i+1}$ and $|B| \ge 2$. Interval $[a, b]$, actual range for z (we assume $z_1 \le a \le b \le z_n$).
Ensure: $\mathcal{H} = \{\tilde{\alpha}_j, \tilde{\beta}_j\}$, pairs of (normalized) maximal lifting coefficients.

$B_{[a,b]} \leftarrow \{a\} \cup \{z_i \in B : a < z_i < b\} \cup \{b\}$ (ordered set)
$n \leftarrow |B_{[a,b]}|, \mathcal{H} \leftarrow \emptyset, k_l \leftarrow 1, k_r \longleftarrow 2$
if $n = 1$ **then**
  $\mathcal{H} \leftarrow \{(\tilde{f}(a, \boldsymbol{v}), 0)\}$
  return $\mathcal{H}$
else
  loop
    $z_l \leftarrow B_{[a,b]}[k_l], f_l \leftarrow \tilde{f}(z_l, \boldsymbol{v}), z_r \leftarrow B_{[a,b]}[k_r], f_r \leftarrow \tilde{f}(z_r, \boldsymbol{v})$
    $\tilde{\beta} \leftarrow \frac{f_r - f_l}{z_r - z_l}, \tilde{\alpha} \leftarrow f_l - \tilde{\beta} z_l$
    if $k_r + 1 \le n$ **then**
      $z_{2r} \leftarrow B_{[a,b]}[k_r + 1], f_{2r} \leftarrow \tilde{f}(z_{2r}, \boldsymbol{v})$
      if $\tilde{\alpha} + \tilde{\beta} z_{2r} \le f_{2r}$ **then**
        $k_l \leftarrow k_r, k_r \leftarrow k_r + 1, \mathcal{H} \leftarrow \mathcal{H} \cup \{(\tilde{\alpha}, \tilde{\beta})\}$
      else
        $k_r \leftarrow k_r + 1$
    else
      $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\tilde{\alpha}, \tilde{\beta})\}$
      return $\mathcal{H}$

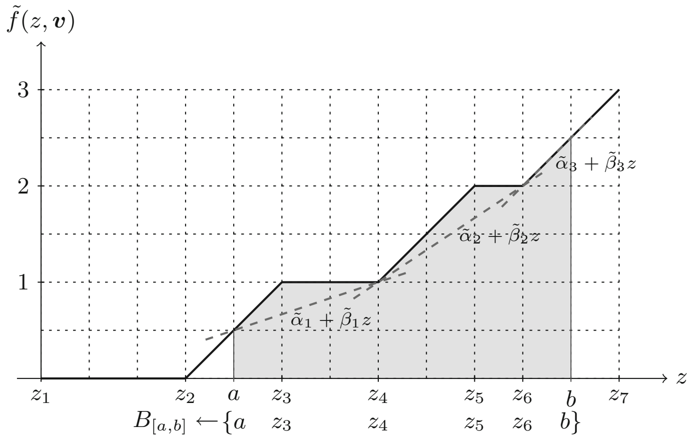
This implies that the number of possible lifted inequalities derived by this method can be exponential. Fortunately, Algorithm 1 provides a complete description of all pairs of maximal lifting coefficients; and its complexity is $\mathcal{O}(|C|)$. Moreover, we perform this process once, before performing any lifting step. Figure 1 shows an example for $\tilde{f}(z, \boldsymbol{v})$ and its lower envelope (for a given range $[a, b]$), which also shows how some key variables of the algorithm change from iteration to iteration.

It follows that a proper method to choose the (set of) inequalities to be used is crucial and it should depend on the fractional values of the current fractional point x^*, y^*. If we want to maximize the resulting violation of the lifted inequality, a possible criterion is to choose $(\alpha^*, \beta^*) \in \text{argmax}\{x_k^* \alpha + y_k^* \beta : (\alpha, \beta) \in \mathcal{H}\}$, where $\mathcal{H}$ is the set of maximal lifting coefficients. In our implementation, we choose α^*, β^* as defined before, as long as $x_k^* \alpha^* + y_k^* \beta^* > 0$; otherwise, we take $(\alpha^*, \beta^*) \in \text{argmax}\{\alpha + \beta : (\alpha, \beta) \in \mathcal{H}\}$.

3.3.2 Finding generalized flow cover inequalities in proper faces of X_G

Finding maximally violated cover inequalities is already $\mathcal{NP}$-hard [10]. Although it is possible to formulate the separation problem of generalized flow cover inequalities as an IP; we propose a simple heuristic described in Algorithm 2, this heuristic is a simple extension of other classical heuristics [11] to find maximally violated cover inequalities.

In this heuristic, for each GUB constraint $g \in G$, we compute the net contribution of the current fractional solution x^*, y^* to the knapsack constraint $a \cdot x^* + m \cdot y^* \le b$. We call this net contribution z_g. If the current GUB constraint is inactive ($z_g = 0$) we discard it from C. Otherwise, we identify the segment (called $\bar{k}_g \in M_g$), whose boundary is closest to z_g, and add it to C. If the closest boundary is the upper



It	$\{k_l, k_r\}$	$\mathcal{H}$
0	$\{a, z_3\}$	$\emptyset$
1	$\{a, z_4\}$	$\{(\tilde{\alpha}_1, \tilde{\beta}_1)\}$
2	$\{z_4, z_5\}$	$\{(\tilde{\alpha}_1, \tilde{\beta}_1)\}$
3	$\{z_4, z_6\}$	$\{(\tilde{\alpha}_1, \tilde{\beta}_1), (\tilde{\alpha}_2, \tilde{\beta}_2)\}$
4	$\{z_6, b\}$	$\{(\tilde{\alpha}_1, \tilde{\beta}_1), (\tilde{\alpha}_2, \tilde{\beta}_2), (\tilde{\alpha}_3, \tilde{\beta}_3)\}$

Fig. 1 Example $\tilde{f}(z, \boldsymbol{v})$, and a lower envelope for given a, b

limit of the segment we also add that segment to C_U. This process can be seen as a greedy construction heuristic. The final step is a local search procedure [1] whose objective is to maximize the likelihood that the resulting inequality will be violated. In our implementation 1-OPT evaluate whether adding or deleting any element from C and/or C_U improves the value of $\sum_{k \in C} \gamma_k (x_k^* - 1)$ while maintaining the condition that (C, C_U) is a GFC. If so, we perform the change; otherwise we evaluate the next element. We repeat this process until no further improvements are found.

Algorithm 2 Heuristic to find a GFC

Require: Fractional point (x^*, y^*).
Ensure: (C, C_U), generalized flow cover.

```
C ← ∅, C_U ← ∅
for g = 1 to G do
    z*_g ← Σ_{k∈M_g} (a_k x*_k + m_k y*_k)
    if z*_g > 0 then
        Select k̄_g from argmin{k ∈ M_g : min{(a_k − z*_g)_+, (z*_g − a_k − m_k)_+}}
        C ← C ∪ {k̄_g}
        if |z*_g − a_{k̄_g}| > |z*_g − a_{k̄_g} − m_{k̄_g}| then
            C_U ← C_U ∪ {k̄_g}
Apply 1-OPT trying to maximize Σ_{k∈C} γ_k (x*_k − 1)
return (C, C_U)
```

4 Numerical experiments

In this section we provide partial evidence on the effect of using the resulting set of lifted inequalities. We test the effect of using only the seed inequality and other simple heuristic methods. We also show the effect of the heuristic separation of the seed inequality. For this, we first describe a wide range of instances; then the set of experiments; and also an analysis of closed gap[2] with respect to the size of the problem.

4.1 Instances

To evaluate the performance of the inequalities presented in this paper, we consider a set of 3000 random instances inspired by the unit commitment problem. We also assume that the GUB structure and semi-continuous variables are already identified. In these instances, $|G| \in \{5, 10, 20, 40, 80\}$ and the number of elements in each GUB constraint were randomly chosen as $|M_g| \sim \{\mathcal{U}[2, 8], \mathcal{U}[7, 13], \mathcal{U}[17, 23]\}$. $a_j \sim \mathcal{U}[10, 150], m_j \sim \mathcal{U}[20, 300], \forall j \in M, b \sim \mathcal{U}[0.25, 0.95]b_{\max}$, where $b_{\max} =$ is the maximum value of the left-hand side of the knapsack constraint. We chose the cost coefficients as $c_k^x \sim 2500a_k - \mathcal{U}[370, 1000] - \mathcal{U}[15, 50]a_k, c_k^y \sim 2500m_k - \mathcal{U}[15, 50]m_k, \forall k \in M$; which represent typical cost functions in unit commitment instances. To evaluate the effect of having $m_k = 0$; half of the instances contain GUB constraints where $m_k = 0$ for 40 % of the elements in each GUB constraint.

4.2 Quality measures

We use performance profiles (see [7]) on two quality measures: closed root gap (*CG*) and closed relative gap (*CRG*), which we define as

$$CG = 100 \times \frac{z_{LP_n} - z_{LP_o}}{z_{MIP} - z_{LP_o}}, \quad CRG = 100 \times \frac{z_{LP_n}}{z_{MIP}},$$

where z_{MIP} is the optimal objective value of the mixed integer problem; z_{LP_o} is the optimal objective value of the original linear relaxation; and z_{LP_n} is that of the final LP relaxation. Note that for all our instances, $z_{LP_o} < z_{MIP}$, and $z_{MIP} > 0$. Thus, when computing *CG*, we never divide by zero. *CRG* is an approximation of the reported gap when using any commercial mixed-integer programming (MIP) solver (which might be more relevant for practitioners). *CG* is the actual improvement in the lower bound due to the given method (which is a proxy for the extent to which we improve the polyhedral representation of the given set for the given objective function).

We do not report running times because the separation process is quick in all instances and the number of calls of the separation heuristic is always less than fourteen. We do not evaluate branch and bound performance because our instances are exactly those of a single X_G problem and are always easy to solve; whereas unit commitment

[2] i.e. the gap between the linear programming optimal value (with cuts and without the cuts) and the integer programming optimal solution value.

problems can have between 24 and 336 sub-structures of this sort in addition to other side constraints. This is why we chose to leave this study for a future work.

4.3 The experiments

4.3.1 The general cutting scheme

In each case, we apply the cutting scheme described in Algorithm 3.

Algorithm 3 General cutting scheme

Require: LP^0, initial LP relaxation of X_G.
Ensure: Z, cut generation scheme optimal values.

1: $k \leftarrow 0, Z \leftarrow \emptyset$
2: **loop**
3:   Solve current relaxation LP^k.
4:   Obtain optimal value z_k^* and candidate solution (x_k^*, y_k^*).
5:   $Z \leftarrow Z \cup \{z_k^*\}$
6:   **if** $x_k^* \in \{0, 1\}^M$ **then**
7:     **return** Z
8:   From (x_k^*, y_k^*) and using Algorithm 2, find base GFC inequality satisfying $\Gamma \geq 0.1$ (it must be not violated).
9:   Lift seed inequality, as expressed in (5), while maximizing resulting violation v_k.
10:   **if** $\Gamma v_k \geq 0.1$ **then**
11:     $k \leftarrow k + 1$
12:     Add lifted seed inequality to LP^k.
13:   **else**
14:     **return** Z

This scheme can be seen as a basic cutting loop at the root node. We will evaluate the following variations of this scheme:

IP: Separation of GFC seed inequality by solving an integer program that maximizes the violation of the resulting GFC inequality (if any) without lifting.
IP+Lift: The same as before, except that we lift the resulting inequality as described in (5).
Heu: We execute Algorithm 2 without performing step 9 and use the resulting inequality (i.e., no lifting step is carried out).
Heu+1-opt: We execute Algorithm 2 and use the resulting inequality (i.e., no lifting step is performed).
Heu+1-opt+Lift: We execute Algorithm 3.
Heu+Lift: We execute Algorithm 2 without performing step 9 and lift the resulting inequality as described in (5).

4.3.2 Effectiveness of the separation heuristic

Figure 2a, b shows the performance profiles of *CG* and for *CRG* in 600 instances with five GUB constraints where we can solve the IP-separation of the base GFC inequality.

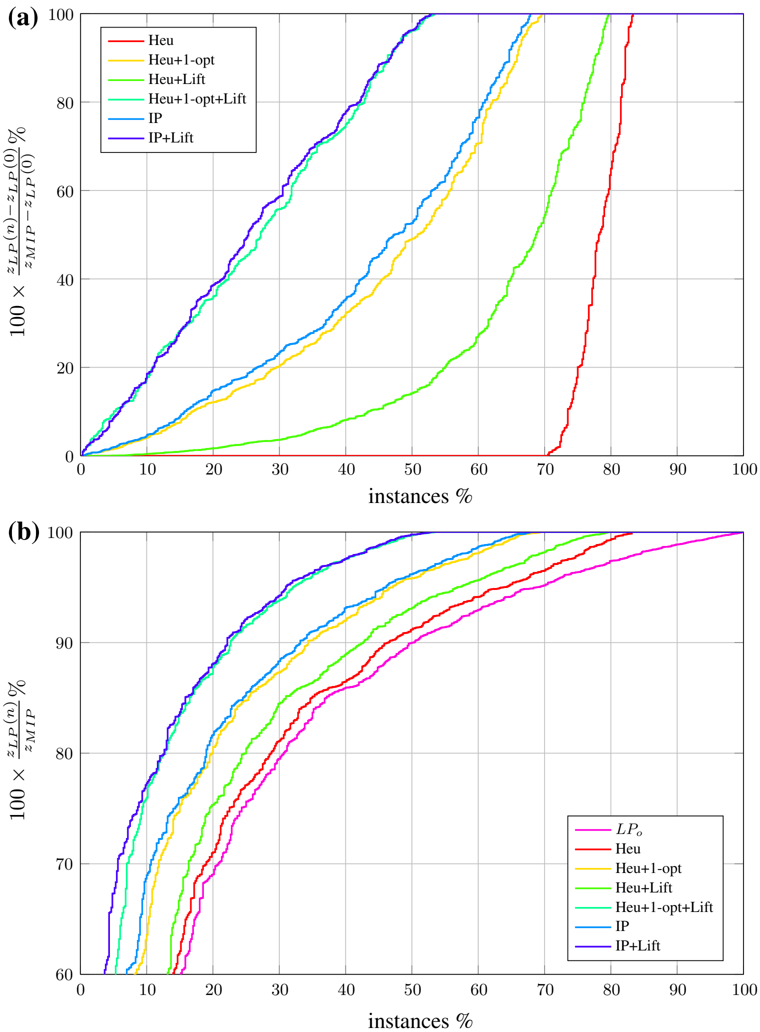


Fig. 2 *Top CG* performance profile; *bottom CRG* performance profile; for instances with $|G| = 5$

Figure 3a, b shows the performance profiles of *CG* and *CRG* for all 3000 instances.[3] Table 1 lists a summary of these results.

From these results it is clear that measured by either *CRG* or *CG*, *Heu+1-opt* performs very close to the IP separation of the base heuristic on the set of small instances while maintaining its edge over the basic heuristic for all instances. This, In addition

[3] For Figs. 2a, and 3b each point (x, y) of the plotted curves means that for the worst $x\%$ of the instances, the given method closes at most $y\%$ of absolute root integrality gap (left). For Figs. 2 and 3 each point
Footnote 3 continued
(x, y) of the plotted curves means that for the worst $x\%$ of the instances, the given method concludes with a lower bound of $y\%$ or less of the actual integer optimal solution value (right).

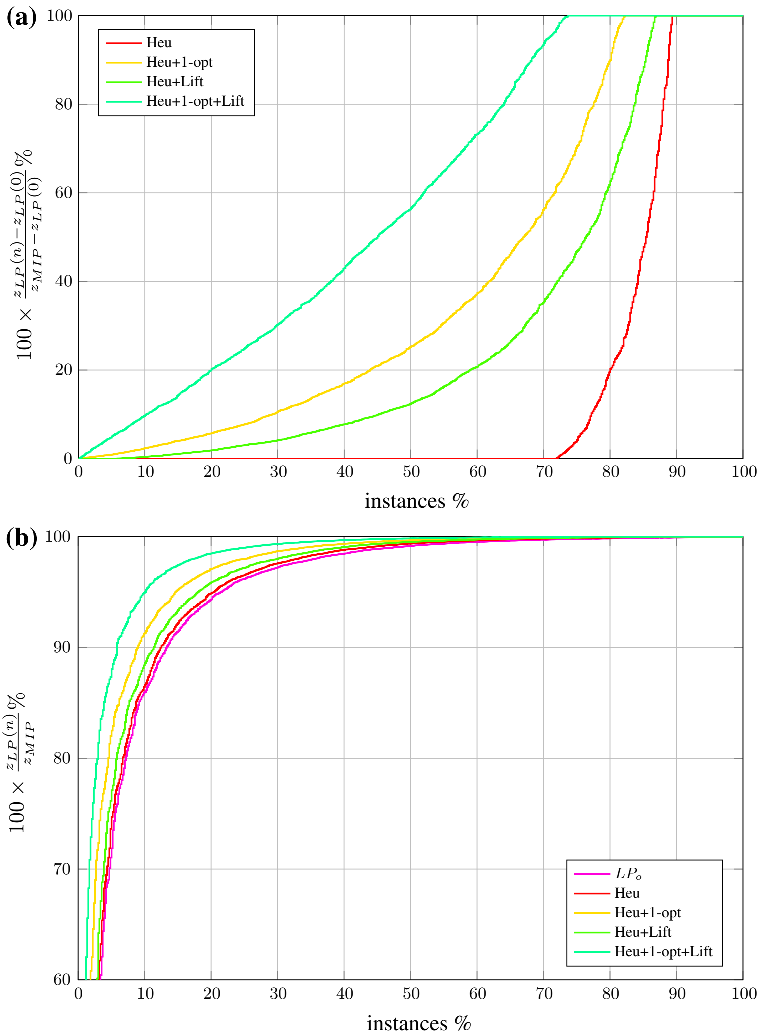


Fig. 3 *Top CG* performance profile; *bottom CRG* performance profile; for all instances

to the excessive running time cost of an exact separation routine, justifies the use of the proposed method for evaluation purposes; however, any practical implementation should address this matter in much greater detail.

4.3.3 Robustness of the results

The robustness of the results given the size of the instances is an important consideration. For this, we categorize our instances according to the number of GUB constraints ($|G|$) and the number of elements in each GUB constraint ($|M_g|$), and calculate the average *CG* and *CRG* for *Heu+1-opt+Lift*. Figure 4 is a graphical representation of

Table 1 Summary results of experiments of the average *CG*, *CRG*, and number of cuts, for small and all instances for all algorithm variations

	Average *CG*				Average *CRG*				Average N_{cuts}			
	All		$\|G\| = 5$		All		$\|G\| = 5$		All		$\|G\| = 5$	
	−Lift	+Lift	−Lift	+Lift	−Lift	+Lift	−Lift	+Lift	−Lift	+Lift	−Lift	+Lift
Heu	15.81	29.63	21.70	35.70	95.01	95.53	82.78	84.56	0.31	1.31	0.31	1.16
Heu+1opt	39.47	57.70	53.73	73.46	96.58	97.70	88.38	92.36	1.64	2.08	1.81	2.14
IP	–	–	55.81	74.13	–	–	89.13	93.11	–	–	3.17	3.36

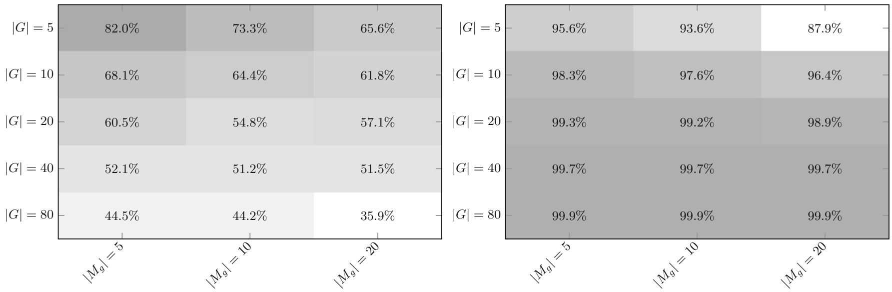


Fig. 4 *Left* average *CG* for categorized instances; *right* average *CRG* for categorized instances

the variation in the average CR and *CRG* values given these two criteria. Although we expected that *CG* performance deteriorates as we increase the number of GUB constraints and the number of elements in each GUB; it is surprising that this tendency is reversed for *CRG*. This might be due to the special cost structure used in these instances. However, if this result holds at a larger scale as well, the fact that the final relative integrality gap decreases can be beneficial.

4.3.4 The effect of lifting

As noted in Sect. 3, our seed inequality is already valid for X_G. From this, a natural question is how much do we gain by performing the lifting process? Table 1 clearly represents this aspect. If we measure *CG*, the effect is a 15–20 % larger closed root gap, and 0.5–4 % more *CRG*. Moreover, in all variations of our cutting scheme where lifting was carried out, there were only two instances where we could not find a cut. For variations without lifting, we could not find cuts for 2167 instances using *Heu* and 174 instances when using *Heu+1-opt*. This shows a strong lifting effect.

4.3.5 Number of added cuts

A common problem with cutting schemes is that they may require too many cutting rounds to achieve the desired quality. Surprisingly, in our experiments, we added an average of 2.14 cuts to all instances; in 92.2 % of instances, we added up to three cuts; for 99.33 % of instances, we added up to six cuts. In the worst case, we added 14 cuts.

5 Final comments

In this paper, we studied sequence-independent multidimensional lifting of generalized flow cover inequalities to obtain strong inequalities for the so-called semi-continuous knapsack problem with GUB constraints. We also proved that under mild assumptions, the starting inequality is facet-defining on a face of our polyhedron. Moreover, with simple assumptions, we showed that the sequence-independent lifting function is indeed the optimal (maximal) lifting function which, together with the previous result, enabled us to obtain high-dimensional facets. Unlike one-dimensional lifting, in our setting, the superadditive lifting function defines a large *class* of valid inequalities. This introduces the problem of selecting the inequality to be added. In our study, we chose the inequality to be added by maximizing the resulting violation. We used a set of 3000 randomly generated instances of different sizes to conduct our experiments. These experiments show that although the separation problem is $\mathcal{NP}$-hard, using simple heuristics and superadditive lifting function, it is possible to close, on average, 57.70 % of the root integrality gap and 97.70 % of the relative gap.

There remain several open issues, some of them are:

- Can we take advantage of GUB-partitioned binary variables in other polyhedral sets to find tight valid inequalities?
- In our setting, can we extend our analysis to the case where a_k might be negative?
- Is it possible to efficiently detect the basic GUB and the semi-continuous structure in general problems?
- Even if we are given the GUB constraints; can we use the proposed methodology in general problems?
- Other relevant questions relate to the selection of the seed inequality, and whether we should simultaneously use several seed inequalities that can better complement each other when we add them to the current LP relaxation.
- Can we prove that some class of seed inequalities are always dominated by others?
- More precisely, should we take only a minimal GFC, or is it better to add small elements to the cover?
- Are our results too specific for instances derived from the unit commitment problem?

We think that all these questions are relevant to the practical use of the proposed inequalities, and we hope to tackle them in the near future.

Acknowledgments We thank the reviewers for their thorough review and highly appreciate the comments and suggestions, which significantly contributed to improving the quality of the publication.

Appendix: Extended proofs

Proof of Proposition 1

1. $\forall j \in M$, define $\boldsymbol{e_j}$ as the index vector of dimension n with a unique one in position j. Let $\mathbf{0_n}$ be the zero vector of size n. Let $\overline{a}$ be the maximum contribution of any binary variable to the knapsack constraint, i.e. $\overline{a} = \max_{j \in M} a_j$ and $0 < \epsilon < b - \overline{a}$.

Then, $\forall k \in M$, construct $\boldsymbol{p^k} = (\boldsymbol{e_k}, \boldsymbol{0_n})$ and $\boldsymbol{q^k} = (\boldsymbol{e_k}, \epsilon \boldsymbol{e_k})$. Define $\boldsymbol{s} = (\boldsymbol{0_n}, \boldsymbol{0_n})$. The points $\boldsymbol{p^k}, \boldsymbol{q^k}$ for $k \in M$, and $\boldsymbol{s}$ belong to X_G because of Eq. (1) and $\bar{a} < b$. These points are affinely independent because $\forall k \in M$, $\boldsymbol{p^k} - \boldsymbol{s}$ (n points) and $\boldsymbol{q^k} - \boldsymbol{s}$ (n points) are linearly independent. Since we have described $2n+1$ affinely independent points in X_G; we have shown that X_G is full-dimensional. □

2. To prove that $y_k \geq 0$ is facet-defining, we use the same definitions as before, but eliminate element $\boldsymbol{q^k}$ from the set of valid points. Since we have described $2n$ affinely independent points in X_G satisfying these inequalities at equality; we have shown that these inequalities are facet-defining. □
3. To prove that $y_k \leq x_k$ is facet-defining; we use the same definitions as in Proposition 1, but eliminate element $\boldsymbol{p^k}$ from the set of valid points and redefine $\boldsymbol{q^j}$ as $(\boldsymbol{e_j}, \boldsymbol{e_j})$ for all $j \in M$. Again, since we have described $2n$ affinely independent points in X_G and these points satisfy our inequality at equality; we have shown that these inequalities are facet-defining. □
4. Note that $\overline{a}_g$ is the maximum contribution to the knapsack constraint when we choose any binary variable and the minimum contribution in M_g at the same time. This allows us to prove that for each $g \in G$, $\sum_{k \in M_g} x_k \leq 1$ is facet-defining, by constructing the points $\boldsymbol{p}^k$ and $\boldsymbol{q}^k$, for each $k \in M$ as follows:

$$\boldsymbol{p^k} = \begin{cases} (\boldsymbol{e_k} + \boldsymbol{e_{k_o}}, \boldsymbol{0_n}) & \forall k \notin M_g \\ (\boldsymbol{e_k}, \boldsymbol{0_n}) & \forall k \in M_g \end{cases}$$

and

$$\boldsymbol{q^k} = \begin{cases} (\boldsymbol{e_k} + \boldsymbol{e_{k_o}}, \epsilon \boldsymbol{e_k}) & \forall k \notin M_g \\ (\boldsymbol{e_k}, \epsilon \boldsymbol{e_k}) & \forall k \in M_g, \end{cases}$$

where $k_o \in \operatorname{argmin}_{k \in M_g}\{a_k\}$ and $0 < \epsilon < b - \overline{a}_g$. Since we have described $2n$ affinely independent points in X_G and these points satisfy the inequality at equality; we have shown that these inequalities are facet-defining. □

Proof of Theorem 1

The validity of inequality (3) was shown by Van Roy and Wolsey [21]. We prove that (3) is facet-defining for $X_o := X \cap \{x_i = 0, i \notin C\}$ by constructing a set of $2s$ affinely independent points in X_o satisfying it at equality, where $s = |C|$. For this, define $C_L = C \backslash C_U$, $C_k^+ = \{j \in C_k : \xi_j > \Gamma\}$, and $\bar{C}_k^+ = C_k \backslash C_k^+$ for $k = L, U$, and recall that $\Gamma = \xi(C) - b > 0$ and our hypothesis is $m(C_U^+) > \Gamma$. Let $t_o \in \operatorname{argmin}_{j \in C_U^+}\{\xi_j\}$ and assume that $\frac{1}{0} = \infty$. Note that $C_U^+ \neq \emptyset$, and thus t_o exists. Let $\boldsymbol{e_j}$ be the index vector of dimension s with a unique one in position j, for all $j \in C$. Let $\boldsymbol{0_s}$ be the zero vector of size s, $\boldsymbol{1_s}$ be the one vector of size s, $\boldsymbol{1_{u^+}} := \sum_{j \in C_U^+} \boldsymbol{e_j}$, $\boldsymbol{1_{\bar{u}^+}} := \sum_{j \in \bar{C}_U^+} \boldsymbol{e_j}$, and $\boldsymbol{1_u} = \boldsymbol{1_{u^+}} + \boldsymbol{1_{\bar{u}^+}}$. Then, construct $\boldsymbol{p^j}$ as

$$\boldsymbol{p^j} = \begin{cases} (\boldsymbol{1_s} - \boldsymbol{e_j}, \boldsymbol{1_u}) & \text{if } j \in C_L^+ \\ (\boldsymbol{1_s} - \boldsymbol{e_j}, \boldsymbol{1_u} - y_{oj}\boldsymbol{1_{u^+}}) & \text{if } j \in \bar{C}_L^+ \\ (\boldsymbol{1_s} - \boldsymbol{e_j}, \boldsymbol{1_u} - \boldsymbol{e_j}) & \text{if } j \in C_U^+ \\ (\boldsymbol{1_s} - \boldsymbol{e_j}, \boldsymbol{1_u} - \boldsymbol{e_j} - y_{oj}\boldsymbol{1_{u^+}}) & \text{if } j \in \bar{C}_U^+ \end{cases}$$

where $y_{oj} = \frac{\Gamma - \xi_j}{m(C_U^+)}$. Now, construct $\boldsymbol{q}^j$ as

$$\boldsymbol{q}^j = \begin{cases} p^{t_o} + (\mathbf{0}_s, \delta \boldsymbol{e}_j) & \text{if } j \in C_L \\ (\mathbf{1}_s, \mathbf{1}_u - y_{+\epsilon}\mathbf{1}_{u^+} + \varepsilon_j \boldsymbol{e}_j) & \text{if } j \in C_U^+ \\ (\mathbf{1}_s, \mathbf{1}_u - y_{-\epsilon}\mathbf{1}_{u^+} - \varepsilon_j \boldsymbol{e}_j) & \text{if } j \in \bar{C}_U^+ \end{cases}$$

where

$$0 < \delta < \min\left\{1, \min_{j \in C_L}\left\{\frac{\xi_{t_o} - \Gamma}{m_j}\right\}\right\},$$
$$0 < \epsilon < \min_{j \in C_U}\{m_j\} \min\{\Gamma, m(C_U^+) - \Gamma\}/m(C_U^+),$$

$y_{\pm\epsilon} = \frac{\Gamma \pm \epsilon}{m(C_U^+)}$, and $\varepsilon_j = \frac{\epsilon}{m_j}$ for $j \in C_U$. Given the above definitions, it is easy to prove that points $\boldsymbol{p}^j$ and $\boldsymbol{q}^j$ belong to X_o. Therefore, it is only necessary to evaluate the knapsack constraint of Eq. (1). Verifying the validity of the other constraints is straightforward. Let LHS_{KN} be the left-hand side of the knapsack constraint of Eq. (1).

Case 1 $\boldsymbol{p}^j$ with $j \in C_L^+$

$$LHS_{KN} = \xi(C) - a_j = b + \Gamma - a_j < b$$

Case 2 $\boldsymbol{p}^j$ with $j \in \bar{C}_L^+$

$$LHS_{KN} = \xi(C) - a_j - y_{oj} m(C_U^+) = b + \Gamma - a_j - \frac{\Gamma - a_j}{m(C_U^+)} m(C_U^+) = b$$

Case 3 $\boldsymbol{p}^j$ with $j \in C_U^+$

$$LHS_{KN} = \xi(C) - \xi_j = b + \Gamma - \xi_j < b$$

Case 4 $\boldsymbol{p}^j$ with $j \in \bar{C}_U^+$

$$LHS_{KN} = \xi(C) - \xi_j - y_{oj} m(C_U^+) = b + \Gamma - \xi_j - \frac{\Gamma - \xi_j}{m(C_U^+)} m(C_U^+) = b$$

Case 5 $\boldsymbol{q}^j$ with $j \in C_L$

$$LHS_{KN} = \xi(C) - \xi_{t_o} + \delta m_j = b + \Gamma - \xi_{t_o} + \delta m_j < b + \Gamma - \xi_{t_o} + \xi_{t_o} - \Gamma = b$$

Case 6 $\boldsymbol{q}^j$ with $j \in C_U^+$

$$LHS_{KN} = \xi(C) - y_{+\epsilon} m(C_U^+) + \varepsilon_j m_j = b + \varepsilon_j m_j - \epsilon = b$$

Case 7 $\boldsymbol{q^j}$ with $j \in \bar{C}_U^+$

$$LHS_{KN} = \xi(C) - y_{-\epsilon} m(C_U^+) - \varepsilon_j m_j = b - \varepsilon_j m_j + \epsilon = b$$

By proceeding in the same manner but now evaluating $\boldsymbol{p^j}$ and $\boldsymbol{q^j}$ in Eq. (3), we can show that these points satisfy this inequality at equality. Remember that the left-hand side of (3) is $\sum_{j \in C} \min\{1, \frac{\xi_j}{\Gamma}\}\left(x_j - 1\right) + \sum_{j \in C_U} \frac{m_j}{\Gamma}\left(y_j - x_j\right)$, and its right-hand side is -1.

Case 1 $\boldsymbol{p^j}$ with $j \in C_L^+$
Case 3 $\boldsymbol{p^j}$ with $j \in C_U^+$
Case 5 $\boldsymbol{q^j}$ with $j \in C_L$

$$LHS = -\min\{1, \frac{\xi_j}{\Gamma}\} = -1$$

Case 2 $\boldsymbol{p^j}$ with $j \in \bar{C}_L^+$
Case 4 $\boldsymbol{p^j}$ with $j \in \bar{C}_U^+$

$$LHS = -\min\left\{1, \frac{\xi_j}{\Gamma}\right\} - y_{oj}\frac{m(C_U^+)}{\Gamma} = -\frac{\xi_j}{\Gamma} - \left(1 - \frac{\xi_j}{\Gamma}\right) = -1$$

Case 6 $\boldsymbol{q^j}$ with $j \in C_U^+$

$$LHS = -y_{+\epsilon}\frac{m(C_U^+)}{\Gamma} + \varepsilon_j \frac{m_j}{\Gamma} = -\frac{\Gamma - \epsilon + \epsilon}{\Gamma} = -1$$

Case 7 $\boldsymbol{q^j}$ with $j \in \bar{C}_U^+$

$$LHS = -y_{-\epsilon}\frac{m(C_U^+)}{\Gamma} - \varepsilon_j \frac{m_j}{\Gamma} = -\frac{\Gamma + \epsilon - \epsilon}{\Gamma} = -1$$

Moreover, these points are affinely independent because $\forall j \in C \backslash \{t_o\}$, $\boldsymbol{p^j} - \boldsymbol{p^{t_o}}$, and $\boldsymbol{q^j} - \boldsymbol{p^{t_o}}$ are linearly independent. To show this, observe the structure of these points:

$$\begin{bmatrix} \boldsymbol{p^j} - \boldsymbol{p^{t_o}}, & j \in C_L^+ \\ \boldsymbol{p^j} - \boldsymbol{p^{t_o}}, & j \in \bar{C}_L^+ \\ \boldsymbol{p^j} - \boldsymbol{p^{t_o}}, & j \in \bar{C}_U^+ \\ \boldsymbol{p^j} - \boldsymbol{p^{t_o}}, & j \in C_U^+ \backslash \{t_o\} \\ \boldsymbol{q^j} - \boldsymbol{p^{t_o}}, & j \in C_L^+ \\ \boldsymbol{q^j} - \boldsymbol{p^{t_o}}, & j \in \bar{C}_L^+ \\ \boldsymbol{q^j} - \boldsymbol{p^{t_o}}, & j \in \bar{C}_U^+ \\ \boldsymbol{q^j} - \boldsymbol{p^{t_o}}, & j \in C_U^+ \backslash \{t_o\} \end{bmatrix} = \begin{bmatrix} \boldsymbol{e_{to}} - \boldsymbol{e_j} & \boldsymbol{e_{to}} \\ \boldsymbol{e_{to}} - \boldsymbol{e_j} & \boldsymbol{e_{to}} - y_{oj}\boldsymbol{1_{u^+}} \\ \boldsymbol{e_{to}} - \boldsymbol{e_j} & \boldsymbol{e_{to}} - \boldsymbol{e_j} - y_{oj}\boldsymbol{1_{u^+}} \\ \boldsymbol{e_{to}} - \boldsymbol{e_j} & \boldsymbol{e_{to}} \\ \boldsymbol{0_s} & \delta \boldsymbol{e_j} \\ \boldsymbol{0_s} & \delta \boldsymbol{e_j} \\ \boldsymbol{e_{to}} & \boldsymbol{e_{to}} - \varepsilon_j \boldsymbol{e_j} - y_{-\epsilon}\boldsymbol{1_{u^+}} \\ \boldsymbol{e_{to}} & \boldsymbol{e_{to}} + \varepsilon_j \boldsymbol{e_j} - y_{+\epsilon}\boldsymbol{1_{u^+}} \end{bmatrix}$$

where, if elements x_{t_o} and y_{t_o} are placed in the last position in the columns, it is possible to obtain a upper-triangular matrix (using the first $2s - 1$ columns) whose diagonal elements are non-zero. Since $2s$ affinely independent points in X_o have been described and these points satisfy inequality (3) at equality, it has been shown that these inequalities are facet-defining for $conv\{X_o\}$. □

References

1. Aarts, E., Lenstra, J.K. (eds.): Local Search in Combinatorial Optimization, 1st edn. Wiley, New York (1997)
2. Atamtürk, A.: Sequence independent lifting for mixed-integer programming. Oper. Res. **52**(3), 487–490 (2004)
3. Balas, E.: Facets of the knapsack polytope. Math. Program. **8**, 146–164 (1975)
4. Balas, E., Jeroslow, R.: Canonical cuts on the unit hypercube. Math. Program. **23**, 61–69 (1972)
5. Balas, E., Zemel, E.: Facets of the knapsack polytope from minimal covers. SIAM J. Appl. Math. **34**, 119–148 (1978)
6. Carrion, M., Arroyo, J.M.: A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Trans. Power Syst. **21**, 1371–1378 (2006)
7. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
8. Easton, T., Hooker, K.: Simultaneously lifting sets of binary variables into cover inequalities for knapsack polytopes. Discrete Optim. **5**(2), 254–261 (2008). (in Memory of George B. Dantzig)
9. Frangioni, A., Gentile, C., Lacalandra, F.: Tighter approximated milp formulations for unit commitment problems. IEEE Trans. Power Syst. **24**, 105–113 (2009)
10. Gu, Z., Nemhauser, G., Savelsbergh, M.: Lifted flow cover inequalities for mixed 0–1 integer programs. Math. Program. **85**, 439–468 (1999)
11. Gu, Z., Nemhauser, G., Savelsbergh, M.: Sequence independent lifting in mixed integer programming. J. Combin. Optim. **4**, 109–129 (2000)
12. Hammer, P., Johnson, E., Peled, U.: Facet of regular 0–1 polytopes. Math. Program. **8**(1), 179–206 (1975)
13. Hartvigsen, D., Zemel, E.: The complexity of lifted inequalities for the knapsack problem. Discrete Appl. Math. **39**(2), 113–123 (1992)
14. Johnson, E.L., Padberg, M.W.: A note on the knapsack problem with special ordered sets. Oper. Res. Lett. **1**, 18–22 (1981)
15. Kaparis, K., Letchford, A.N.: Cover inequalities. Wiley Encyclopedia of Operations Research and Management Science (2010)
16. Louveaux, Q., Wolsey, L.A.: Lifting, superadditivity, mixed integer rounding and single node flow sets revisited. Ann. OR **153**, 47–77 (2007)
17. Nemhauser, G., Wolsey, L.: Integer and Combinatorial Optimization. Wiley, New York (1988)
18. Ott, A.L.: Evolution of computing requirements in the pjm market: past and future. In: Power and Energy Society General Meeting (2010)
19. Padberg, M.W., Van Roy, T.J., Wolsey, L.A.: Valid linear inequalities for fixed charge problems. Oper. Res. **33**(4), 842–861 (1985)
20. Padberg, M.: (1, k)-Configurations and facets for packing problems. Math. Program. **18**, 94–99 (1980)
21. Roy, T.V., Wolsey, L.: Valid inequalities for mixed 0–1 programs. Discrete Appl. Math. **14**, 199–213 (1986)
22. Sherali, H., Lee, Y.: Sequential and simultaneous lifting of minimal cover inequalities for generalized upper bound constrained knapsack polytopes. SIAM J. Disc. Math. **8**, 133–153 (1995)
23. Wolsey, L.: Valid inequalities for 0–1 knapsack and mips with generalized upper bound constraints. Discrete Appl. Math. **29**, 251–261 (1988)
24. Wolsey, L.A.: Facets of linear inequalities in 0–1 variables. Math. Program. **8**, 165–178 (1975)
25. Wolsey, L.A.: Valid inequalities and superadditivity for 0/1 integer programs. Math. Oper. Res. **2**, 65–77 (1977)
26. Wood, A.J., Wollemberg, B.F.: Power Generation, Operation and Control, 2nd edn. Wiley, New York (1996)

27. Zemel, E.: Lifting the facets of zero-one polytopes. Math. Program. **15**(1), 268–277 (1978)
28. Zeng, B.: Efficient Lifting Methods for Unstructured Mixed Integer Programs with Multiple Constraints. Ph.D. thesis, Purdue University, Industrial Engineering Department (2007)
29. Zeng, B., Richard, J.P.P.: Sequence independent lifting for 0–1 knapsack problems with disjoint cardinality constraints (2006)
30. Zeng, B., Richard, J.P.P.: A framework to derive multidimensional superadditive lifting functions and its applications. In: Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science, vol. 4513, pp. 210–224 (2007)