P. Avella, M. Boccia, S. Mattia

COMPUTATIONAL TESTING OF LIFTED COVER INEQUALITIES WITH A SINGLE CONTINUOUS VARIABLE

R. 5, 2013

Pasquale Avella, Maurizio Boccia – Dipartimento di Ingegneria, Università del Sannio, Piazza Roma, 82100 Benevento (Italy). E-mail: {avella, maboccia}@unisannio.it.

Sara Mattia – Istituto di Analisi dei Sistemi ed Informatica (IASI), Consiglio Nazionale delle Ricerche (CNR), viale Manzoni 30, 00185 Roma, Italy. Email: sara.mattia@iasi.cnr.it.
Abstract

Reformulating Mixed-Integer Programming (MIP) problems by cutting planes in order to speed up the solution process is a common practice. All the best known MIP solvers now offer built-in separation routines for generating inequalities to refine the formulation and speed-up the optimization. Mixed-Integer Rounding (MIR) cuts are recognized among the most effective cutting planes to reformulate general MIP problems. They can be derived from the binary knapsack set with a single continuous variable. Here we consider another family of cuts derived from the same knapsack set, namely the Lifted Cover Inequalities with a Single Continuous Variable (LCISCV), and present an effective separation procedure, LCISCV-SEP. To validate the effectiveness of LCISCV-SEP, we exhibit a set of benchmark instances from MIPLIB 2010, where LCISCV-SEP leads to significant improvements in the lower bound, with little additional computational effort.
1. Introduction

Reformulating Mixed-Integer Programming (MIP) problems by cutting planes in order to speed up the solution process is a common practice. All the best known MIP solvers now offer built-in separation routines for generating inequalities to refine the formulation and speed-up the optimization. Cutting planes that can be used to strengthen a given MIP problem, can be roughly classified into two categories: ad-hoc and general purpose cutting planes. Ad-hoc cutting planes only apply to a specific structured problem, as they rely on the combinatorial properties of the considered problem. General purpose cutting planes are valid inequalities that can be used on any problem. They include Chvátal-Gomory (CG) cuts, Gomory fractional cuts, lift-and-project cuts, disjunctive cuts, split cuts [9, 10, 12, 13, 14, 17] and cutting planes derived from combinatorial structures. Cutting planes from structures are general purpose cutting planes that can be derived from a given submatrix (structure) of the constraint matrix, defining a polyhedron for which valid, and possibly facet defining, inequalities are known and can be separated in a reasonable amount of time. Examples of structures that can be used to generate valid inequalities for MIP problems are (capacitated) network flow blocks, (multidimensional) knapsack sets and set packing submatrices. Inequalities from combinatorial structures include Clique, (Lifted) Cover, GUB Cover, Flow Cover and Mixed-Integer Rounding (MIR) inequalities [1, 3, 19, 20, 23, 26, 27]. In principle, any structure can be used to derive cuts for general problems, provided that the structure can be recognized, possibly reordering and aggregating rows and columns in the constraint matrix, and that the corresponding polyhedron meets the hypotheses above, i.e. known valid inequalities and availability of separation algorithms. However, it is easy to see that the simpler the structure is, the larger the class of problems where the corresponding cuts can be used becomes. Knapsack and mixed knapsack sets are among the simplest structures, therefore knapsack based inequalities are the most popular cutting planes from structures for MIP problem. They can be used on a large variety of problems, as single (or aggregated) rows of any formulation can be seen as knapsack sets: for this reason, several variants of the knapsack problem have been investigated.

1.1. Literature review

Many authors studied the mixed integer knapsack set. Formally, let $a^T y + g^T s \leq b$ be a linear inequality, where $y \in \mathbb{Z}_n^+$ are integer variables and $s \in \mathbb{R}_p^+$ are continuous variables, $a \in \mathbb{R}^n$, $g \in \mathbb{R}^p$ are the corresponding coefficients and $b \in \mathbb{R}$ is the right-hand-side. Let $u \in \mathbb{R}^n_+$ and $v \in \mathbb{R}^p_+$ be upper bounds for variables $y$ and $s$, respectively. Both $u$ and $v$ may have entries equal to infinite, so that the variables are not necessarily bounded. The most general mixed integer knapsack set $Y^{MI}$ can be defined as follows:

$$Y^{MI} = \{(y, s) \in \mathbb{Z}_n^+ \times \mathbb{R}_p^+ : a^T y + g^T s \leq b, y \leq u, s \leq v\},$$

When $y$ are binary variables then we get the mixed binary knapsack set $Y^{MB}$:

$$Y^{MB} = \{(y, s) \in \{0, 1\}^n \times \mathbb{R}_p^+ : a^T y + g^T s \leq b, s \leq v\},$$

A simpler mixed integer knapsack structure - that we will be the main object of this paper - is the mixed binary knapsack set with a single unbounded continuous variable $s$, that we denote by $Y$ and that can be defined as follows.

$$Y = \{(y, s) \in \{0, 1\}_n^+ \times \mathbb{R}_+ : a^T y \leq b + s\},$$
where we can assume $a \in \mathbb{R}_+^n$ and $b \in \mathbb{R}_+$ [24]. It quite obvious that if there are no continuous variables ($p = 0$), then we have the traditional binary knapsack problem $Y^B$.

Classic results for the knapsack problem can be found in [8, 11, 28, 31]. Additional references can be found in [21, 25]. Atamtürk [2, 3] studied facets of $\text{conv}(Y^{MI})$, with and without upper bounds on continuous variables. Richard et al. [30] presented a theoretical study for the lifting of the continuous variables for the mixed binary knapsack set $Y^{MB}$ with bounded continuous variables. Narisetty et al. [26] presented lifted inequalities for MIP problems derived from the knapsack set $Y^{MB}$ defined using the simplex tableaux instead of using the constraints of the original formulation. The polyhedral structure of the mixed binary knapsack set with a single unbounded continuous variable was studied by Marchand and Wolsey [24], who showed that it has a rich structure and that the corresponding cutting planes can be successfully used in the solution of MIP problems. Atamtürk and Günliük [4] and Atamtürk and Kianfar [5] presented facets for the mixed integer knapsack set with a single unbounded continuous variable. Implementation issues related to the use of lifted cover inequalities and lifted GUB cover inequalities derived from knapsack set $Y^B$ for solving MIP problems were investigated in [19]. In [22] Kapoor and Letchford presented exact and heuristic separation routines for several classes of valid inequalities (Lifted Cover, Extended Cover, Weight and Lifted Pack inequalities) for knapsack set $Y^B$ and reported computational results on MIPLIB 2003 and OR-Lib instances. Implementations of separation procedures for MIR inequalities for knapsack set $Y$ were presented by Goncalves and Ladanyi [18] and Wolter [32]. Fukasawa and Goycoolea [16] presented an exact separation procedure consisting of a branch-and-bound algorithm using exact arithmetic for finding cutting planes for general MIP problems using knapsack set $Y^{MI}$. In particular, MIR inequalities proved to be very effective in closing the gap, but the proposed exact approach is computationally heavy. Avelle et al. [6] studied an exact separation procedure for $\text{conv}(Y)$, and performed a computational study on a wide set of MIP instances. The proposed exact separation procedure was able to significantly raise the lower bounds given by Lifted Cover and MIR inequalities for several benchmark instances, but computation times still remain too large to be of practical use.

1.2. This paper

In this paper we investigate from a computational standpoint another family of inequalities derivable from $\text{conv}(Y)$, that we call Lifted Cover Inequalities with a Single Continuous Variable (LCISCV), to reformulate general MIP problems. For this new family we present an effective separation procedure, LCISCV-SEP. To demonstrate its effectiveness, LCISCV-SEP has been tested on benchmark instances of the MIPLIB 2010. Computational experiments show that LCISCV can improve the lower bound provided by MIR inequalities, with little additional computational effort.

The remainder of the paper is organized as follows. In Section 2 we formally define the family of the Lifted Cover inequalities with a Single Continuous Variable and present the details of the separation procedure LCISCV-SEP. In Section 3 we describe the cutting plane algorithm based on LCISCV-SEP used in our computational experience to solve MIP problems. In Section 4 we provide computational results. In Section 5 conclusions are discussed.
2. Lifted Cover Inequalities with a Single Continuous Variable

As pointed out in [24], one way to generate valid inequalities of \( \text{conv}(Y) \) is to adapt inequalities from the binary knapsack polytope obtained fixing the continuous variable \( s \) to a given value \( \bar{s} \). Let \( b_\bar{s} = b + \bar{s} \) and let \( Y_{\bar{s}} = \{ (y, s) \in Y : s = \bar{s} \} = \{ y \in \{0, 1\}_+^n \times \mathbb{R}_+ : a^T y \leq b_\bar{s} \} \) be the binary knapsack set obtained by setting \( s = \bar{s} \). Given a set \( C \subseteq \{1, \ldots, n\} \) such that \( \sum_{j \in C} a_j > b_\bar{s} \) (cover), the corresponding cover inequality is the inequality below.

\[
\sum_{j \in C} y_j \leq |C| - 1
\]

\( C \) is minimal if none of its proper subsets is also a cover. It is possible to strengthen a cover inequality by recomputing (lifting) the coefficients for the missing variables [8, 31], obtaining the lifted cover inequality below, that is valid for \( \text{conv}(Y_{\bar{s}}) \).

\[
\sum_{j \in C} y_j + \sum_{j \notin C} \eta_j y_j \leq |C| - 1
\]

For considerations about lifting see [15, 22, 33] and references therein. If we also lift the continuous variable \( s \), we obtain the Lifted Cover Inequality with a Single Continuous Variable, that can be written as below and is valid for \( \text{conv}(Y) \).

\[
\alpha^T y \leq \beta + \gamma s
\]

Hence, LCISCV can be constructively obtained as follows:

1. the continuous variable \( s \) is fixed to a given value \( \bar{s} \);
2. a valid inequality \( \alpha^T y \leq \beta_\bar{s} \) for the resulting binary knapsack polytope \( \text{conv}(Y_{\bar{s}}) \) (seed inequality) is computed;
3. the continuous variable \( s \) is lifted to get a valid inequality for \( \text{conv}(Y) \).

Therefore, the separation procedure LCISCV-SEP consists of two steps: first a seed inequality for \( \text{conv}(Y_{\bar{s}}) \) is generated and then it is lifted to produce a valid inequality for \( \text{conv}(Y) \). We call our inequalities Lifted Cover Inequalities with a Single Continuous Variable, as we focus on Lifted Cover inequalities, but the method is general and can be used for any kind of seed inequalities. The choice of the Lifted Cover as seed inequalities, the computation of a seed inequality and the lifting of the continuous variable \( s \) are discussed below.

2.1. Finding a seed inequality

To find a seed inequality \( \alpha^T y \leq \beta_\bar{s} \) valid for the knapsack polytope \( \text{conv}(Y_{\bar{s}}) \), we restrict our attention to Lifted Cover Inequalities, as fast separation algorithms are available for them [19, 22]. The separation of Lifted Cover Inequalities consists of three steps:

i) Let \( I \) the index set of the binary variables and let \( L0 \subset I \) and \( L1 \subset I \) be the index subsets of the variables \( i \) such that \( \bar{x}_i = 0 \) and \( \bar{x}_i = 1 \), respectively. Set \( x_i = 0 \) for each \( i \in L0 \) and \( x_i = 1 \) for each \( i \in L1 \) to get a core problem defined by the fractional variables \( I \setminus (L0 \cup L1) \).
ii) Look for a maximally violated Cover Inequality \( x(C) \leq |C| - 1 \) over the core problem by solving to optimality by Dynamic Programming [29] the binary knapsack problem:

\[
\begin{align*}
\max & \quad (1 - \bar{x})^T w \\
\text{subject to} & \quad a^T w > b_g \\
& \quad w \in \{0, 1\}^n
\end{align*}
\]

where \( w_i = 1 \) iff \( i \in C \), 0 otherwise.

iii) Make the resulting inequality globally valid by sequential lifting of the variables of \( L_0 \) (uplifting) and \( L_1 \) (down-lifting). Computing a lifting coefficient amounts to solve a binary knapsack problem by Dynamic Programming (see [19, 22]).

The lifting procedure is sequential and the resulting valid inequality depends on the lifting order.

2.2. Lifting of the continuous variable

Let \( \alpha^T y \leq \beta_s \) be a generic valid inequality for the binary knapsack polytope \( \text{conv}(Y_s) \). The inequality \( \alpha^T y \leq \beta_s \) requires to be lifted to become valid for \( \text{conv}(Y) \). To this aim, it is possible to consider the following proposition.

**Proposition 2.1.** [24] Let \( \eta(s) \) be the optimal value of the problem below.

\[
\eta(s) = \max \alpha^T y \\
\text{subject to} & \quad a^T y \leq b + s \\
& \quad y \in \{0, 1\}^n
\]

The inequality \( \alpha^T y \leq \beta_s + \gamma s \) is valid for \( \text{conv}(Y) \) if \( \eta(s) \leq \beta_s + \gamma s \) for each \( s \in \mathbb{R}_+ \).

To compute a value of \( \gamma \) satisfying the condition given in Proposition 2.1, we adopt the following procedure, which consists of iteratively updating a lower estimator \( \gamma_{lb} \) of \( \gamma \) until the corresponding inequality \( \alpha^T y \leq \beta_s + \gamma_{lb} s \) becomes valid for \( \text{conv}(Y) \), then we set \( \gamma = \gamma_{lb} \) and the procedure terminates.

**Initialization**

Set \( \gamma_{lb} = \frac{\alpha^T - \beta_s}{\alpha^T - b - s} \).

**Step 1**

Solve the problem:

\[
\zeta = \max \alpha^T y - \gamma_{lb} s - \beta_s \\
\text{subject to} & \quad a^T y \leq b + s \\
& \quad y \in \{0, 1\}^n \\
& \quad s \geq 0
\]

Let \((y^*, s^*)\) be an optimal solution of problem (2) and let \( \zeta^* \) be its value.

**Step 2**

If \( \zeta^* \leq 0 \) then the inequality \( \alpha^T y \leq \beta_s + \gamma_{lb} s \) is valid for \( \text{conv}(Y) \), then we set \( \gamma = \gamma_{lb} \) and the procedure terminates. Otherwise go to Step 3.
Step 3

Increase the slope coefficient \( \gamma_{lb} \) by setting \( \gamma_{lb} = \frac{\alpha^T y^* - \beta_0}{\bar{s} - \bar{s}} \) and go back to Step 1.

The procedure can be illustrated by a graphical example. Let \( Y = \{(y,s) \in \{0,1\}^5 \times \mathbb{R}_+ : 7y_1 + 6y_2 + 5y_3 + 3y_4 + 2y_5 - s \leq 6\} \) be a mixed binary knapsack set with 5 binary variables \( y \) and a single continuous variable \( s \). Moreover suppose that \( \bar{s} \) is set to 1. The inequality \( y_1 + y_2 + y_3 + y_4 \leq 1 \) is valid for \( Y_{\bar{s}} \). Figure 1 reports the function \( \eta(s) \) for \( s \geq \bar{s} \).

![Figure 1: The function \( \eta(s) \)](image1)

Initially we set \( \gamma_{lb} = 3/14 \) and approximate the function \( \eta(s) \) by the line \( 1 + 3/14s \). This is not a valid approximation because the inequality \( y_1 + y_2 + y_3 + y_4 - 3/14s \leq 1 \) is not valid for \( \text{conv}(Y) \). An optimal solution of the problem:

\[
\zeta = \max \quad y_1 + y_2 + y_3 + y_4 - 3/14s - 1 \\
7y_1 + 6y_2 + 5y_3 + 3y_4 + 2y_5 \leq b + s \\
y \in \{0,1\}^5 \\
s \geq 0
\]

is \( y_3 = 1, \ y_4 = 1 \) and \( s = 2 \) with \( \zeta = 1.578 \). As indicated in Step 3, we set the new value of \( \gamma_{lb} \) to \( [(1 + 1) - 1]/(2 - 1) \). As we can see in Figure 2, the line \( 1 + s \) is a valid approximation for \( \eta(s) \) and the inequality \( y_1 + y_2 + y_3 + y_4 - s \leq 1 \) is valid for \( \text{conv}(Y) \).

![Figure 2: Two linear approximations of \( \eta(s) \)](image2)
3. Using LCISCV-SEP to solve MIP problems

In this section we describe the cutting plane algorithm based on the LCISCV-SEP procedure used in our computation experience to solve MIP problems. When embedded into a cutting plane algorithm, the separation procedure LCISCV-SEP is applied to every constraint of the problem formulation including only binary and continuous variables (equality constraints are split into two inequalities). Every constraint defines a mixed binary knapsack set $Y$ (base inequality) with positive and negative coefficients:

$$Y = \{(y_j, s_j) \in \{0, 1\}^{|I|} \times \mathbb{R}^{|P|} : \sum_{j \in I} a_j y_j + \sum_{j \in P} g_j s_j \leq b\}$$

which can be used to derive valid inequalities to strengthen the formulation. For each mixed binary set $Y^{MB}$ we consider a corresponding set $Y$ such that $Y^{MB} \subseteq Y$, obtained by performing the following operations:

1. a bound substitution procedure (see Section 3.1) is applied to replace some continuous variables by their respective simple or variable bounds;

2. all the continuous variables with non-negative coefficients can be discarded, as noted in [2], because all the non-dominated valid inequalities of $Y^{MB}$ have zero coefficients for these variables;

3. the continuous variables with negative coefficients are aggregated into a single continuous variable $s$:

$$s = -\sum_{j \in P^-} g'_j x'_j,$$

where $P^- = \{j \in P : g'_j < 0\}$.

4. the integer variables with negative coefficients are complemented, i.e. we set

$$y_j = \begin{cases} u_i - y'_i & i \in I^- \\ y'_i & i \in I \setminus I^- \end{cases}$$

where $I^- = \{j \in I : a'_j < 0\}$.

The resulting binary knapsack set with a single continuous variable is:

$$\sum_{i \in I} a''_i y'_i - s \leq b''.$$  

where

$$a''_i = \begin{cases} -a'_i & i \in I^- \\ a'_i & i \in I \setminus I^- \end{cases}$$

and $b'' = b' - \sum_{i \in I^-} a'_i u_i$.

5. the coefficients of the integer variables of the resulting mixed binary knapsack set are converted into integers (see Section 3.2) in order to use the dynamic programming algorithm in [22].

The implementation details of the algorithm are analyzed below.
3.1. Bound substitution

In our computational experience, we found that it is crucial to the success of the algorithm to reduce the weight of the continuous part of the knapsack constraint. Consider the mixed binary knapsack set $Y^{MB}$ and let $s_j$, $j \in P$, be a continuous variable such that both simple $l_j \leq s_j \leq v_j$ and variable bounds $\tilde{l}_j y_i \leq s_j \leq \tilde{v}_j y_k$ are defined and at least one of these bounds is finite. Bound substitution \cite{24, 32} consists of replacing $s_j$ by its respective simple/variable bound by performing one of the following substitutions using an auxiliary (slack) variable $s'_j$:

\[
\begin{align*}
  s_j &= l_j + s'_j \\
  s_j &= v_j - s'_j \\
  s_j &= \tilde{l}_j y_i + s'_j \\
  s_j &= \tilde{v}_j y_k - s'_j
\end{align*}
\]

Let $(\bar{y}, \bar{s})$ be the current fractional solution. To choose which bound is used to replace $s_j$ we adopt a simple heuristic rule which consists of selecting the bound with the smallest slack. We first compute:

\[
\mu = \min\{\bar{s}_j - l_j, v_j - \bar{s}_j, \bar{s}_j - \tilde{l}_j y_i, \tilde{v}_j y_k - \bar{s}_j\}.
\]

Then we set:

\[
s_j = \begin{cases} 
  l_j + s'_j & \text{if } \mu = \bar{s}_j - l_j \\
  v_j - s'_j & \text{if } \mu = v_j - \bar{s}_j \\
  \tilde{l}_j y_i + s'_j & \text{if } \mu = \tilde{s}_j - \tilde{l}_j y_i \\
  \tilde{v}_j y_k - s'_j & \text{if } \mu = \tilde{v}_j y_k - \bar{s}_j
\end{cases}
\]

Once the separation procedure has terminated successfully, all the auxiliary variables are substituted back to get a cutting plane in the original variables.

3.2. Converting the coefficients into integers

Converting all the coefficients into integers is necessary because the algorithm used to find lifted cover inequalities for the binary knapsack set \cite{22} uses dynamic programming. Moreover we also use dynamic programming to solve the knapsack problem in Step 1 of LCISP-SEP by algorithm in \cite{29} (see Section 3.3). To this aim we also convert the objective function of the problem, if not integer. The problem of converting a set of coefficients into integers can be formulated as a mixed integer programming problem as done in \cite{7} for the generalized assignment problem. But, in order to get faster computational times, a brute-force approach is used. Let $d$ be an entry to be converted into integer, we enumerate all the $q \in \mathbb{N}$ belonging to the interval $[1, 10^5]$, and we stop when $qb - \lfloor qb \rfloor \leq \epsilon$ and $qd - \lfloor d \rfloor \leq \epsilon$. In our experiments we set $\epsilon = 10^{-6}$. If the procedure fails, we discard the inequality since too large coefficients may cause numerical instability.

3.3. Efficient solution of Step 1 of LCISP-SEP

In order for LCISP-SEP to be computationally effective, the problem to be solved in Step 1 of LCISP-SEP, must be efficiently solved. We observe that any optimal solution $(y^*, s^*)$ of the lifting problem (2) has the property that $s^* = \max\{0, a^T y - b\}$. In addition, by the definition of $Y$, we have that $a^T y - b \geq 0$, therefore $s^* = a^T y - b$. It follows that the knapsack constraint in
problem (2) has to be satisfied with strict equality and thus the problem can be re-written as below.

\[
\zeta = \max \alpha^T y - \gamma \nu s - \beta s \\
a^T y = b + s \\
y \in \{0, 1\}^n \\
s \geq 0
\]

By projecting out the \(s\) variable, the lifting problem (5) reduces to the binary knapsack problem:

\[
\zeta = \max \sum_{j=1}^{n} (\alpha_j - \gamma \nu a_j) y_j - \beta s - \gamma \nu b \\
a^T y \geq b \\
y \in \{0, 1\}^n
\]

Problem (6) can be solved using the dynamic programming algorithm presented in [29].

4. Computational experiments

Computational experiments were carried out on a laptop with Centrino i7 processor and 8 Gb RAM using instances from the MIPLIB 2010. We compared the lower bound produced the root node by two algorithms: Algorithm SCIP and Algorithm LCISCV-SEP. In Algorithm SCIP, implemented using solver SCIP version 3.0.0 [34], the linear relaxation of the considered MIP problems is strengthened using Lifted Cover and MIR inequalities. All cut generation routines but Lifted Cover and MIR inequalities in SCIP were disabled and SCIP parameters were set to perform the MIR separation procedure on single rows, forbidding constraint aggregation [32]. Algorithm LCISCV-SEP, implemented using the FICO Xpress 7 callable library, starts from the formulation returned by Algorithm SCIP and tries to additionally improve the bound by adding LCISCV cuts. LCISCV cuts were generated starting from single rows of the original formulation using the algorithm proposed in Section 3. FICO Xpress was used only as LP solver, that is the cuts were turned off and hence the bound increasing obtained using Algorithm LCISCV-SEP, if any, is due only to LCISCV. All the other options for SCIP and FICO Xpress were set to their default values. To evaluate the quality of the lower bounds, upper bounds for the instances whose optimal solution was unknown were computed by running FICO Xpress 7.3 for 1800 secs with the option HEUSTRATEGY set to 3. The corresponding values are denoted by an asterisk.

4.1. The test bed

We first extracted from the MIPLIB 2010 the instances corresponding to mixed-integer and mixed-binary problems, obtaining 184 instances. Then we removed from the list the 23 instances where SCIP could not solve the LP-relaxation within the time limit of 1800 secs. Two more instances (neos-824661 and neos-824695) were removed since their gap was completely closed using only MIR inequalities, without need of further cut generation. LCISCV inequalities are derived from the mixed knapsack set, so it is reasonable to expect them to be effective on the problems where other, possibly simpler, cuts derived from the knapsack set - like MIR - already proved to be useful. we used the results of the MIR cuts as an “indicator” to recognize promising
instances where LCISCV-SEP could do well. So we partitioned the remaining instances into two sets, namely $A$ and $B$.

Set $A$ was our main test bed, including all the “promising” instances where MIR inequalities were effective in closing the gap (37 instances). For six of the selected instances the optimal solution was unknown (they are classified in the MIPLIB 2010 as open), for eight of them the optimal solution is known, but the problem was considered difficult to solve (they were classified as hard), while the remaining ones had a known optimal solution that could be computed within one hour (they are classified as easy). The smallest problem has 93 rows and 888 columns, the largest has 328818 rows and 164547 columns. Set $B$ included the instances where MIR inequalities turned out to be ineffective and hence knapsack-based cuts, as like LCISCV, were not expected to provide a significant contribution in tightening lower bounds. Nevertheless, for the sake of completeness we also tested our algorithm LCISCV-SEP on the instances of set $B$.

4.2. The results

In Tables 1 and 2 we report the results we obtained testing Algorithm MIR and LCISCV-SEP on the instances sets $A$ and $B$, respectively. For the set $B$ we only report the successful instances, i.e. those where LCISCV-SEP was able to significantly improve the lower bound. For both tables, LP and BUB are the values of the LP-relaxation and of the best known integer solution, respectively. For both approaches we report the lower bound obtained ($LB$), the corresponding closed gap ($\%Gap$) computed as $(LB - LP)/(UB - LP) \times 100$ and the computational time in seconds ($Time$). LCISCV-SEP runs on the formulation returned by SCIP after the addition of MIR inequalities, so the computational time for LCISCV-SEP represent the time increasing due to the separation of LCISCV. The lower bounds of the instances were Algorithm LCISCV-SEP performs better than Algorithm MIR are denoted in bold.

For set $A$, where MIR cuts were already very effective, LCISCV-SEP could further tighten the lower bounds for 30 of the 37 instances. The average improvement in the gap due to the use of LCISCV, computed as $\sum_{i \in A}(gap_{LCISCV-SEP}^{i} - gap_{SCIP}^{i})/|A|$, is 5.02%, with an average time increasing of 11.86 seconds. It is worth noting that two of the successful instances are currently classified as open, while six are classified as hard. For set $B$, knapsack-based cuts were not expected to be effective, as MIR inequalities were not able to improve the lower bounds returned by the LP relaxation on any of them. Nonetheless for five instances (one currently classified as open and one as hard) of the set $B$, LCVISCV-SEP produced tighter lower bounds.

5. Conclusions

In this paper we have studied from a computational standpoint an efficient separation algorithm for the Lifted Cover Inequalities with a Single Continuous Variable, a family of cutting planes that can be derived from mixed knapsack problems. We test the separation algorithm on a set of benchmark instances of the MIPLIB 2010, showing that LCISCV-SEP leads to significant improvements in the lower bound, with little additional computational effort, even on hard instances, while the time increase with respect to the separation of MIR inequalities is limited. In our opinion, this makes the separation algorithm a good candidate to be embedded into MIP solvers.
<table>
<thead>
<tr>
<th>Name</th>
<th>LP</th>
<th>SCP</th>
<th>LB % Gap</th>
<th>Time</th>
<th>LCSCV-SEP</th>
<th>LB % Gap</th>
<th>Time</th>
<th>BUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1c1s1</td>
<td>997.53</td>
<td>3190.72</td>
<td>20.87 1</td>
<td>5218.16</td>
<td>40.17 1</td>
<td>11503.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>afollow4b</td>
<td>1005.68</td>
<td>1022.02</td>
<td>10.66 2</td>
<td>1024.95</td>
<td>11.87 1</td>
<td>1168</td>
<td></td>
<td></td>
</tr>
<tr>
<td>appl-2</td>
<td>-263.71</td>
<td>-91.28</td>
<td>77.42 947</td>
<td>-91.28</td>
<td>77.42 37</td>
<td>-41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>beasleyC3</td>
<td>237.99</td>
<td>241.89</td>
<td>0.75 1</td>
<td>255.56</td>
<td>3.4 1</td>
<td>754</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bharar40.1</td>
<td>6637.12</td>
<td>6700.98</td>
<td>60.77 1</td>
<td>6700.98</td>
<td>60.77 1</td>
<td>6742.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>blp-ar88</td>
<td>6063.25</td>
<td>6067.53</td>
<td>3.02 14</td>
<td>6068.95</td>
<td>4.01 1</td>
<td>6205.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>blp-ic97</td>
<td>3877.83</td>
<td>3886.35</td>
<td>5.79 5</td>
<td>3886.92</td>
<td>6.17 6</td>
<td>4025.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b2c1s1</td>
<td>4034.21</td>
<td>8541.93</td>
<td>20.82 2</td>
<td>9349.77</td>
<td>24.55 1</td>
<td>25687.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>csched007</td>
<td>389089332</td>
<td>389070188</td>
<td>3.17 5</td>
<td>389089332</td>
<td>3.29 5</td>
<td>404228000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>csched010</td>
<td>350.81</td>
<td>24.33 1</td>
<td>11503.44</td>
<td>4025.02</td>
<td>6.17 6</td>
<td>408</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g200x740i</td>
<td>6063.25</td>
<td>6067.53</td>
<td>3.02 14</td>
<td>6068.95</td>
<td>4.01 1</td>
<td>6205.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g200x740j</td>
<td>6063.25</td>
<td>6067.53</td>
<td>3.02 14</td>
<td>6068.95</td>
<td>4.01 1</td>
<td>6205.21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>k16x240</td>
<td>2769.83</td>
<td>5139.53</td>
<td>29.98 1</td>
<td>6588.59</td>
<td>48.31 1</td>
<td>10674</td>
<td></td>
<td></td>
</tr>
<tr>
<td>leo1</td>
<td>389089332</td>
<td>389070188</td>
<td>3.17 5</td>
<td>389089332</td>
<td>3.29 5</td>
<td>404228000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>liu</td>
<td>346</td>
<td>443.33</td>
<td>10.69 1</td>
<td>496.04</td>
<td>16.48 1</td>
<td>1256.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>map14</td>
<td>-776.94</td>
<td>1.54 41</td>
<td>-776.87 1</td>
<td>6588.59</td>
<td>48.31 1</td>
<td>10674</td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxgasflow</td>
<td>-54892380</td>
<td>-54669024</td>
<td>2.16 2</td>
<td>-54015095</td>
<td>8.49 5</td>
<td>-4565819</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ncl1</td>
<td>608.84</td>
<td>1564.1</td>
<td>6.62 1</td>
<td>1817.67</td>
<td>10.91 2</td>
<td>11689</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nkc</td>
<td>-611.85</td>
<td>-593.01</td>
<td>39.26 1</td>
<td>-593.01</td>
<td>39.26 1</td>
<td>-563.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n3700</td>
<td>972305</td>
<td>1006875</td>
<td>29.98 1</td>
<td>1009701</td>
<td>35.13 3</td>
<td>1308889</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n370a</td>
<td>972919</td>
<td>1011515</td>
<td>35.13 3</td>
<td>1015646</td>
<td>35.13 3</td>
<td>1308889</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n3705</td>
<td>973381</td>
<td>1008222</td>
<td>35.13 3</td>
<td>1008222</td>
<td>35.13 3</td>
<td>1308889</td>
<td></td>
<td></td>
</tr>
<tr>
<td>raeos-911880</td>
<td>23.26</td>
<td>36.64 1</td>
<td>42.94 1</td>
<td>36.64</td>
<td>42.94 1</td>
<td>54.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set12</td>
<td>68.40</td>
<td>72.94 1</td>
<td>3.12 18</td>
<td>74.43</td>
<td>4.14 24</td>
<td>214*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pg</td>
<td>-11824.66</td>
<td>-9233.18</td>
<td>79.40 6</td>
<td>-9165.68</td>
<td>84.40 3</td>
<td>-8674.34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pg5.34</td>
<td>-16646.59</td>
<td>-14672.51</td>
<td>85.56 5</td>
<td>-14496.77</td>
<td>93.18 2</td>
<td>-14339.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p100x588b</td>
<td>555.01</td>
<td>26732.60</td>
<td>50.04 1</td>
<td>30333.55</td>
<td>58.54 1</td>
<td>47878</td>
<td></td>
<td></td>
</tr>
<tr>
<td>p80x400b</td>
<td>6418.80</td>
<td>20187.33</td>
<td>41.41 1</td>
<td>24362.63</td>
<td>53.97 1</td>
<td>39667</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ran14x18</td>
<td>3016.94</td>
<td>3208.15</td>
<td>27.51 1</td>
<td>3243.85</td>
<td>32.65 1</td>
<td>3712</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ran16x36</td>
<td>3116.43</td>
<td>3397.12</td>
<td>39.72 1</td>
<td>3463.63</td>
<td>49.14 1</td>
<td>3823</td>
<td></td>
<td></td>
</tr>
<tr>
<td>r80x800</td>
<td>3561.48</td>
<td>4284.98</td>
<td>37.70 1</td>
<td>4569.44</td>
<td>54.62 1</td>
<td>5332</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set3-15</td>
<td>8364.01</td>
<td>13894.22</td>
<td>4.74 1</td>
<td>27139.85</td>
<td>16.11 1</td>
<td>124886</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set3-20</td>
<td>10347.38</td>
<td>15030.89</td>
<td>2.72 2</td>
<td>38841.12</td>
<td>16.59 1</td>
<td>18211.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shipsched</td>
<td>0</td>
<td>40722.79</td>
<td>27.60 25</td>
<td>50265.50</td>
<td>34.06 8</td>
<td>147553</td>
<td></td>
<td></td>
</tr>
<tr>
<td>transportmoment</td>
<td>348923000000</td>
<td>50749603126</td>
<td>7.90 20</td>
<td>-5021347035</td>
<td>9.03 4</td>
<td>-306310000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>uc-case3</td>
<td>7181.42</td>
<td>7185.44</td>
<td>17.09 1</td>
<td>7186.18</td>
<td>20.24 1</td>
<td>7204.92</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unitcal_7</td>
<td>19450550</td>
<td>19472170</td>
<td>11.68 97</td>
<td>19494367</td>
<td>23.68 1</td>
<td>19635558</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zib54-ueue</td>
<td>40810200</td>
<td>44587260</td>
<td>6.04 2</td>
<td>4948278</td>
<td>13.87 1</td>
<td>10334015</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: results for set A

References


Table 2: results for successful instances of set $B$

<table>
<thead>
<tr>
<th>Name</th>
<th>LP</th>
<th>SCIP</th>
<th>LCISCV-SEP</th>
<th>BUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>atlanta-ip</td>
<td>81.24</td>
<td>81.26</td>
<td>83.77</td>
<td>90.01</td>
</tr>
<tr>
<td>bienst2</td>
<td>11.72</td>
<td>11.72</td>
<td>14.89</td>
<td>54.6</td>
</tr>
<tr>
<td>lotsize</td>
<td>348385.36</td>
<td>348385.36</td>
<td>519463.83</td>
<td>1480195</td>
</tr>
<tr>
<td>newdano</td>
<td>11.72</td>
<td>11.72</td>
<td>17.23</td>
<td>65.6</td>
</tr>
<tr>
<td>set3-10</td>
<td>788.89</td>
<td>788.89</td>
<td>54598.48</td>
<td>338745.37</td>
</tr>
</tbody>
</table>


