Mathematical Programming manuscript No. 049a
2nd June 1999 12:51 CET

Zonghao Gu · George L. Nemhauser · Martin W.P. Savelsbergh

Lifted flow cover inequalities for mixed 0-1 integer programs

Received May 15, 1998 / Revised version received August 7, 1998

Abstract. We investigate strong inequalities for mixed 0-1 integer programs derived from flow cover inequalities. Flow cover inequalities are usually not facet defining and need to be lifted to obtain stronger inequalities. However, because of the sequential nature of the standard lifting techniques and the complexity of the optimization problems that have to be solved to obtain lifting coefficients, lifting of flow cover inequalities is computationally very demanding. We present a computationally efficient way to lift flow cover inequalities based on sequence independent lifting techniques and give computational results that show the effectiveness of our lifting procedures.

1. Introduction

A mixed integer program (MIP) with binary integer variables (BMIP) is the appropriate mathematical model for many practical optimization problems. This model is used, for example, for facility location problems, distribution problems, network design problems and more generally when fixed or concave costs are required in the objective function of an otherwise linear system.

Traditionally, BMIPs are solved by linear programming based branch-and-bound algorithms. The more modern technology of branch-and-cut, which is widely used for binary pure integer programs (BIP), has seen only limited use in solving BMIPs. Most commercial code use branch-and-cut with lifted cover inequalities for solving BIPs but only branch-and-bound for solving BMIPs. This successful commercialization of branch-and-cut for BIPs has occurred because there are efficient algorithms for finding strong inequalities (lifted cover inequalities) that cut off fractional solutions to LP relaxations.

The purpose of this paper is to present strong inequalities for BMIPs that can be used successfully in branch-and-cut. Our inequalities are derived from the flow cover inequalities that were introduced by Padberg, Van Roy, and Wolsey [12] and Van Roy and Wolsey [13] and implemented by Van Roy and Wolsey [14]. Balas et al. [2] use disjunctive cuts and Balas et al. [3] use Gomory cuts to solve BMIPs by a branch-and-cut algorithm.

Usually, flow cover inequalities are not facet-defining and need to be lifted to obtain stronger inequalities. However, because of the sequential nature of the standard lifting techniques and the complexity of the optimization problems that have to be solved

Z. Gu, G.L. Nemhauser, M.W.P. Savelsbergh: Georgia Institute of Technology, School of Industrial and Systems Engineering, Atlanta, GA 30332-0205, USA

Mathematics Subject Classification (1991): ■ ■ ■

to obtain lifting coefficients, lifting of flow cover inequalities is computationally very demanding. Our main contribution is a computationally efficient way to lift flow cover inequalities. This is accomplished by a technique called sequence independent lifting, which we described in detail in Gu, Nemhauser and Savelsbergh [9].

To demonstrate the effectiveness of these inequalities, we have incorporated them into the MINTO branch-and-cut system, see Nemhauser, Savelsbergh and Sigismondi [10], and have tested the system on numerous benchmark problems from MIPLIB and other sources. The results are striking, especially on difficult fixed-charge problems. We are able to solve fast, without producing large search trees, some problems that cannot be solved within reasonable time limits by commercial codes. The reason is simply that the LP relaxations without the lifted flow cover inequalities are just too weak to prune the nodes. The time spent on finding the lifted cover inequalities pays off handsomely by tightening the LP relaxations to the point where the size of the search tree is kept within reasonable limits.

The paper is organized as follows. In Section 2, we review some results on flow cover inequalities. In Section 3, we briefly review sequential and sequence independent lifting. In Section 4, we derive characteristics of the lifting problems associated with single-node flow models that are independent of the inequality under consideration. In Sections 5 and 6, we study lifted flow cover inequalities and lifted simple generalized flow cover inequalities. In Section 7, we present computational experiments.

2. Flow cover inequalities

Flow cover inequalities are valid inequalities for the system

$$X = \{(x, y) \in R^n_+ \times B^n : \sum_{j \in N^+} x_j - \sum_{j \in N^-} x_j \le d,\ x_j \le m_j y_j,\ j \in N\}, \qquad (1)$$

where $N = N^+ \cup N^-$ and $n = |N|$. The system (1) can be viewed as a single node capacitated network design model, see Figure 1.

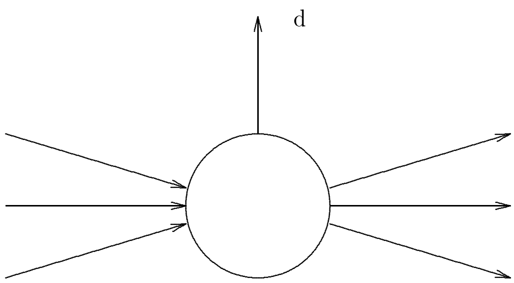

Fig. 1. Single node flow model

The x variables are arc flows that are constrained by the conservation inequality with external demand of d and the variable upper bound constraints where m_j is the capacity of arc j and y_j is a 0-1 variable that indicates whether arc j is open. The usefulness of this system for general BMIPs arises from the fact that a general inequality in a BMIP that contains both real and binary variables can be relaxed to (1).

Families of valid inequalities for (1) have been derived by Padberg, Van Roy and Wolsey [12] and Van Roy and Wolsey [13]. See Section II.2.4 of Nemhauser and Wolsey [11] and Wolsey [15] for an exposition and survey of these results.

First, we consider a single-node flow model with only inflow arcs, i.e., $N^- = \emptyset$. A set $C^+ \subseteq N^+$ is called a *flow cover* if $\sum_{j\in C^+} m_j > d$. The inequality

$$0 \le d - \sum_{j\in C^+} x_j - \sum_{j\in C^{++}} (m_j - \lambda)(1 - y_j), \tag{2}$$

where $\lambda = \sum_{j\in C^+} m_j - d > 0$ and $C^{++} = \{j \in C^+ : m_j > \lambda\}$, is called a *flow cover inequality* and is valid for (1) when $N^- = \emptyset$.

Next, we consider a general single-node flow model. A set $C = C^+ \cup C^-$ is called a *flow cover* if $C^+ \subseteq N^+$, $C^- \subseteq N^-$ and $\sum_{j\in C^+} m_j - \sum_{j\in C^-} m_j = d + \lambda$ with $\lambda > 0$. The inequality

$$0 \le d + \sum_{j\in C^-} m_j - \sum_{j\in C^+} x_j - \sum_{j\in C^{++}} (m_j - \lambda)(1 - y_j) + \sum_{j\in L^-} \lambda y_j + \sum_{j\in L^{--}} x_j, \tag{3}$$

where $L^- = \{j \in N^- \setminus C^- : m_j > \lambda\}$ and $L^{--} = N^- \setminus (L^- \cup C^-)$, is called a *simple generalized flow cover inequality* (SGFCI) and is valid for (1). The inequality

$$\begin{aligned} 0 \le d \; &+ \sum_{j\in C^-} m_j - \sum_{j\in C^+\cup L^+} x_j - \sum_{j\in C^{++}} (m_j - \lambda)(1 - y_j) \\ &+ \sum_{j\in L^+} (\max\{\overline{m}, m_j\} - \lambda) y_j - \sum_{j\in C^-} \min\left\{\lambda, [m_j - (\overline{m} - \lambda)]^+\right\}(1 - y_j) \\ &+ \sum_{j\in L^-} \max\{\lambda, m_j - (\overline{m} - \lambda)\} y_j + \sum_{j\in L^{--}} x_j, \end{aligned} \tag{4}$$

where $L^+ \subseteq N^+ \setminus C^+$, $L^- \subseteq N^- \setminus C^-$, $L^{--} = N^- \setminus (C^- \cup L^-)$, and $\overline{m} = \max_{j\in C^+} m_j$, is called an *extended generalized flow cover inequality* (EGFCI) and is also valid for (1).

Both the simple and the extended generalized flow cover inequalities have been incorporated in the mixed integer optimizers MPSARX [14] and MINTO [10].

After reviewing some results on sequence independent lifting, we will present two new families of flow cover inequalities, one of which dominates EGFCIs and the other dominates SGFCIs.

3. Sequential and sequence independent lifting

In this section, we summarize some of the results of Gu, Nemhauser, and Savelsbergh [9] on lifting without proof. The interested reader is also referred to this paper for references to earlier work on lifting.

Although the model considered here is slightly more general than that of Gu, Nemhauser, and Savelsbergh [9], the proofs are essentially the same.

Consider the set of feasible points for a BMIP given by

$$X = \Big\{ x \in R_+^{|N|} : \sum_{j \in N} a_j x_j \le d', \\ \sum_{j \in C_k} w_j x_j \le r_k,\ k = 0, \ldots, t, \\ x_j \in \{0, 1\},\ j \in I \subseteq N, \\ x_j \le u_j,\ j \in N \setminus I \Big\}.$$

Here $C_k,\ k = 0, \ldots, t$ is a partition of N; $a_j,\ j \in N$, and d' are $m \times 1$; and $w_j,\ j \in N$, and r_k are $m_k \times 1$.

Initially, we consider the subset of X with $x_j = b_j$, i.e., x_j is fixed at one of its bounds, for $j \in N \setminus C_0$ given by

$$X^0 = \Big\{ x \in R_+^{|C_0|} : \sum_{j \in C_0} a_j x_j \le d, \\ \sum_{j \in C_0} w_j x_j \le r_0, \\ x_j \in \{0, 1\},\ j \in I \cap C_0, \\ x_j \le u_j,\ j \in (N \setminus I) \cap C_0 \Big\},$$

where $d = d' - \sum_{j \in N \setminus C_0} a_j b_j$.

Let

$$0 \le \alpha_0 - \sum_{j \in C_0} \alpha_j x_j \tag{5}$$

be an arbitrary valid inequality for X^0. We want to construct a valid inequality for X of the form

$$0 \le \alpha_0 - \sum_{j \in C_0} \alpha_j x_j - \sum_{1 \le k \le t} \sum_{j \in C_k} \alpha_j (x_j - b_j). \tag{6}$$

To construct such an inequality, we start with (5) and lift the variables in $N \setminus C_0$. Without loss of generality, we assume that the variables with indices in $C_1, \ldots, C_t$ are lifted sequentially in that order and that in a given set C_k they are lifted simultaneously. Note that this contains as special cases simultaneous lifting of all variables and sequential lifting of all variables.

The intermediate sets of feasible points X^i for $i = 1, ..., t$ are defined by

$$X^i = \Big\{x \in R_+^{\sum_{0\le k\le i} |C_k|} : \sum_{j\in C_0} a_j x_j + \sum_{1\le k\le i} \sum_{j\in C_k} a_j (x_j - b_j) \le d,$$
$$\sum_{j\in C_k} w_j x_j \le r_k,\ \ k = 0, \dots, i,$$
$$x_j \in \{0, 1\},\ \ j \in I \cap (\cup_{k=0}^i C_k),$$
$$x_j \le u_j,\ \ j \in (N \setminus I) \cap (\cup_{k=0}^i C_k)\Big\}.$$

Note that if we extend X^i to X by setting $x_j = b_j$ for $j \in \cup_{k=i+1}^t C_k$, then $X^{i-1} \subseteq X^i$ for $i = 1, ..., t$ and $X^t = X$.

For $i = 1, ..., t$, the *lifting problem* associated with C_i, given a valid inequality

$$0 \le \alpha_0 - \sum_{j\in C_0} \alpha_j x_j - \sum_{1\le k<i} \sum_{j\in C_k} \alpha_j (x_j - b_j) \tag{7}$$

for X^{i-1}, is to find α_j for $j \in C_i$ such that

$$\sum_{j\in C_i} \alpha_j (x_j - b_j) \le \alpha_0 - \sum_{j\in C_0} \alpha_j x_j - \sum_{1\le k<i} \sum_{j\in C_k} \alpha_j (x_j - b_j) \tag{8}$$

is a valid inequality for X^i.

For $i = 1, ...t$, let

$$Z^i = \Big\{z \in R^m : \exists x \in X^i :\ \sum_{j\in C_i} a_j (x_j - b_j) = z \text{ and}$$
$$\sum_{j\in C_0} a_j x_j + \sum_{1\le k<i} \sum_{j\in C_k} a_j (x_j - b_j) \le d - z\Big\},$$

and for $z \in Z^i$ let

$$h_i(z) = \max \sum_{j\in C_i} \alpha_j (x_j - b_j)$$
$$\text{s.t.} \ \sum_{j\in C_i} a_j (x_j - b_j) = z$$
$$\sum_{j\in C_i} w_j x_j \le r_i$$
$$x_j \in \{0, 1\},\ \ j \in I \cap C_i$$
$$0 \le x_j \le u_j,\ \ j \in (N \setminus I) \cap C_i,$$

and

$$
\begin{aligned}
f_i(z) = \min \alpha_0 - & \sum_{j \in C_0} \alpha_j x_j - \sum_{1 \le k < i} \sum_{j \in C_k} \alpha_j (x_j - b_j) \\
\text{s.t.} \quad & \sum_{j \in C_0} a_j x_j + \sum_{1 \le k < i} \sum_{j \in C_k} a_j (x_j - b_j) \le d - z \\
& \sum_{j \in C_k} w_j x_j \le r_k, \; k = 0, \dots, i-1 \\
& x_j \in \{0, 1\}, \; j \in I \cap (\cup_{k=0}^{i-1} C_i) \\
& 0 \le x_j \le u_j, \; j \in (N \setminus I) \cap (\cup_{k=0}^{i-1} C_i).
\end{aligned}
$$

Proposition 1. *For $i = 1, ..., t$, inequality (8) is valid for X^i for any choice of α_j for $j \in C_i$ such that $h_i(z) \le f_i(z)$ for $z \in Z^i$.*

When α_j for $j \in C_i$ are such that $h_i(z) = f_i(z)$ has $|C_i|$ solutions $x^1, x^2, \dots, x^{|C_i|}$ such that the components in C_i of $x^1 - b, x^2 - b, \dots, x^{|C_i|} - b$ are linearly independent, we say that the lifting is *maximal*.

Theorem 1. *For $i = 1, ..., t$, if $conv(X^{i-1})$ and $conv(X^i)$ are full dimensional, (7) defines a facet of $conv(X^{i-1})$ and $\alpha_0 \ne 0$, then (8) defines a facet of $conv(X^i)$ if and only if the lifting is maximal.*

Corollary 1. *Given an arbitrary valid inequality (5) for X^0, we can construct a valid inequality (6) for X by sequentially lifting sets C_i for $i = 1, ..., t$. At each step i, the lifting coefficients have to be such that $h_i(z) \le f_i(z)$ for $z \in Z^i$. If (5) defines a facet of $conv(X^0)$, $conv(X^i)$ is full dimensional for $i = 0, ..., t-1$, and at each step i the lifting is maximal, then (6) defines a facet of $conv(X)$.*

It should be clear that lifting coefficients are, in general, dependent on the lifting sequence $C_1, C_2, \dots, C_t$.

To avoid having to work with individual domains Z^i, we choose to work with a bounded convex set Z such that $Z^i \subseteq Z$ for $i = 1, ..., t$. Let $L_k = \sum_{j \in (N \cap I) \setminus C_0 : a_{kj} < 0} a_{kj} + \sum_{j \in (N \setminus I) \setminus C_0 : a_{kj} < 0} a_{kj} u_j$ and $U_k = \sum_{j \in (N \cap I) \setminus C_0 : a_{kj} > 0} a_{kj} + \sum_{j \in (N \setminus I) \setminus C_0 : a_{kj} > 0} a_{kj} u_j$ for $k = 1, 2, \dots, m$. Then we define $Z_k = \{z \in R : z = \lambda L_k + (1 - \lambda) U_k, \; 0 \le \lambda \le 1\}$ and $Z = Z_1 \times Z_2 \times \dots \times Z_m$.

Definition 1. *The lifting function f with respect to valid inequality (5) for X^0 is defined to be $f(z) = f_1(z)$ for all $z \in Z$.*

Note that if $f(z) = f_i(z)$ for $z \in Z$ and $i = 2, ..., t$, then the lifting coefficients in a sequential lifting of the elements of $C_1, C_2, \dots, C_t$ are independent of the ordering of the set $\{C_1, C_2, \dots, C_t\}$.

Definition 2. *If $f(z) = f_i(z)$ for $z \in Z$ and $i = 2, ..., t$, the lifting is said to be sequence independent.*

Definition 3. *A function f is superadditive on U if f is bounded for all $u \in U$ and $f(u_1) + f(u_2) \le f(u_1 + u_2)$ for all u_1, u_2 and $u_1 + u_2$ in U. Now we give a sufficient condition for sequence independent lifting.*

Theorem 2. *If f is superadditive on Z, then lifting is sequence independent.*

Obviously, a superadditive lifting function greatly reduces the computational burden of the lifting process. Instead of having to compute lifting functions f_i for all i, we only have to compute f. Unfortunately f is often not superadditive. To be able to profit from the computational advantages of sequence independent lifting, we consider the class of superadditive valid lifting functions.

Definition 4. *A superadditive function g is called a superadditive valid lifting function for f, if $g(z) \leq f(z)$ for all $z \in Z$.*

Theorem 3. *If g is a superadditive valid lifting function and if α_j for $j \in C_i$ are such that $h_i(z) \leq g(z)$ for $z \in Z$ and for $i = 1, \ldots, t$, then the lifted inequality (6) is valid for X.*

Next, we address the problem of choosing a 'good' superadditive valid lifting function. A desirable property is that g should not be *dominated* by another superadditive valid lifting function g', i.e., there is no superadditive g' with $g(z) \leq g'(z)$ for all $z \in Z$ and $g(z') < g'(z')$ for some $z' \in Z$.

Another interesting property is maximality. Let $E = \{z \in Z : f_i(z) = f(z)$ for $i = 1, \ldots, t\}$. We say that g is a *maximal* superadditive valid lifting function if $g(z) = f(z)$ for all $z \in E$. Note that if f is superadditive, then $E = Z$, and that if $\gamma(z) = f(z)$, then $z \in E$.

4. Lifting for single-node flow models

Again, consider the set X given by (1). In this section, we analyze the simultaneous lifting of a single variable pair (x_p, y_p). Let

$$\overline{X} = \{(x, y) \in X : (x_p, y_p) = (b_x, b_y)\},$$

where (b_x, b_y) is equal to $(0, 0)$ or $(m_p, 1)$, i.e., $\overline{X}$ is the subset of feasible solutions in which x_p and y_p are fixed to either their lower or upper bounds.

Suppose that both X and $\overline{X}$ are full dimensional, in $R^n_+ \times B^n$ and $R^{n-1}_+ \times B^{n-1}$ respectively, and that

$$0 \leq \alpha_0 - \sum_{j \in N^+ \cup N^- \setminus \{p\}} (\alpha_j x_j + \beta_j y_j) \tag{9}$$

defines a facet of $\text{conv}(\overline{X})$.

Lifting the variable pair (x_p, y_p) means finding coefficients (α_p, β_p) such that

$$\alpha_p(x_p - b_x) + \beta_p(y_p - b_y) \leq \alpha_0 - \sum_{j \in N^+ \cup N^- \setminus \{p\}} (\alpha_j x_j + \beta_j y_j) \tag{10}$$

is a valid inequality for X.

Let

$$\begin{aligned} h(z) = \max\ & \alpha_p(x_p - b_x) + \beta_p(y_p - b_y) \\ \text{s.t.}\ & \delta_p(x_p - b_x) = z \\ & 0 \le x_p \le m_p y_p \\ & y_p \in \{0, 1\}, \end{aligned}$$

where $\delta_p = 1$ if $p \in N^+$ and $\delta_p = -1$ if $p \in N^-$, and let

$$\begin{aligned} f(z) = \min\ & \alpha_0 - \sum_{j \in N^+ \cup N^- \setminus \{p\}} (\alpha_j x_j + \beta_j y_j) \\ \text{s.t.}\ & \sum_{j \in N^+} x_j - \sum_{j \in N^-} x_j \le d - z \\ & 0 \le x_j \le m_j y_j,\ \ y_j \in \{0, 1\},\ \ j \in N^+ \cup N^- \\ & x_p = b_x,\ \ y_p = b_y. \end{aligned}$$

To ensure that (10) is a valid inequality for X, we require that $h(z) \le f(z)$ for all z. Furthermore, (10) will be facet inducing if and only if $h(z) = f(z)$ has two solutions (x^1, y^1) and (x^2, y^2) such that $(x^1_p - b_x, y^1_p - b_y)$ and $(x^2_p - b_x, y^2_p - b_y)$ are linearly independent. Note that h is independent of the specific facet inducing inequality (9) we are trying to lift. We will use the above observations to analyze the form of h for the lifted inequality (10) to be facet inducing. We distinguish four cases depending on whether $p \in N^+$ or $p \in N^-$ and on whether $(b_x, b_y) = (0, 0)$ or $(b_x, b_y) = (m_p, 1)$.

Theorem 4. *1. If $p \in N^+$, $(b_x, b_y) = (0, 0)$ and (10) defines a facet of conv(X), then*

$$h(z) = \begin{cases} 0 & z = 0 \\ \alpha_p z + \beta_p & 0 < z \le m_p \end{cases}$$

with $\alpha_p \ge 0$ and $\beta_p \le 0$.

2. If $p \in N^+$, $(b_x, b_y) = (m_p, 1)$ and (10) defines a facet of conv(X), then

$$h(z) = \begin{cases} \alpha_p z & -m_p < z \le 0 \\ -\alpha_p m_p - \beta_p & z = -m_p \end{cases}$$

with $\alpha_p \ge 0$ and $\beta_p \le 0$.

3. If $p \in N^-$, $(b_x, b_y) = (0, 0)$ and (10) defines a facet of conv(X), then

$$h(z) = \begin{cases} 0 & z = 0 \\ -\alpha_p z + \beta_p & -m_p \le z < 0 \end{cases}$$

with $\alpha_p \le 0$ and $\beta_p \le 0$.

4. If $p \in N^-$, $(b_x, b_y) = (m_p, 1)$ and (10) defines a facet of conv(X), then

$$h(z) = \begin{cases} -\alpha_p z & 0 \le z < m_p \\ -\alpha_p m_p - \beta_p & z = m_p \end{cases}$$

with $\alpha_p \le 0$ and $\beta_p \le 0$.

Proof. Case 1. Since $\delta_p = 1$, $b_x = 0$ and $x_p \geq 0$, we have $z \geq 0$. We prove $\beta_p \leq 0$. Suppose not. Since (9) defines a facet, $f(0) = 0 < \beta_p = h(0)$, which is a contradiction. Next we show $\alpha_p \geq 0$. If $\alpha_p < 0$, then $h(z) < 0$ for $z > 0$. Since $f(z) \geq 0$ for $z \geq 0$, $h(z) < f(z)$ for $z > 0$, and $h(z) = f(z)$ cannot have two solutions. Hence $\alpha_p \geq 0$.

Case 2. Since $\delta_p = 1$, $b_x = m_p$ and $x_p \leq m_p$, we have $z \leq 0$. If $\beta_p > 0$, then any optimal solution (x_p, y_p) must have $y_p = 1$. Then $f(z) = h(z)$ cannot have two solutions with linearly independent $(x_p^1 - m_p, y_p^1 - 1)$ and $(x_p^2 - m_p, y_p^2 - 1)$. Hence $\beta_p \leq 0$. If $\alpha_p < 0$, then $h(z) = \alpha_p z > 0$ for $-m_p < z < 0$. Obviously $f(z) \leq 0$ for $z \leq 0$. So $h(z) > f(z)$ for $-m_p < z < 0$, which gives a contradiction.

Case 3. Similar to the proof of Case 1.

Case 4. Similar to the proof of Case 2.

□

In Figures 2 and 3, we illustrate the first two cases discussed in Theorem 4. Observe that f is a monotone nondecreasing function with $f(0) = 0$ (since we have assumed that (9) is facet inducing). Furthermore, observe that $h(0) = 0$ in all four cases and that h is continuous at 0 in Cases 2 and 4, but not continuous at 0 in Cases 1 and 3.

First, we consider Case 1. Figure 2 shows a typical realization of the function f on the interval $[0, m_p]$. In this situation, there are five possible functions h_i $i = 1, \ldots, 5$ that satisfy the conditions of Case 1 and give rise to facets. To be facet inducing $f(z) = h_i(z)$ needs to have two solutions (x^1, y^1) and (x^2, y^2) such that (x_p^1, y_p^1) and (x_p^2, y_p^2) are linearly independent. Note that each h_i defines a different pair of lifting coefficients (α_p, β_p).

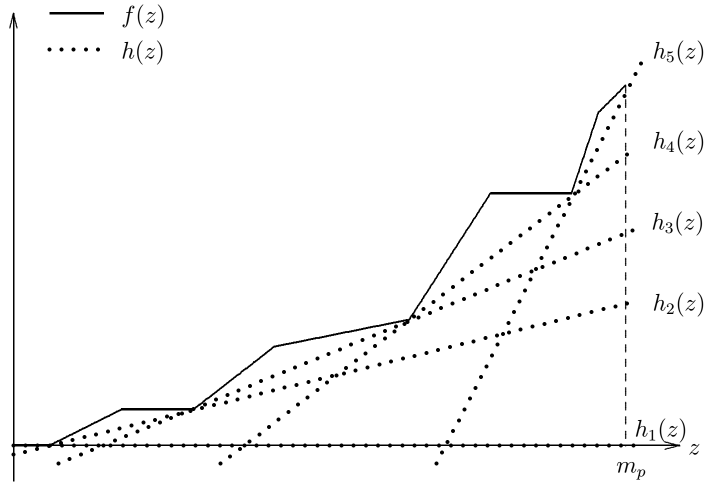

Fig. 2. $f(z)$ and $h(z)$ for Case 1

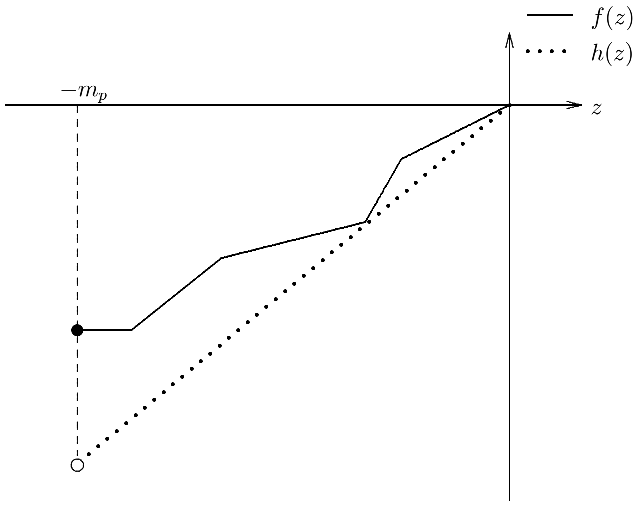

Fig. 3. $f(z)$ and $h(z)$ for Case 2

Next, we consider Case 2. Figure 3 shows a typical realization of the function f on the interval $[-m_p, 0]$. In this situation, there is only one function h that satisfies the conditions of Case 2 and gives rise to a facet. This is caused by the fact that $h(0) = 0$ and h has to be continuous at 0. Since h is unique, the pair of lifting coefficients (α_p, β_p) is unique as well.

Observe that if we know f, then we can obtain all facets. Unfortunately, f is usually very hard to compute. However, the LP relaxation f_{LP} of the lifting function f is easy to compute and can be used to do approximate lifting. Let

$$
\begin{aligned}
f_{LP}(z) = \alpha_0 - \max & \sum_{j \in N^+ \cup N^- \setminus \{p\}} (\alpha_j x_j + \beta_j y_j) \\
\text{s.t} & \sum_{j \in N^+} x_j - \sum_{j \in N^-} x_j \le d - z \\
& 0 \le x_j \le m_j y_j, \ 0 \le y_j \le 1, \ j \in N^+ \cup N^- \\
& x_p = b_x, \ y_p = b_y.
\end{aligned}
$$

If $h \le f_{LP}$, then $h \le f$, since $f_{LP} \le f$. Therefore, by Proposition 1, the inequality (10) is valid but may no longer define a facet of $\text{conv}(X)$. However, in some situations, we may still get facets by using f_{LP}. In Figure 4, we show f_{LP} for Case 1 . If we use f_{LP}, we still get facets with h_3 and h_4. Note that the use of f_{LP} has the possibility of yielding facets only for Cases 1 and 3. We have to know $f(-m_p)$ for Case 2 and $f(m_p)$ for Case 4 to get a facet.

In the next two sections, we show how sequence independent lifting can be used to derive classes of lifted flow cover inequalities.

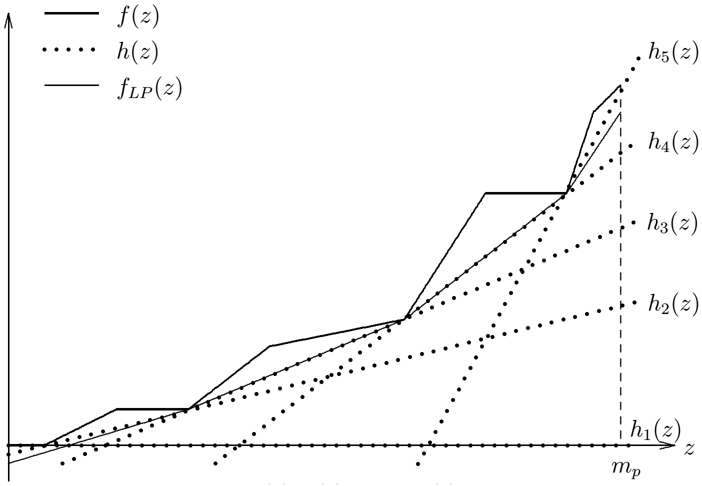

Fig. 4. $f(z)$, $h(z)$ and $f_{LP}(z)$ for Case 1

5. Lifted flow cover inequalities

Consider the set X of feasible solutions of the single-node flow model defined by (1). Given a flow cover C, if we fix variable pairs (x_j, y_j) to $(0, 0)$ for $j \in (N^+ \cup N^-) \setminus C$ and to $(m_j, 1)$ for $j \in C^-$, then we get a single-node flow model with only inflow arcs and an associated subset X^0 of feasible solutions

$$\begin{aligned} X^0 = \{(x, y) : & \sum_{j \in C^+} x_j \le d + \sum_{j \in C^-} m_j \\ & 0 \le x_j \le m_j y_j \\ & y_j \in \{0, 1\}, \;\; j \in C^+\}. \end{aligned}$$

In this case, the flow cover inequality (2) defines a facet of conv(X^0), see Section II.2.4 of Nemhauser and Wolsey [11].

Let $d' = d + \sum_{j \in C^-} m_j$. The lifting function associated with (2) is

$$\begin{aligned} f(z) = \min \; & d' - \sum_{j \in C^+} x_j - \sum_{j \in C^{++}} (m_j - \lambda)(1 - y_j) \\ \text{s.t.} \; & \sum_{j \in C^+} x_j \le d' - z \qquad\qquad (11) \\ & x_j \le m_j y_j, \;\text{ for } j \in C^+ \\ & (x, y) \in R_+^{|C^+|} \times B^{|C^+|}. \end{aligned}$$

Let $C^{++} = \{j_1, \ldots, j_r\}$ with $m_{j_i} \ge m_{j_{i+1}}$ for $i = 1, \ldots, r - 1$. Let $M_0 = 0$ and $M_i = \sum_{k=1}^{i} m_{j_k}$ for $i = 1, \ldots, r$.

Theorem 5. *The lifting function f is given by*

$$f(z) = \begin{cases} -\lambda & z \le -\lambda \\ z & -\lambda \le z \le 0 \\ i\lambda & M_i \le z \le M_{i+1} - \lambda \quad i = 0, ..., r-1 \\ z - M_i + i\lambda & M_i - \lambda \le z \le M_i \quad i = 1..., r-1 \\ z - M_r + r\lambda & M_r - \lambda \le z \le d'. \end{cases}$$

Proof. Observe that in any optimal solution either $x_j \ge m_j - \lambda$ and $y_j = 1$ or $x_j = 0$ and $y_j = 0$ for $j \in C^{++}$, either $x_j > 0$ and $y_j = 1$ or $x_j = 0$ and $y_j = 0$ for $j \in C^+ \setminus C^{++}$, and that if we cannot set all x_j's to their upper bounds, we should decrease the variables in C^{++} in the order $j_1, j_2, ..., j_r$ since $m_{j_i} > m_{j_{i+1}}$ for $i = 1, ..., r-1$. This gives the function f.

□

Theorem 6. *The function f is superadditive on* $[0, d']$*, but not on* $(-\infty, d']$.

Proof. To show that f is superadditive on $[0, d']$, we use the superadditive function g_2 defined in the Appendix. Let $l = \lambda$, $v_i = \lambda$ for all i, and $u_i = m_i - \lambda$ for all i. Then

$$g_2(z) = \begin{cases} h\lambda & M_h \le z \le M_{h+1} - \lambda \\ z - M_h + h\lambda & M_h - \lambda < z < M_h, \end{cases}$$

and $f(z) = g_2(z)$ for $0 \le z \le d'$.

To show that f is not superadditive on $(-\infty, d']$, observe that $f(-M_1 - \lambda) + f(M_1) = -\lambda + \lambda = 0 > f(-\lambda)$.

□

To lift (2), we define the following superadditive valid lifting function g

$$g(z) = \begin{cases} i\lambda & im_1 \le z \le (i+1)m_1 - \lambda, \ i = 0, \pm 1, \pm 2, \dots \\ z - im_1 + i\lambda & im_1 - \lambda \le z \le im_1, \ i = 0, \pm 1, \pm 2, \dots \end{cases}$$

Figure 5 shows functions f and g.

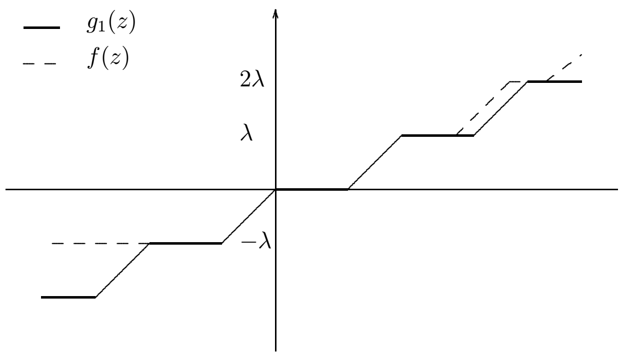

Fig. 5. Functions f and g.

Theorem 7. *The function g is a maximal and nondominated superadditive valid lifting function for f.*

Proof. First, we show that $g(z) \leq f(z)$. Comparing $g(z)$ and $f(z)$, we see that $g(z) \leq f(z)$ for $z \leq 0$ and for $z > d'$. Since $im_1 \geq M_i$, if $M_i \leq z \leq M_{i+1} - \lambda$, then $z \leq im_i - \lambda$ and thus $g(z) \leq i\lambda = f(z)$. If $M_i - \lambda \leq z \leq M_i$, then if $im_1 - \lambda > z$, $g(z) \leq i\lambda \leq f(z)$, and if $im_1 - \lambda \leq z$, $z - im_1 \leq z - M_i$ and thus $g(z) \leq f(z)$. Similarly we can show $g(z) \leq f(z)$ for $M_r - r \leq z \leq d'$.

Second, we show that $g(z)$ is superadditive. If we let $l = v_1 = \lambda$, and $w_1 = m_1$, then $g(z)$ is the function $g_1(z)$ defined in the Appendix. Hence $g(z)$ is superadditive.

Third, we show that $g(z)$ is nondominated. Suppose not. Then there is another superadditive lifting function $g'(z)$ such that $g'(z) \geq g(z)$ for all z and there exists a z' with $g'(z') > g(z')$.
Case $z' < 0$*:* Let z^* be the largest $z \leq 0$ such that $g'(z^*) > g(z^*)$. If $z^* \in [i^*m_1 - \lambda, i^*m_1]$, then $i^* < -1$, since $g(z) = f(z)$ for $-m_1 \leq z \leq m_1$, and $i^*m_1 - z^* \leq \lambda$. So $g'[m_1 - \lambda + (i^*m_1 - z^*)] = g[m_1 - \lambda + (i^*m_1 - z^*)] = i^*m_1 - z^*$ and $g'[(i^*+1)m_1 - \lambda] = g[(i^*+1)m_1 - \lambda] = i^*\lambda$. Thus, by superadditivity of $g'(z)$, we have

$$\begin{aligned} g'(z^*) &\leq g'(m_1 - \lambda + i^*m_1) - g'[m_1 - \lambda + (i^*m_1 - z^*)] \\ &= i^*\lambda - (i^*m_1 - z^*) \\ &= g(z^*), \end{aligned}$$

which is a contradiction. Since $g'(z)$ must be nondecreasing in z, z^* cannot be in $[im_1, (i+1)m_1 - \lambda]$. *Case* $z' \geq 0$*:* If $z^* \in [im_1, (i+1)m_1 - \lambda]$, then $(-i+1)m_1 - \lambda \leq -z' + m_1 - \lambda \leq (-i+1)m_1$. Thus, by superadditivity of $g'(z)$, we have

$$\begin{aligned} g'(z') &\leq g'(m_1 - \lambda) - g'(-z' + m_1 - \lambda) \\ &\leq f(m_1 - \lambda) - g(-z' + m_1 - \lambda) \\ &= 0 - [-z + m_1 - \lambda - (-i+1)m_1 + (-i+1)\lambda] \\ &= z - im_1 + i\lambda \\ &= g(z), \end{aligned}$$

which is a contradiction. Since $g'(z)$ must be nondecreasing in z, z' cannot be in $[im_1, (i+1)m_1 - \lambda]$. Hence there does not exist a z' such that $g(z') > g(z)$ and $g(z)$ is nondominated.

Finally, we show that $g(z)$ is maximal. For any $z > m_1$, if we first lift a variable pair $j \in N^- \setminus C^-$ with $m_j = -z - m_1$, we get lifting function $f_2(z)$, see Section 2. It is easy to see that $f_2(z) = 0 < f(z)$. So $z \notin E(Z)$. Similarly we can prove that if $z < -m_1$, then $z \notin E(Z)$. Thus $E(Z) \subseteq [-m_1, m_1]$. Since $g(z) = f(z)$ for $z \in [-m_1, m_1]$, $g(z)$ is maximal.

□

We can use $g(z)$ to do sequence independent lifting for variables in $N^+ \setminus C^+$, C^- and $N^- \setminus C^-$. Lifting of variable pairs in $N^+ \setminus C^+$ is covered by Theorem 4, Case 1, lifting of variable pairs in C^- is covered by Theorem 4, Case 4, and lifting of variable pairs in $N^- \setminus C^-$ is covered by Theorem 4, Case 3.

Theorem 8. *The lifted flow cover inequality (LFCI)*

$$\sum_{j\in C^+\cup L^+} x_j + \sum_{j\in C^{++}} (m_j-\lambda)(1-y_j) - \sum_{j\in L^+} \beta_j y_j \\ \leq d' - \sum_{j\in C^-} g(m_j)(1-y_j) + \\ \sum_{j\in L^-} -g(-m_j)y_j + \sum_{j\in L^{--}} x_j, \tag{12}$$

where $\beta_j = m_j - i\lambda$ *if* $im_1 \leq m_j \leq (i+1)m_1 - \lambda$ *and* $\beta_j = i(m_1-\lambda)$ *if* $im_1 - \lambda < z < im_1$*, is valid for* X*.*

Proof. To prove validity, based on Proposition 1, we have to show that for each of the lifted variable pairs we have $h(z) \leq g(z)$ for all z. Below we list the lifting coefficients (α_j, β_j) that have been used to obtain (12). Note that they satisfy the conditions of Theorem 4 and that for these choices $h(z) \leq g(z)$.

- For $j \in L^+$, $(\alpha_j, \beta_j) = (1, -m_j + i\lambda)$ if $im_1 \leq z \leq (i+1)m_1 - \lambda$ and $(\alpha_j, \beta_j) = (1, -i(m_1-\lambda))$ if $im_1 - \lambda \leq z \leq im_1$.
- For $j \in N^+ \setminus (C^+ \cup L^+)$, $(\alpha_j, \beta_j) = (0, 0)$.
- For $j \in C^-$, $(\alpha_j, \beta_j) = (0, -g(m_j))$.
- For $j \in L^-$, $(\alpha_j, \beta_j) = (0, g(-m_j))$.
- For $j \in L^{--}$, $(\alpha_j, \beta_j) = (-1, 0)$.

□

An LFCI is a strengthening of an EGFCI as shown in the following theorem.

Theorem 9. *The inequality (12) is at least as strong as (4).*

Proof. It is easy to check that $\beta_j \leq \max\{m_1, m_j\} - \lambda$ for $j \in L^+$, $g(m_j) \geq \min\{\lambda, [m_j - (m_1-\lambda)]^+\}$ and $-g(-m_j) \leq \max\{\lambda, m_j - (m_1 - \lambda)\}$. Hence (12) is at least as strong as (4).

□

6. Lifted simple generalized flow cover inequalities

Again, consider the set X of feasible solutions of a single-node flow model defined by (1). Given a flow cover C, if we fix variable pairs (x_j, y_j) to $(0, 0)$ for $j \in N^+ \setminus C^+$ and to $(m_j, 1)$ for $j \in C^-$, then we get a reduced single-node flow model and an associated subset X^0 of feasible solutions

$$X^0 = \Big\{(x,y) : \sum_{j\in C^+} x_j - \sum_{j\in N^-\setminus C^-} x_j \leq d + \sum_{j\in C^-} m_j \\ 0 \leq x_j \leq m_j y_j, \text{ for } j \in C^+ \cup (N^- \setminus C^-) \\ y_j \in \{0,1\}, \text{ for } j \in C^+ \cup (N^- \setminus C^-)\Big\}.$$

In this case, the simple generalized flow cover inequality (3) defines a facet of conv(X^0), see Section II.6.4 of Nemhauser and Wolsey [11].

Let $d' = d + \sum_{j\in C^-} m_j$. The lifting function $f(z)$ associated with (3) is

$$\begin{aligned} f(z) = \min\ & d' - \sum_{j\in C^+} x_j - \sum_{j\in C^{++}} (m_j - \lambda)(1 - y_j) + \sum_{j\in L^-} \lambda y_j + \sum_{j\in L^{--}} x_j \\ s.t.\quad & \sum_{j\in C^+} x_j - \sum_{j\in N^-\setminus C^-} x_j \le d' - z \\ & x_j \le m_j y_j,\ \text{for } j \in C^+ \cup (N^- \setminus C^-) \\ & (x, y) \in R_+^{|C^+\cup(N^-\setminus C^-)|} \times B^{|C^+\cup(N^-\setminus C^-)|}. \end{aligned}$$

Let $r = |C^{++} \cup L^-|$, $C^{++} \cup L^- = \{j_1, j_2, \ldots j_r\}$ with $m_{j_i} \ge m_{j_{i+1}}$ for $i = 1, \ldots r-1$, $M_0 = 0$, and $M_i = \sum_{k=1}^{i} m_{j_k}$ for $i = 1, \ldots, r$. Furthermore, let $mp = \min_{j\in C^{++}} m_j$, $m = \sum_{j\in C^+\setminus C^{++}} m_j + \sum_{j\in L^{--}} m_j$, $ml = \min(m, \lambda)$, t be the largest index in $C^{++} \cup L^-$ such that $m_{j_t} = mp$. Without loss of generality, we assume $t \in C^{++}$. Let $\rho_i = \max[0, m_{i+1} - (mp - \lambda) - ml]$ for $i = t, \ldots, r-1$ and let $d'' = d' + \sum_{j\in N^-\setminus C^-} m_j$

Theorem 10. *The lifting function f is given by*

$$f(z) = \begin{cases} i\lambda & M_i \le z \le M_{i+1} - \lambda,\ i = 0, \ldots, t-1 \\ z - M_i + i\lambda & M_i - \lambda \le z \le M_i,\ i = 1, \ldots, t-1 \\ z - M_i + i\lambda & M_i - \lambda \le z \le M_i - \lambda + ml, \\ & i = t, \ldots, r-1 \\ z - M_i + (i+1)\lambda - ml - \rho_i & M_i - \lambda + ml < z \le M_i - \lambda + ml + \rho_i, \\ & i = t, \ldots, r-1 \\ i\lambda & M_i - \lambda + ml + \rho_i \le z \le M_{i+1} - \lambda, \\ & i = t, \ldots, r-1 \\ z - M_r + r\lambda & M_r - \lambda \le z \le d''. \end{cases}$$

Proof. First, we rewrite f as

$$\begin{aligned} f(z) = \min\ & d' + \sum_{j\in N^-\setminus C^-} m_j - \sum_{j\in C^+} x_j - \sum_{j\in C^{++}} (m_j - \lambda)(1 - y_j) - \\ & \sum_{j\in L^-} (m_j - \lambda y_j) - \sum_{j\in L^{--}} (m_j - x_j) \\ s.t.\quad & \sum_{j\in C^+} x_j + \sum_{j\in N^-\setminus C^-} (m_j - x_j) \le d' + \sum_{j\in N^-\setminus C^-} m_j - z \\ & x_j \le m_j y_j,\ \text{for } j \in C^+ \cup (N^- \setminus C^-) \\ & (x, y) \in R_+^{|C^+\cup(N^-\setminus C^-)|} \times B^{|C^+\cup(N^-\setminus C^-)|}. \end{aligned}$$

Next, we observe that the flows x_j, $j \in C^+ \setminus C^{++}$ and $m_j - x_j$, $j \in L^{--}$ behave similarly in the optimization. Note that we can assume, without loss of generality, that all these arcs are open, i.e., $y_j = 1$. For convenience, we aggregate these variables into a single variable $x = \sum_{j\in C^+\setminus C^{++}} x_j + \sum_{j\in L^{--}}(m_j - x_j)$. We get

$$
\begin{aligned}
f(z) = \min\ & d'' - x - \sum_{j\in C^{++}} x_j - \sum_{j\in C^{++}} (m_j - \lambda)(1 - y_j) - \sum_{j\in L^-} (m_j - \lambda y_j)\\
& x + \sum_{j\in C^{++}} x_j + \sum_{j\in L^-} (m_j - x_j) \le d'' - z\\
& x \le m\\
& x_j \le m_j y_j, \text{ for } j \in C^{++} \cup L^-\\
& (x, y) \in R_+^{|C^+\cup(N^-\setminus C^-)|} \times B^{|C^+\cup(N^-\setminus C^-)|}.
\end{aligned}
$$

Now observe that in any optimal solution either $x_j > m_j - \lambda$ and $y_j = 1$ or $x_j = 0$ and $y_j = 0$ for $j \in C^{++}$, $x_j = 0$ and $y_j = 0$ or $x_j = m_j$ and $y_j = 1$ for $j \in L^-$, $x \ge 0$. If we cannot set $x_j = m_j$ for $j \in C^{++}$ and $x_j = 0$ for $j \in L^-$, we should decrease the variables in C^{++} and increase the variables in L^- in the order $j_1, j_2, ..., j_r$ since $m_{j_i} > m_{j_{i+1}}$ for $i = 1, ..., r - 1$. Also observe that $0 \le x \le m$.

We proceed by giving a feasible solution for each of the six cases distinguished in the definition of f. From the above discussion on optimality conditions, it is easy to see that each of the given solutions is also optimal.

Since $\lambda = \sum_{j\in C^+} m_j - \sum_{j\in C^-} m_j - d$,

$$
\begin{aligned}
d'' &= d + \sum_{j\in N^-} m_j\\
&= \sum_{j\in C^+} m_j - \lambda - \sum_{j\in C^-} m_j + \sum_{j\in N^-} m_j\\
&= \sum_{j\in C^{++}} m_j + \sum_{j\in C^+\setminus C^{++}} m_j - \lambda + \sum_{j\in L^-} m_j + \sum_{j\in L^{--}} m_j\\
&= m + \sum_{j\in C^{++}} m_j + \sum_{j\in L^-} m_j - \lambda\\
&= m + M_r - \lambda.
\end{aligned}
$$

Since showing feasibility proceeds along the same lines in all cases, we only show it explicitly for Case 1.

Case 1. $M_i \le z \le M_{i+1} - \lambda$, $i = 0, \ldots, t - 1$.

If $j_k \in C^{++}$, $x_{j_k} = 0$, for $k \le i+1$, $x_{j_k} = m_{j_k}$ for $k > i+1$. If $j_k \in L^-$, $x_{j_k} = m_{j_k}$, for $k \le i + 1$, $x_{j_k} = 0$ for $k > i + 1$. $x = m$.

Then it is easy to see

$$
x + \sum_{j\in C^{++}} x_j + \sum_{j\in L^-} (m_j - x_j) = m + M_r - M_{i+1}.
$$

Since $d'' - z \geq m + M_r - \lambda - (M_{i+1} - \lambda) = m + M_r - M_{i+1}$, the solution is feasible. The objective value is

$$\begin{aligned} d'' - x - \sum_{j\in C^{++}} [x_j + (m_j - \lambda)(1 - y_j)] - \sum_{j\in L^-} (m_j - \lambda y_j) \\ = m + M_r - \lambda - m - \sum_{j\in C^{++}} m_j - \sum_{j\in L^-} m_j + (i+1)\lambda \\ = i\lambda. \end{aligned}$$

Case 2. $M_i - \lambda \leq z \leq M_i$, $i = 1, \dots, t-1$.

If $j_k \in C^{++} \setminus \{j_t\}$, $x_{j_k} = 0$, for $k \leq i$, $x_{j_k} = m_j$ for $k > i$. $x_{j_t} = mp - z - M_i - \lambda$. If $j_k \in L^-$, $x_{j_k} = m_j$ for $k \leq i$, $x_{j_k} = 0$ for $k > i$. $x = m$.

Case 3. $M_i - \lambda \leq z \leq M_i - \lambda + ml$, $i = t, \dots, r-1$.

If $j \in C^{++}$, $x_j = 0$. If $j_k \in L^-$, $x_{j_k} = m_j$ for $k \leq i$, $x_{j_k} = 0$ for $k > i$. $x = m + M_i - \lambda - z$.

Case 4. $M_i - \lambda + ml < z \leq M_i - \lambda + ml + \rho_i$, $i = t, \dots, r-1$.

If $j \in C^{++} \setminus \{j_t\}$, $x_j = 0$. $x_{j_t} = M_{i+1} - \lambda - z$. If $j_k \in L^-$, $x_{j_k} = m_j$ for $k \leq i+1$, $x_{j_k} = 0$ for $k > i+1$. $x = m$.

Case 5. $M_i - \lambda + ml + \rho_i \leq z \leq M_{i+1} - \lambda$, $i = t, \dots, r-1$.

If $j \in C^{++}$, $x_j = 0$. If $j_k \in L^-$, $x_{j_k} = m_j$ for $k \leq i+1$, $x_{j_k} = 0$ for $k > i+1$. $x = m$.

Case 6. $M_r - \lambda \leq z \leq d''$.

If $j \in C^{++}$, $x_j = 0$. If $j \in L^-$, $x_j = m_j$. $x = m + M_r - \lambda - z$.

□

Let

$$g(z) = \begin{cases} i\lambda & M_i \leq z \leq M_{i+1} - \lambda,\ i = 0, \dots, t-1 \\ z - M_i + i\lambda & M_i - \lambda \leq z \leq M_i,\ i = 1, \dots, t-1 \\ z - M_i + i\lambda & M_i - \lambda \leq z \leq M_i - \lambda + ml + \rho_i, \\ & i = t, \dots, r-1 \\ i\lambda & M_i - \lambda + ml + \rho_i \leq z \leq M_{i+1} - \lambda, \\ & i = t, \dots, r-1 \\ z - M_r + r\lambda & M_r - \lambda \leq z \leq d''. \end{cases}$$

Theorem 11. *The function g is a superadditive valid lifting function for f.*

Proof. If $\rho_i = 0$, then $\lambda - ml - \rho_i = \lambda - ml \geq 0$. If $\rho_i > 0$, then $\lambda - ml - \rho_i = \lambda - ml - [m_{i+1} - (mp - \lambda) - ml] = mp - m_{i+1} > 0$. Thus $g(z) \leq f(z)$.

The function g is superadditive because we can choose the parameters u_i and v_i of the function g_3 defined in the Appendix in such a way that $g_3 = g$ on $0 \leq z \leq d''$. To see this, observe that the form of both functions is identical, that the break points of g are $0, M_1 - \lambda, M_1, M_2 - \lambda, \ldots, M_t - \lambda, M_t - \lambda + ml + \rho_t, M_{t+1} - \lambda, \ldots, d''$, that the break points of g_3 are $0, u_1, W_1, W_1 + u_2, W_2, \ldots$, and that $W_i = W_{i-1} + u_i + v_i$. Now, it is a simple calculation to find values for u_i and v_i such that $g_3 = g$.

□

Corollary 2. *The lifting function $f(z)$ is superadditive if any one of the following conditions is satisfied.*

(1) $t = r$, i.e. $\min_{j \in C^{++}} m_j \leq \min_{j \in L^-} m_j$.
(2) $m \geq \lambda$, i.e. $\sum_{j \in C^+ \setminus C^{++}} m_j - \sum_{j \in L^{--}} m_j \geq \lambda$.
(3) $m_{t+1} - (mp - \lambda) - m > 0$ if $t \neq r$ and $m < \lambda$.

Proof. It is easy to check that if any one of the above conditions is satisfied, then $f(z) = g(z)$.

□

We can use g to perform sequence independent lifting on variables in $N^+ \setminus C^+$ and C^-. Lifting of variable pairs in $N^+ \setminus C^+$ is covered by Proposition 4, Case 1 and lifting of variable pairs in C^- is covered by Proposition 4, Case 4.

Theorem 12. *The lifted simple generalized flow cover inequality (LSGFCI)*

$$\sum_{j \in C^+} x_j + \sum_{j \in C^{++}} (m_j - \lambda)(1 - y_j) + \sum_{j \in N^+ \setminus C^+} \alpha_j x_j - \sum_{j \in N^+ \setminus C^+} \beta_j y_j$$
$$\leq d' - \sum_{j \in C^-} g(m_j)(1 - y_j) + \sum_{j \in L^-} \lambda y_j + \sum_{j \in L^{--}} x_j$$

where $(\alpha_j, \beta_j) = (0, 0)$ if $M_i \leq m_j \leq M_{i+1} - \lambda$ and $(\alpha_j, \beta_j) = (1, M_i - i\lambda)$ if $M_i - \lambda < m_j < M_i$, is valid for X.

Proof. The proof is similar to that of Theorem 8.

□

Note that f has a very good chance of being superadditive, since it is very likely that one of the conditions of Corollary 2 is satisfied. Recall that even if f is not superadditive, the use of the superadditive valid lifting function g may still give a facet.

Example 1. Consider the single-node flow model given below

$$X = \Big\{(x, y) \in R^4_+ \times B^4 : x_1 + x_2 - x_3 - x_4 \leq 8,$$
$$x_1 \leq 14y_1, x_2 \leq 10y_1, x_3 \leq 12y_3, x_4 \leq 2y_4\Big\}.$$

For the flow cover given by $C^+ = \{1\}$ and $C^- = \emptyset$, we get the following parameters: $\lambda = 6$, $C^{++} = \{1\}$, $L^- = \{3\}$, $L^{--} = \{4\}$, $r = 2$, $M_0 = 0$, $M_1 = 14$,

$M_2 = 26,\ m = 2,\ mp = 14,\ t = 1,\ ml = 2,\ \rho_1 = 2$. Figure 6 gives $f(z)$ and $g(z)$. The resulting LSGFCI is

$$x_1 - 8(1 - y_1) + x_2 - 8y_2 \le 8 + 8y_3 + x_4.$$

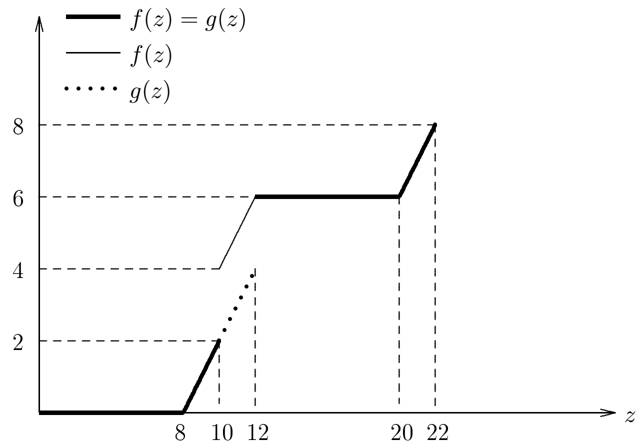

Fig. 6. $f(z)$ and $g(z)$

7. A computational study

In this section, we will compare the performance of the different variants of lifted flow cover inequalities discussed above, i.e., SGFCI, EGFCI, LFCI, and LSGFCI. However, before doing so, we address some important implementation issues. First, we discuss how the initial cover inequalities are derived. Second, we indicate how we select the lifting coefficients for a pair of variables if there are many choices, as is the case when we apply Case 1 and Case 3 of Theorem 4.

7.1. Initial flow cover

The initial flow cover $C = C^+ \cup C^-$ is obtained by solving the problem

$$\begin{aligned} \max &\left\{ \sum_{j \in N^+} (y_j^* - 1)\alpha_j + \sum_{j \in N^-} y_j^* \beta_j \right\} \\ \text{s.t.} & \sum_{j \in N^+} m_j \alpha_j - \sum_{j \in N^-} m_j \beta_j > d \\ & \alpha \in B^{|N^+|}, \beta \in B^{|N^-|}, \end{aligned}$$

where (x^*, y^*) is the current LP solution and where α and β represent incidence vectors of C^- and C^+ respectively.

This problem is a relaxation of the separation problem for simple generalized flow cover inequalities (3), in which it is assumed that $L^- = N^- \setminus C^-$ and $C^{++} = C^+$, and in which x^* is replaced by $m_j y_j^*$. For a more elaborate treatment we refer the reader to Section II.6.4 of Nemhauser and Wolsey [11]. Since the separation problem is a 0-1 knapsack problem, which is NP-Hard, we use its LP relaxation to solve it approximately.

A flow cover C is all that is required to define a flow cover inequality and a simple generalized flow cover inequality. However, for an extended generalized flow cover, we also need to determine sets L^+ and L^-. We use

$$L^+ = \left\{ j \in N^+ \setminus C^+ : x_j^* > \beta_j y_j^* \right\}$$

and

$$L^- = \left\{ j \in N^- \setminus C^- : -g(-m_j) y_j^* < x_j^* \right\},$$

where (x^*, y^*) is the current LP solution.

Whenever, there are many choices for the lifting coefficients of a variable pair, we pick those lifting coefficients that result in the largest violation for the current LP solution.

7.2. MINTO

We have used MINTO, a Mixed INTeger Optimizer [10], to implement the different variants of lifted flow cover inequalities. MINTO is a software package for the solution of mixed integer programming problems. It has many built-in features, such as preprocessing and probing, branching, and cut generation. In addition, MINTO allows the user to customize the code to take advantage of problem specific characteristics. Among the classes of cuts generated by MINTO are lifted cover inequalities and lifted GUB cover inequalities. MINTO was used for all our computational experiments and was invoked with run time options $-p1 - cikg - e$, i.e., simple preprocessing, cut generation for cover inequalities only, best-bound search of the tree, branching on a fractional variable closest to 0.5, and calling a primal heuristic based on recursive rounding every 25 nodes. All computational experiments were performed on an IBM RS/6000 model 550 using MINTO 2.0 on top of CPLEX 3.0.

7.3. Test set

Our test set consists of 26 mixed 0-1 integer programs, 11 are taken from MIPLIB [4, **?**], 5 are real-world problems that we obtained from different sources, and 10 are randomly generated capacitated facility location problems according to the scheme proposed by Cornuejols, Sridharen and Thizy [6].

MIPLIB has 28 mixed integer programming problems. Of the 28 problems, 9 are very easy, i.e., a plain LP based branch-and-bound algorithm solves them very fast, so we don't use them. Of the remaining 19 problems, 8 are very difficult. Although MINTO can solve most of them, some take quite a bit of time and some require other types of cuts as well, such as 0-1 knapsack cover inequalities. Furthermore, some have general integer variables and flow cover inequalities are most useful for mixed 0-1 integer programs. Therefore, we have decided not to use these either. That leaves us with 11 problems from MIPLIB.

Cornuejols et al. [6] developed a set of instances of capacitated facility location problems to be used to compare the performance of heuristics and relaxations for the capacitated facility location problem. Each instance is represented by a string 'cflp$vxxyy.z$', where v is the capacity level, xx the number of clients, yy the number of depots, and z the number of the instance in the series of test problems having the same values of v, xx, and yy. The computational results of Cornuejols et al. [6] and Aardal [1] show that instances with 50 clients, 33 depots, and small capacity levels are the most difficult ones. Therefore, we included 10 such instances with capacity levels 1 and 2.

Table 1 gives a summary of the test problems. Note that several of these instances are well-known to be hard.

7.4. Computational results for LFCI, SGFCI, LSGFCI, EGFCI

There are many alternative ways of using the flow cover inequalities developed in the previous sections. We may choose to generate only one specific type or we may choose to generate several types and select the most violated one among them. Since the goal of our experiments is to study the effectiveness of each type of flow cover inequality, we apply them separately with one exception. We also include the combination of SGFCI and EGFCI which is incorporated in MINTO 2.0. We did experiment with using other combinations, but combining different types of flow cover inequalities does not yield better results.

Table 2 shows the computational results for LSGFCI. This choice yields the best performance. For all other choices, we present only the differences in the performance with respect to the best choice.

Table 3 shows the computational results for SGFCI, Table 4 for EGFCI, Table 5 for LFCI, Table 6 for combined SGFCI and EGFCI. We see that with EGFCI we cannot solve the problems "khb05250", "modglob" and "fiber" within 10000 nodes. The performance of SGFCI is much better than that of EGFCI. The reason, we believe, is that the chance that SGFCI defines a facet is high, while the chance that EGFCI defines a facet is small. Combined SGFCI and EGFCI does not produce better results than SGFCI.

Using a superadditive valid lifting function to lift SGFCI to LSGFCI reduces total CPU-time by about 40% and always reduces the number of evaluated nodes of the branching tree. Therefore, we conclude that there is a clear computational advantage in applying the sequence independent lifting functions developed here to improve the performance of flow cover inequalities in a branch-and-cut system for 0-1 mixed integer programs.

Table 1. Test problem summary

Name	VARS	0-1 VARS	ROWS	Z_{lp}	Z_{ip}
egout	141	55	98	149.589	568.101
fixnet3	878	378	478	40717.018	51973
fixnet4	878	378	478	4257.97	8936
fixnet6	878	378	478	1200.88	3983
khb05250	1350	24	101	95919464.0	106940226
mod013	96	48	62	256.02	280.95
modglob	422	98	291	20430947.0	20740508
rentacar	9557	55	6803	28806137.644	30356761
rgn	180	100	24	48.7999	82.1999
set1al	712	240	492	11145.63	15869.75
set1cl	712	240	492	1671.96	6484.25
soda1	675	75	1182	13818.58	22606.82
soda2	675	75	1182	13818.54	22603.16
fiber	1298	44	363	156082.52	405935.18
utrans.2	240	120	150	169.786	239.217
utrans.3	284	142	182	313.798	432.285
cflp15033.1	1716	33	1767	26786.65	27189
cflp15033.2	1716	33	1767	27564.55	27891
cflp15033.3	1716	33	1767	28142.19	28194
cflp15033.4	1716	33	1767	28777.61	29080
cflp15033.5	1716	33	1767	28494.22	28963
cflp25033.1	1716	33	1767	22230.69	22570
cflp25033.2	1716	33	1767	24079.72	24362
cflp25033.3	1716	33	1767	23230.48	23591
cflp25033.4	1716	33	1767	24551.45	24718
cflp25033.5	1716	33	1767	23640.82	23962

Table 2. Computational results for LSGFCI

Problem	Time	Nodes	Zroot	LPs	Cuts
egout	0.82	25	556.4	27	14
fixnet3	3.36	5	51880.8	18	83
fixnet4	34.64	111	8307.8	170	190
fixnet6	122.98	465	3507.4	545	169
khb05250	6.16	13	106608880.0	35	122
mod013	2.46	57	267.3	80	54
modglob	132.22	651	20662084.0	704	366
rentacar	253.57	35	29219168.0	57	60
rgn	91.66	2881	64.6	2922	74
set1al	5.55	21	15867.2	23	400
set1cl	3.57	1	6484.2	2	400
soda1	31.82	41	17644.7	57	147
soda2	41.99	83	17644.5	106	164
fiber	71.47	137	381837.8	310	360
utrans.2	5.66	53	233.6	83	88
utrans.3	36.97	449	416.4	506	144
cflp15033.1	55.92	39	26835.1	92	78
cflp15033.2	75.11	63	27786.5	116	78
cflp15033.3	7.87	1	28194.0	2	2
cflp15033.4	109.43	87	28784.3	175	130
cflp15033.5	30.85	29	28877.5	57	45
cflp25033.1	25.01	11	22245.2	26	23
cflp25033.2	93.12	47	24081.6	108	82
cflp25033.3	55.78	27	23438.7	62	48
cflp25033.4	27.37	21	24678.8	34	18
cflp25033.5	62.64	61	23656.5	109	69
total	1388.00	5414	157253423.0	6426	3408

Table 3. Computational results for SGFCI

Problem	ΔTime	ΔNodes	ΔZroot	ΔLPs	ΔCuts
egout	-0.07	0	0.0	0	0
fixnet3	-0.14	0	0.0	0	0
fixnet4	42.99	208	15.0	228	14
fixnet6	89.88	338	0.0	378	46
khb05250	-0.16	0	0.0	0	-1
mod013	0.26	12	-0.1	13	6
modglob	-29.39	0	15684.0	-5	-65
rentacar	-1.73	0	0.0	0	0
rgn	77.64	328	0.0	517	140
set1al	0.89	0	0.0	0	0
set1cl	-0.05	0	0.0	0	0
soda1	4.22	18	-68.7	20	-10
soda2	-4.62	8	-56.6	1	-18
fiber	10.07	24	-504.5	25	33
utrans.2	-0.48	0	-0.3	-1	-5
utrans.3	8.60	102	-0.4	115	5
cflp15033.1	146.04	184	-36.9	269	110
cflp15033.2	97.55	112	-189.2	197	119
cflp15033.3	0.00	0	0.0	0	0
cflp15033.4	149.53	208	0.0	317	133
cflp15033.5	142.11	198	-288.7	323	165
cflp25033.1	77.42	76	0.0	116	59
cflp25033.2	133.64	144	0.0	222	120
cflp25033.3	-18.29	6	-183.1	-10	-22
cflp25033.4	52.41	46	-107.6	77	40
cflp25033.5	83.50	88	-0.0	147	81
total	1061.82	2100	14262.9	2949	950

Table 4. Computational results for EGFCI

Problem	ΔTime	ΔNodes	ΔZroot	ΔLPs	ΔCuts
egout	0.08	6	-17.6	10	1
fixnet3	9.46	82	-1465.1	91	-60
fixnet4	38.85	326	-602.7	308	-149
fixnet6	195.84	1014	-314.2	993	-109
khb05250	1932.90	9987	-10269456.0	10668	258
mod013	2.75	50	-2.5	77	19
modglob	1443.24	9349	-230954.0	9538	-204
rentacar	54.36	12	-290788.0	-10	-60
rgn	68.31	1850	-15.8	1986	30
set1al	-1.45	0	0.0	0	-200
set1cl	-1.17	0	0.0	0	-200
soda1	7.91	22	0.9	26	5
soda2	-4.87	-12	0.9	-16	-12
fiber	3498.17	9863	-213965.7	11767	437
utrans.2	126.03	1116	-17.1	1234	105
utrans.3	157.91	1538	-14.5	1632	38
cflp15033.1	301.18	444	-47.7	686	218
cflp15033.2	70.26	156	-209.3	240	59
cflp15033.3	0.30	0	0.0	0	-1
cflp15033.4	165.67	250	0.0	382	90
cflp15033.5	275.34	512	-380.7	792	261
cflp25033.1	101.11	90	94.2	142	44
cflp25033.2	300.32	196	0.4	405	188
cflp25033.3	28.76	70	-186.8	86	3
cflp25033.4	53.06	46	-107.6	78	27
cflp25033.5	165.27	160	4.4	270	89
total	8989.59	37127	-11008444.4	41385	877

Table 5. Computational results for LFCI

Problem	ΔTime	ΔNodes	ΔZroot	ΔLPs	ΔCuts
egout	1.80	106	-36.3	117	6
fixnet3	9.47	82	-1465.1	91	-60
fixnet4	38.42	326	-602.7	308	-149
fixnet6	195.35	1014	-314.2	993	-109
khb05250	1922.65	9987	-10269456.0	10668	258
mod013	2.17	74	-3.0	92	6
modglob	1436.00	9349	-230954.0	9538	-204
rentacar	53.41	12	-290788.0	-10	-60
rgn	66.73	1850	-15.8	1986	30
set1al	-1.46	0	0.0	0	-200
set1cl	-1.21	0	0.0	0	-200
soda1	4.43	18	-68.7	20	-10
soda2	-4.46	8	-56.6	1	-18
fiber	4272.93	9863	-193520.2	12992	1068
utrans.2	150.47	1484	-17.6	1624	92
utrans.3	234.81	2064	-15.9	2206	53
cflp15033.1	243.30	390	-47.7	569	154
cflp15033.2	69.22	156	-209.3	238	58
cflp15033.3	0.02	0	0.0	0	-1
cflp15033.4	161.59	250	0.0	381	89
cflp15033.5	271.87	512	-380.7	792	261
cflp25033.1	128.71	138	-7.3	203	57
cflp25033.2	232.51	188	0.0	387	178
cflp25033.3	23.60	66	-186.8	78	-1
cflp25033.4	46.81	44	-107.6	73	24
cflp25033.5	125.02	132	-0.0	218	65
total	9684.16	38113	-10988253.3	43565	1387

Table 6. Computational results for combining SGFCI and EGFCI

Problem	ΔTime	ΔNodes	ΔZroot	ΔLPs	ΔCuts
egout	-0.07	0	0.0	0	0
fixnet3	-0.14	0	0.0	0	3
fixnet4	61.44	282	13.2	302	21
fixnet6	87.23	362	-0.2	390	40
khb05250	-0.06	0	0.0	0	2
mod013	0.37	12	1.9	12	10
modglob	-14.13	2	15684.0	0	-60
rentacar	2.13	0	0.0	0	0
rgn	79.91	328	0.0	517	140
set1al	-0.11	0	0.0	0	0
set1cl	0.01	0	0.0	0	0
soda1	8.59	22	0.9	26	5
soda2	-4.65	-12	0.9	-16	-12
fiber	10.15	34	85.0	48	76
utrans.2	2.43	24	0.0	39	18
utrans.3	13.24	100	-0.4	120	27
cflp15033.1	187.43	192	-36.9	295	134
cflp15033.2	99.64	112	-189.2	198	120
cflp15033.3	0.22	0	0.0	0	0
cflp15033.4	149.43	208	0.0	317	133
cflp15033.5	141.78	198	-288.7	323	165
cflp25033.1	53.87	42	94.2	83	63
cflp25033.2	159.00	146	0.4	247	140
cflp25033.3	-18.80	6	-183.1	-10	-22
cflp25033.4	53.19	42	-107.6	73	42
cflp25033.5	127.47	136	4.4	215	104
total	1199.57	2236	15078.8	3179	1149

Appendix

Given

$$l > 0,$$
$$u_i \geq 0, u_i \geq u_{i+1}, \quad \textit{for } i = 1, 2, \ldots, \infty,$$
$$v_i > 0, v_i \geq v_{i+1}, \quad \textit{for } i = 1, 2, \ldots, \infty,$$

let

$$w_i = u_i + v_i, \quad \textit{for } i = 1, 2, \ldots, \infty,$$
$$W_h = \sum_{i=1}^{h} w_i, \quad \textit{for } h = 1, 2, \ldots, \infty.$$

We define

$$g_1(z) = \begin{cases} hl & hw_1 \leq z \leq hw_1 + u_1, \ h \text{ is an integer} \\ hl + l[z - (hw_1 + u_1)]/v_1 & hw_1 + u_1 < z < (h+1)w_1, \ h \text{ is an integer} \end{cases}$$

$$g_2(z) = \begin{cases} 0 & z = 0 \\ hl & W_h < z \leq W_h + u_{h+1}, \ h = 0, 1, \ldots, \infty \\ l[h + 1 - (W_{h+1} - z)/v_1] & W_h + u_{h+1} < z \leq W_{h+1}, \ h = 0, 1, \ldots, \infty \end{cases}$$

$$g_3(z) = \begin{cases} 0 & z = 0 \\ hl & W_h < z \leq W_h + u_{h+1}, \ h = 0, 1, \ldots, \infty \\ l[h + (z - W_h - u_{h+1})/v_1] & W_h + u_{h+1} < z \leq W_{h+1}, \ h = 0, 1, \ldots, \infty \end{cases}$$

The functions $g_i(z)$ for $i = 1, 2, 3$ are shown in Figures 7 and 8. Let Z_i be the region where $g_i(z)$ is defined, then $Z_1 = (-\infty, +\infty)$ and $Z_i = [0, +\infty)$ for $i = 2, 3$. Obviously, the functions $g_i(z)$, $i = 1, 2, 3$ are nondecreasing in z. Let

$$D_i = \max\{g_i(z_1) + g_i(z_2) - g_i(z_1 + z_2) : z_1, z_2, z_1 + z_2 \in Z_i\},$$

then $g_i(z)$ is superadditive if and only if $D_i \leq 0$.

Theorem 13. *The functions $g_i(z)$ for $i = 1, 2, 3$ are superadditive.*

Proof for $g_1(z)$. First, we show that to compute D_1, we only need to consider $z_1 = hw_1$. Suppose $hw_1 < z_1 \leq hw_1 + u_1$. Then, since $g_1(z)$ is nondecreasing, $g_1(z_1 + z_2) \geq g_1(hw_1 + z_2)$. Consequently

$$\begin{aligned} g_1(z_1) + g_1(z_2) - g_1(z_1 + z_2) &= hl + g_1(z_2) - g_1(z_1 + z_2) \\ &\leq g_1(hw_1) + g_1(z_2) - g_1(hw_1 + z_2). \end{aligned}$$

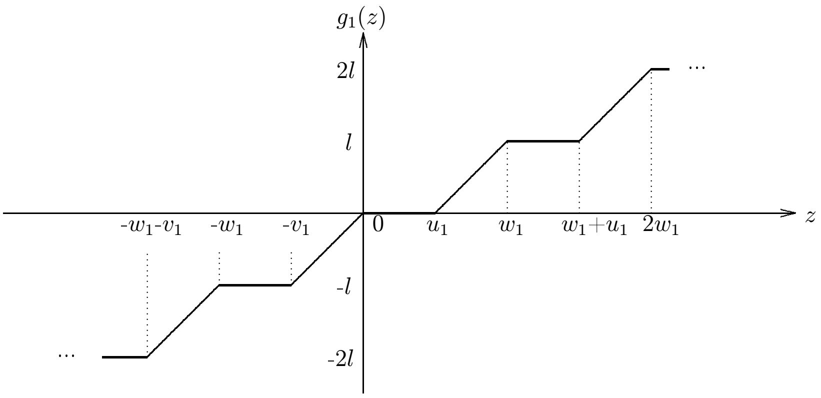

Fig. 7. Function $g_1(z)$

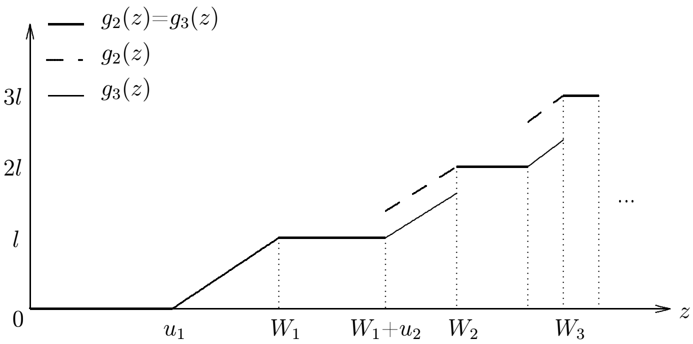

Fig. 8. Functions $g_2(z)$ and $g_3(z)$

Suppose $hw_1 + u_1 < z_1 < (h+1)w_1$. Then $g_1(z + \Delta z) \leq g_1(z) + (l/v_1)\Delta z$. Consequently

$$\begin{aligned}
&g_1(z_1) + g_1(z_2) - g_1(z_1 + z_2)\\
&= hl + l(z_1 - hw_1 - u_1)/v_1 + g_1(z_2) - g_1(z_1 + z_2)\\
&= (h+1)l + l(z_1 - hw_1 - u_1 - v_1)/v_1 + g_1(z_2) - g_1(z_1 + z_2)\\
&= (h+1)l + g_1(z_2) - g_1(z_1 + z_2) - l[(h+1)w_1 - z_1]/v_1\\
&\leq (h+1)l + g_1(z_2) - g_1[(h+1)w_1 + z_2]\\
&= g_1[(h+1)w_1] + g_1(z_2) - g_1[(h+1)w_1 + z_2].
\end{aligned}$$

Secondly, let $z_1 = h_1 w_1$ and $z_2 = h_2 w_2$, then

$$\begin{aligned}
g_1(z_1) + g_2(z_2) - g(z_1 + z_2) &= lh_1 + lh_2 - g[(h_1 + h_2)w_1]\\
&= lh_1 + lh_2 - l(h_1 + h_2)\\
&= 0.
\end{aligned}$$

Hence $D_1 = 0$ and $g_1(z)$ is superadditive.

Proof for $g_2(z)$. Suppose $W_h < z_1 \le W_h + u_{h+1}$. Since $g_2(z_1)$ is nondecreasing, $g_2(z_1 + z_2) \ge g_2(W_h + z_2)$. Consequently

$$\begin{aligned} g_2(z_1) + g_2(z_2) - g_2(z_1 + z_2) &= hl + g_2(z_2) - g_2(z_1 + z_2) \\ &\le g_2(W_h) + g_2(z_2) - g_2(W_h + z_2). \end{aligned}$$

Therefore, to compute D_2, we only need to consider the case where $W_{h_1} + u_{h_1+1} < z_1 \le W_{h_1+1}$ and $W_{h_2} + u_{h_2+1} < z_2 \le W_{h_2+1}$. Since

$$\begin{aligned} z_1 + z_2 &> W_{h_1} + u_{h_1+1} + W_{h_2} + u_{h_2+1} \\ &= \sum_{i=1}^{h_1} w_i + \sum_{i=1}^{h_2} w_i + u_{h_1+1} + u_{h_2+1} \\ &\ge \sum_{i=1}^{h_1} w_i + \sum_{i=h_1+1}^{h_1+h_2} w_i + u_{h_1+1} + u_{h_2+1} \\ &= W_{h_1+h_2} + u_{h_1+1} + u_{h_2+1} \\ &\ge W_{h_1+h_2} + u_{h_1+h_2+1}, \end{aligned}$$

we have the following cases.

Case 1. $z_1 + z_2 \ge W_{h_1+h_2+2}$.

Then

$$\begin{aligned} g_2(z_1 + z_2) &\ge (h_1 + h_2 + 2)l, \\ g_2(z_1) &\le (h_1 + 1)l, \\ g_2(z_2) &\le (h_2 + 1)l. \end{aligned}$$

So

$$g_2(z_1) + g_2(z_2) - g_2(z_1 + z_2) \le 0.$$

Case 2. $W_{h_1+h_2+1} + u_{h_1+h_2+2} \le z_1 + z_2 < W_{h_1+h_2+2}$.

Then

$$\begin{aligned} &g_2(z_1) + g_2(z_2) - g_2(z_1 + z_2) \\ &\quad \le l[h_1 + 1 - (W_{h_1+1} - z_1)/v_1] + l[h_1 + 1 - (W_{h_2+1} - z_1)/v_1] \\ &\qquad -l[(h_1 + h_2 + 2) - (W_{h_1+h_2+2} - z_1 - z_2)/v_1] \\ &\quad = -(l/v_1)(W_{h_1+1} + W_{h_2+1} - W_{h_1+h_2+2}) \\ &\quad \le 0. \end{aligned}$$

Case 3. $W_{h_1+h_2+1} \le z_1 + z_2 < W_{h_1+h_2+1} + u_{h_1+h_2+2}$.

Because $g_2(z)$ is nondecreasing and the result for Case 2, we have

$$\begin{aligned} &g_2(z_1) + g_2(z_2) - g_2(z_1 + z_2) \\ &\quad = g_2(z_1) + g_2(z_2) - g_2(W_{h_1+h_2+1} + u_{h_1+h_2+2}) \\ &\quad \le g_2(z_1 + W_{h_1+h_2+1} + u_{h_1+h_2+2} - z_1 - z_2) + g_2(z_2) - g_2(W_{h_1+h_2+1} + u_{h_1+h_2+2}) \\ &\quad \le 0. \end{aligned}$$

Case 4. $W_{h_1+h_2} + u_{h_1+h_2+1} \leq z_1 + z_2 < W_{h_1+h_2+1}$.

Then

$$\begin{aligned}
&g_2(z_1) + g_2(z_2) - g_2(z_1 + z_2)\\
&\leq l[h_1 + 1 - (W_{h_1+1} - z_1)/v_1] + l[h_1 + 1 - (W_{h_2+1} - z_1)/v_1]\\
&\quad -l[(h_1 + h_2 + 1) - (W_{h_1+h_2+1} - z_1 - z_2)/v_1]\\
&= l - (l/v_1)(W_{h_1+1} + W_{h_2+1} - W_{h_1+h_2+1})\\
&= l - (l/v_1)(u_1 + v_1) - (l/v_1)(\sum_{i=2}^{h_1+1} w_i + \sum_{i=1}^{h_2+1} w_i - \sum_{i=1}^{h_1+h_2+1} w_i)\\
&\leq -(l/v_1) \sum_{i=2}^{h_1+1} (w_i - w_{i+h_1})\\
&\leq 0.
\end{aligned}$$

Hence $D_2 \leq 0$ and $g_2(z)$ is superadditive.

Proof for $g_3(z)$. Similar to the proof for $g_2(z)$, we find that to compute D_3, we only need to consider the case where $W_{h_1}+u_{h_1+1} < z_1 \leq W_{h_1+1}$ and $W_{h_2}+u_{h_2+1} < z_2 \leq W_{h_2+1}$.

Case 1. $z_1 + z_2 \in [W_h, W_h + u_{h+1}]$.

Then $g_3(z_1 + z_2) = hl = g_2(z_1 + z_2)$. Since $g_3(z) \leq g_2(z), \forall z \in [0, +\infty)$,

$$\begin{aligned}
g_3(z_1) + g_3(z_2) - g_3(z_1 + z_2) &\leq g_2(z_1) + g_2(z_2) - g_3(z_1 + z_2)\\
&= g_2(z_1) + g_2(z_2) - g_2(z_1 + z_2)\\
&= 0.
\end{aligned}$$

Case 2. $z_1 + z_2 \in (W_h + u_{h+1}, W_{h+1})$ and $z_1 - W_{h_1} - u_{h_1+1} + z_2 - W_{h_2} - u_{h_2+1} \geq z_1 + z_2 - W_h - u_{h+1}$.

It is easy to see that if we fix z_2 and slightly decrease z_1, then $g_3(z_1)$ and $g_3(z_1+z_2)$ will decrease by the same amount. It is also easy to see that if we decrease z_1 and z_2 appropriately, we can decrease $z_1 + z_2$ to W_h without changing $g_3(z_1) + g_3(z_2) - g_3(z_1 + z_2)$. By the result for Case 1, we have

$$g_3(z_1) + g_3(z_2) - g_3(z_1 + z_2) \leq 0.$$

Case 3. $z_1 + z_2 \in (W_h + u_{h+1}, W_{h+1})$ and $z_1 - W_{h_1} - u_{h_1+1} + z_2 - W_{h_2} - u_{h_2+1} < z_1 + z_2 - W_h - u_{h+1}$.

Using the same idea that was used in Case 2, we can decrease z_1 and z_2 to $W_{h_1}+u_{h_1+1}$ and $W_{h_2} + u_{h_2+1}$ respectively, without changing $g_3(z_1) + g_3(z_2) - g_3(z_1 + z_2)$. This implies that we only need to consider the case $z_1 = W_{h_1} + u_{h_1+1}$, $z_2 = W_{h_2} + u_{h_2+1}$ and $z_1 + z_2 \in (W_h + u_{h+1}, W_{h+1})$. Since $g_2(z_1) + g_2(z_2) - g_2(z_1 + z_2) \leq 0$,

$$\begin{aligned}
0 &\geq \lceil (g_2(z_1) + g_2(z_2) - g_2(z_1 + z_2))/l \rceil l\\
&= g_2(z_1) + g_2(z_2) - \lfloor g_2(z_1 + z_2)/l \rfloor l\\
&= g_3(z_1) + g_3(z_2) - \lfloor g_3(z_1 + z_2)/l \rfloor l\\
&\geq g_3(z_1) + g_3(z_2) - g_3(z_1 + z_2).
\end{aligned}$$

Hence $D_3 \leq 0$ and $g_3(z)$ is superadditive.

□

Acknowledgements. This research was supported by US Army Research Office DAAH04-94-G-0017, NSF Grant No. DDM-9115768, and NSF Grant No. DMI-9410102

References

1. Aardal, K. (1994): Capacitated facility location: separation algorithms and computational experience. Report CCOP 261, Cornell University, Ithaca
2. Balas, E., Ceria, S., Cornuejols, G. (1993): A lift-and-project cutting plane algorithm for mixed 0-1 programs. Math. Program. **58**, 295–324
3. Balas, E., Ceria, S., Cornuejols, G., Natraj, N. (1996): Gomory cuts revisited. Oper. Res. Lett. **19**, 1–9
4. Bixby, R.E., Boyd, E.A., Dadmehr, S.S., Indovina, R.R. (1992): The MIPLIB Mixed Integer Programming Library. Technical Report R92-36, Rice University
5. Bixby, R.E., Ceria, S., McZeal, C.M., Savelsbergh, M.W.P. (1996): An Update Mixed Integer Programming Library: MIPLIB 3.0. Report LEC-96-03, Georgia Institute of Technology, Atlanta
6. Cornuejols, G., Sridharan, R., Thizy, J.-M. (1991): A comparison of heuristics and relaxations for the capacitated plant location problem. Eur. J. Oper. Res. **50**, 280–297
7. Crowder, H., Johnson, E.L., Padberg, M.W. (1983): Solving Large Scale Zero-One Linear Programming Problems. Oper. Res. **31**, 803–834
8. Gu, Z., Nemhauser, G.L., Savelsbergh, M.W.P. (1994): Lifted Cover Inequalities for 0-1 Integer Programs: Computation. INFORMS Journal on Computing. To appear
9. Gu, Z., Nemhauser, G.L., Savelsbergh, M.W.P. (1995): Sequence Independent Lifting. Report LEC-95-08, Georgia Institute of Technology, Atlanta
10. Nemhauser, G.L., Savelsbergh, M.W.P., Sigismondi, G.C. (1994): MINTO, a Mixed INTeger Optimizer. Oper. Res. Lett. **15**, 47–58
11. Nemhauser , G.L., Wolsey, L.A. (1988): Integer and Combinatorial Optimization. Wiley, New York
12. Padberg, M.W., Van Roy, T.J., Wolsey, L.A. (1985): Valid Linear Inequalities for Fixed Charge Problems. Oper. Res. **33**, 842–861
13. Van Roy, T.J., Wolsey, L.A. (1986): Valid Inequalities for Mixed 0-1 Programs. Discrete Appl. Math. **14**, 199–213
14. Van Roy, T.J., Wolsey, L.A. (1987): Solving mixed integer programming problems using automatic reformulation. Oper. Res. **35**, 45–57
15. Wolsey, L.A. (1989): Strong formulations for mixed integer programming: a survey. Math. Program. **45**, 173–191

■ In the proof of corollary 2 you referred to Proposition 4. I think it should be Theorem 4. ■