CORE DISCUSSION PAPER
2007/24

Coefficient strengthening: A tool for formulating mixed integer programs

Kent Andersen[1] and Yves Pochet[2]

March 2007

Abstract

Providing a good formulation is an important part of solving a mixed integer program. We suggest to measure the quality of a formulation by whether it is possible to strengthen the coefficients of the formulation. Sequentially strengthening coefficients can then be used as a tool for improving formulations. We believe this method could be useful for analyzing and producing tight formulations of problems that arise in practice. We illustrate the use of the approach on a problem in production scheduling. We also prove that coefficient strengthening leads to formulations with a desirable property: if no coefficient can be strengthened, then no constraint can be replaced by an inequality that dominates it. The effect of coefficient strengthening is tested on a number of problems in a computational experiment. The strengthened formulations are compared to reformulations obtained by the preprocessor of a commercial software package. For several test problems, the formulations obtained by coefficient strengthening are substantially stronger than the formulations obtained by the preprocessor. In particular, we use coefficient strengthening to solve two difficult problems to optimality that have only recently been solved.

Keywords: Mixed integer programming, Cutting plane, Coefficient strengthening

[1]Institute for Mathematical Sciences, University of Copenhagen, Denmark. email: kha@math.ku.dk.
[2]Center for Operations Research and Econometrics (CORE) and Louvain School of Management (IAG-LSM), Université Catholique de Louvain, Belgium. email: pochet@core.ucl.ac.be

This work was partly carried out within the framework of ADONET, a European network in Algorithmic Discrete Optimization, contract no. MRTN-CT-2003-504438.
This text presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

1 Introduction

Mixed integer linear programming (MILP) is used to analyze and provide solutions to many practical problems. Examples of such problems can be found in airline crew scheduling, combinatorial auctions, electricity generation, financial engineering, sports scheduling and telecommunication network design. One reason for the success of MILP in these areas is that it is possible to model important practical problems as MILP problems. This is not an easy task for many applications. Another important reason is that the MILP problems that are formulated can be solved with current MILP solvers within a reasonable amount of time.

Unfortunately, the performance of a MILP solver is very dependent on the quality of the formulation of the MILP problem. It is crucial that the LP relaxation gives a good approximation to the set of mixed integer solutions. For many classes of problems, there is literature on how to obtain such MILP formulations. If a formulation has many problem specific and/or non-standard constraints, however, this knowledge might not be useful. It therefore seems desirable to have a general tool available that can be used to produce tight formulations of MILP problems.

Most MILP solvers have a component called a preprocessor that attempts to repair a bad formulation (see Brearley, Mitra and Williams [5], Hoffman and Padberg [9] and Savelsbergh [13]). A typical preprocessing technique examines the formulation of a problem and checks whether a given structure is present. If so, a number of variables are removed, some coefficients are changed and/or some constraints are removed. A disadvantage of this heuristic approach is that the structure of certain problems might not be recognized.

In this paper, we present a general approach for creating tight formulations of MILP problems. We consider two possible measures for the quality of a formulation. The first measure considers whether it is possible to *replace* an inequality constraint of the formulation with a better inequality. By a "better" inequality, we mean a valid inequality that strictly dominates the constraint on the LP relaxation. We call the new inequality a *dominating* inequality, and the operation of replacing a constraint with a dominating inequality is called *constraint replacement*. This operation gives a new formulation with an LP relaxation that provides a better approximation to the set of mixed integer solutions. A formulation for which it is *not* possible to perform constraint replacement is then considered a good formulation. Observe that constraint replacement does *not* increase the number of constraints. A better approximation to the set of mixed integer solutions can also be obtained by adding cutting planes. However, this also increases the size of the formulation (see Nemhauser and Wolsey [11] for a general introduction to polyhedral theory and cutting plane algorithms).

The idea of constraint replacement can be viewed as a natural extension of an approach of Bradley, Hammer and Wolsey [4], where a single constraint of the LP relaxation is considered, and the integrality constraints are used to improve the coefficients in the constraint. This gives a new valid inequality that dominates the constraint. Constraint replacement extends this idea to *sets* of inequalities, exploiting that several different sets of inequalities (identical in number) can have the same set of mixed integer solutions, but very different linear relaxations.

An alternative measure of the quality of a formulation is to consider whether it is possible to strengthen a coefficient. We show (Theorem 1) that it is possible to obtain a formulation that is good with respect to constraint replacement by sequentially strengthening the coefficients of the formulation. In other words, a formulation that is good with respect to coefficient strengthening is also good with respect to constraint replacement. We present the optimization problem for strengthening a coefficient as much as possible, and we give an algorithm for solving this problem.

How good are formulations obtained from coefficient strengthening compared to reformulations obtained from a preprocessor? To answer this question, we apply coefficient strengthening to a number of test problems in MIPLIB, an electronically available library of MILP problems (Bixby et al. 1998). The formulations are compared to formulations obtained from the preprocessor of CPLEX version 9.1. For several MIPLIB instances, the formulations obtained by coefficient strengthening were substantially stronger than the formulations obtained from the preprocessor of CPLEX 9.1. For one instance the solution of the LP relaxation was integer after modifying coefficients. Using coefficient strengthening and branch-and-cut, we were able to solve to optimality two difficult instances in MIPLIB that have only recently been solved.

Coefficient strengthening can also be used to strengthen the coefficients in the cuts that are used in MILP solvers. A natural question is how strong these strengthened cuts can be compared to cuts that have not been strengthened. We investigate this question for mixed integer Gomory cuts, a class of general purpose cutting planes, in a computational experiment on the MIPLIB instances. On several test problems, the amount of integrality gap that is closed with strengthened cuts is more than twice as large as the amount of gap that is closed without strengthening the cuts. This suggests that it might be useful to include an option in MILP solvers that allows coefficient strengthening of cuts for difficult instances.

For a practical MILP problem, it seems likely that a reasoning can be found, which is specific to that particular application, that explains why a coefficient can be strengthened. This suggests that coefficient strengthening can be used as follows as a tool for constructing tight formulations of practical MILP problems. First, produce an initial small instance of the problem and strengthen the coefficients as much as possible. Then try to trace the logic of each strengthened coefficient. If an explanation can be found, a new model can be constructed that does not have the same weaknesses as the old model. The result is an understanding of how to build tight formulations for this problem class. We illustrate how to use this approach to construct a tight formulation for a problem in production scheduling.

Consider any mixed integer linear program (MILP)

$$\begin{aligned} \min \quad & c^T x & & \\ \text{s.t.} \quad & a_{i.}^T x = b_i, & \text{for } i \in M^{=}, & \quad (1) \\ & a_{i.}^T x \geq b_i, & \text{for } i \in M^{\geq} \text{ and} & \quad (2) \\ & x_j \text{ integer}, & \text{for } j \in N_I, & \quad (3) \end{aligned}$$

where $c, x \in \mathbb{R}^n$, $M^{\geq}$ and $M^{=}$ are index sets for the inequality and equality constraint respectively, $N_I \subseteq N := \{1, 2, \ldots, n\}$ and $(a_{i.}, b_i) \in \mathbb{R}^{n+1}$ for $i \in M := M^{=} \cup M^{\geq}$. The linear programming relaxation of MILP is obtained from MILP by dropping (3) and is denoted LP. The sets P_I and P denote the feasible solutions to MILP and LP respectively, and we assume $P_I \neq \emptyset$. By a *formulation* of MILP we mean the set P and the constraints that define P.

The following notation is used to describe polyhedra obtained by eliminating some of the constraints defining P. Let $S := S^{=} \cup S^{\geq}$ denote some subset of the constraints that define P. The set $P(S) := \{x \in \mathbb{R}^n : a_{i.}^T x = b_i, i \in S^{=}, a_{i.}^T x \geq b_i, i \in S^{\geq}\}$ denotes the polyhedron obtained from P by only considering the constraints in $S \subseteq M$.

Constraint replacement is defined from the following concept. Given a constraint $a_{k.}^T x \geq b_k$, the inequality $\alpha^T x \geq \beta$ *dominates* $a_{k.}^T x \geq b_k$ on P if $P' := P(M \setminus \{k\}) \cap \{x \in \mathbb{R}^n : \alpha^T x \geq \beta\} \subseteq P$. If also $P' \subsetneq P$, the inequality $\alpha^T x \geq \beta$ *strictly* dominates $a_{k.}^T x \geq b_k$ on P. In this paper, an inequality $\alpha^T x \geq \beta$ that strictly dominates an inequality constraint $a_{k.}^T x \geq b_k$ on P is called a *strictly dominating* inequality. If a strictly dominating inequality is valid for P_I, it can replace the constraint it dominates in the description of P and give a tighter approximation to P_I.

We are only interested in dominating inequalities $\alpha^T x \geq \beta$ that are valid for P_I, *i.e.*, no mixed integer solution should be cut off by $\alpha^T x \geq \beta$. In other words, $\alpha^T x \geq \beta$ must be valid for some relaxation R of P_I, where a relaxation of P_I is defined to be a superset of P_I. Throughout this paper, R denotes a relaxation of P_I. We assume R is a polyhedron, and that $P_I \subseteq R \subseteq P$.

The remainder of this paper is organized as follows. In Section 2 we derive theoretical properties of constraint replacement and coefficient strengthening. We show that a formulation for which no constraint can be replaced with a dominating inequality can be obtained by sequentially strengthening coefficients. We also present the optimization problem for strengthening a coefficient as much as possible, and we give an algorithm for solving this problem. Computational results are presented in Section 3. We test the quality of the formulations in MIPLIB, and we attempt to strengthen the coefficients in the mixed integer Gomory cuts. In Section 4 we introduce a problem in production scheduling, and we use coefficient strengthening to produce a tight formulation for this problem. The proofs of the statements in Section 2 are given in the appendix.

2 Theoretical foundation

In this section we discuss how to improve the formulation of MILP by either strengthening its coefficients, or by replacing a constraint with a dominating inequality. In Section 2.1 we show how to use a relaxation R of P_I to strengthen the coefficients of an inequality constraint. We characterize the set of dominating inequalities that can replace a constraint in the formulation of MILP in Section 2.2. The concepts of constraint replacement and coefficient strengthening are related in Section 2.3. In Section 2.4 we present a disjunctive program that formulates the problem of strengthening a left hand side coefficient. A branch and bound algorithm for solving this disjunctive program is given in Section 2.5. The proofs of the statements in this section are given in the appendix.

2.1 Coefficient strengthening relative to a polyhedral relaxation

We now discuss how to improve the inequality $a_{k.}^T x \geq b_k$ by modifying a coefficient. We require that the resulting inequality is valid for R. First consider strengthening the coefficient on a non-negative surplus variable. Let $l \in M^{\geq}$ be arbitrary, $l \neq k$, and let $s_l := a_{l.}^T x - b_l$ denote the surplus variable in the l^{th} constraint. The variable s_l is non-negative, and it currently appears in the constraint $a_{k.}^T x \geq b_k$ with a coefficient of zero. Suppose we can find $w_{k,l} > 0$ such that the inequality

$$a_{k.}^T x - w_{k,l}(a_{l.}^T x - b_l) \geq b_k \tag{4}$$

is valid for R. Inequality (4) is stronger than the inequality $a_{k.}^T x \geq b_k$. Any $x \in P$ that satisfies $a_{l.}^T x \geq b_l$ and (4) also satisfies $a_{k.}^T x \geq b_k$. Replacing $a_{k.}^T x \geq b_k$ with (4) in the description of P therefore leads to a polyhedron P' satisfying $P' \subseteq P$, *i.e.*, (4) dominates $a_{k.}^T x \geq b_k$ on P. Furthermore, any $x' \in P$ satisfying $a_{k.}^T x' = b_k$ and $a_{l.}^T x' > b_l$ violates (4). Therefore, if such a point $x' \in P$ exist, (4) *strictly* dominates $a_{k.}^T x \geq b_k$ on P. Since (4) is valid for R, (4) is also valid for P_I. The best value $w_{k,l}^*(R)$ of $w_{k,l}$ can be found by solving the following linear program ($\text{SP}_{k,l}(R)$)

$$w_{k,l}^*(R) := \max\{w_{k,l} : (4) \text{ is valid for R }\}. \tag{5}$$

The problem $\mathrm{SP}_{k,l}(R)$ is a linear program, because the cone of valid inequalities for R is a polyhedral cone. Also, since $a_{k.}^T x \geq b_k$ is valid for R, $\mathrm{SP}_{k,l}(R)$ is feasible. Finally, if $\mathrm{SP}_{k,l}(R)$ is unbounded, then $a_{l.}^T x = b_l$ for all $x \in R$. In other words, if $\mathrm{SP}_{k,l}(R)$ is unbounded, l can be moved from $M^{\geq}$ to $M^{=}$. We also consider this an improvement in the formulation LP of MILP. If the problem $\mathrm{SP}_{k,l}(R)$ is both feasible and bounded, and $w_{k,l}^*(R) > 0$, then the coefficient on s_l in the constraint $a_{k.}^T x \geq b_k$ can be strengthened.

Observe that, if $a_{l.}^T x \geq b_l$ is of the form $x_j \geq 0$ where $j \in N$, then solving the problem $\mathrm{SP}_{k,l}(R)$ corresponds to strengthening the coefficient on x_j in the constraint $a_{k.}^T x \geq b_k$. The above procedure can therefore be used to strengthen the coefficient on *any* non-negative variable.

It is also possible to use R to strengthen the right hand side of the inequality $a_{k.}^T x \geq b_k$. This can be done by solving the following linear program ($\mathrm{SP}_k(R)$)

$$b_k^*(R) := \max\{\beta : a_{k.}^T x \geq \beta \text{ is valid for R }\}. \tag{6}$$

The problem $\mathrm{SP}_k(R)$ is feasible since $a_{k.}^T x \geq b_k$ is valid for R. Furthermore, the problem $\mathrm{SP}_k(R)$ is unbounded if and only if $R = \emptyset$. Also, if $R = \emptyset$, then $P_I = \emptyset$. If $\mathrm{SP}_k(R)$ is feasible and bounded, and $b_k^*(R) > b_k$, the right hand side of $a_{k.}^T x \geq b_k$ can be strengthened from b_k to $b_k^*(R)$.

In the remainder of the paper, by sequential strengthening, we mean strengthening the formulation LP of MILP by solving a sequence of problems $\mathrm{SP}_{k,l}(R)$ and $\mathrm{SP}_k(R)$ for various combinations of rows $k, l \in M^{\geq}$ and relaxations R. Once a strengthened coefficient is found, or it is realized that a surplus variable can be fixed to zero, we say the formulation LP of MILP can be improved.

In Section 2.3 our main interest is in formulations LP of MILP that can *not* be improved by using R and sequential strengthening. We say that such formulations are optimal relative to R and sequential strengthening.

Definition 1 *Let R be a polyhedral relaxation of P_I satisfying $P_I \subseteq R \subseteq P$. The formulation LP of MILP is optimal relative to R and sequential strengthening, iff*

(i) For every $k \in M^{\geq}$, we have $b_k^*(R) = b_k$ (no right hand side can be increased).

(ii) For every $k, l \in M^{\geq}$, $k \neq l$, we have $w_{k,l}^*(R) = 0$ (no left hand side coefficient can be reduced).

In Section 2.3 we show that formulations which are optimal relative to R and sequential strengthening are also optimal relative to R and constraint replacement. We next consider dominating inequalities.

2.2 Constraint replacement relative to a polyhedral relaxation

We now characterize the set of dominating inequalities. A dominating inequality can replace the constraint it dominates in the formulation of LP and give a tighter LP relaxation. As in the previous section, R denotes a polyhedral relaxation of P_I satisfying $P_I \subseteq R \subseteq P$. We say the formulation LP of MILP is optimal relative to R and constraint replacement, if there is no strictly dominating inequality that is valid for R.

Definition 2 *Assume $P_I \subseteq R \subseteq P$ and R is a polyhedron. The formulation LP of MILP is optimal relative to R and constraint replacement iff there is no strictly dominating inequality that is valid for R*

Below we present a polyhedral cone that characterizes the set of valid inequalities for R that dominate a given inequality constraint $a_{k.}^T x \geq b_k$ of LP, where $k \in M^{\geq}$. We call this cone the *reformulation cone*. An inequality that belongs to the reformulation cone has to be both valid for R and dominate the inequality $a_{k.}^T x \geq b_k$. The set of valid inequalities for R is denoted $C^V(R)$. The set of inequalities $\alpha^T x \geq \beta$ that dominate the constraint $a_{k.}^T x \geq b_k$ on P are given by the following description. To give the description, we need auxiliary variables w_0 and $\{w_i\}_{i \in M \setminus \{k\}}$. The inequalities that dominate $a_{k.}^T x \geq b_k$ on P are described by the following system

$$\alpha = w_0 a_{k.} - \sum_{i \in M \setminus \{k\}} w_i a_{i.} \tag{7}$$

$$\beta \geq w_0 b_k - \sum_{i \in M \setminus \{k\}} w_i b_i \tag{8}$$

$$w_0 \geq 0 \tag{9}$$

$$w_i \geq 0, \qquad i \in M^{\geq} \setminus \{k\}. \tag{10}$$

Lemma 1 below gives the properties of the set of $(\alpha, \beta, w_0, \{w_i\}_{i \in M \setminus \{k\}})$ that satisfy (7)-(10). We assume $a_{k.}^T x \geq b_k$ is *not* redundant for P, *i.e.*, we assume $P \neq P(M \setminus \{k\})$. Observe that, if $a_{k.}^T x \geq b_k$ is redundant for P, then *every* valid inequality $\alpha^T x \geq \beta$ for R dominates $a_{k.}^T x \geq b_k$ on P. Also observe that, if $\alpha^T x \geq \beta$ cuts off $P(M \setminus \{k\})$, *i.e.*, if $P(M \setminus \{k\}) \cap \{x \in \mathbb{R}^n : \alpha^T x \geq \beta\} = \emptyset$, then $\alpha^T x \geq \beta$ trivially dominates $a_{k.}^T x \geq b_k$ on P.

Lemma 1 *Assume $a_{k.}^T x \geq b_k$ is not redundant for P, and that the inequality $\alpha^T x \geq \beta$ does not cut off every point in $P(M \setminus \{k\})$.*

(i) $\alpha^T x \geq \beta$ dominates $a_{k.}^T x \geq b_k$ on P $\iff$ there is a solution to (7)-(10) in which $w_0 > 0$.

(ii) $\alpha^T x \leq \beta$ is valid for $P(M \setminus \{k\})$ $\iff$ there is a solution to (7)-(10) in which $w_0 = 0$.

The system (7)-(10) describes the set of inequalities $\alpha^T x \geq \beta$ that dominate the constraint $a_{k.}^T x \geq b_k$ on P. Hence, if we add the condition $(\alpha, \beta) \in C^V(R)$ to conditions (7)-(10), we obtain a description of the set of valid inequalities $\alpha^T x \geq \beta$ for R that dominate $a_{k.}^T x \geq b_k$ on P.

$$\begin{aligned} &(7) - (10), \\ &(\alpha, \beta) \in C^V(R). \end{aligned} \tag{11}$$

The *reformulation cone* $\mathrm{RC}_k(R)$ for a constraint $a_{k.}^T x \geq b_k$ and a polyhedral relaxation R is defined to be the set of $(\alpha, \beta, w_0, \{w_i\}_{i \in M \setminus \{k\}}) \in \mathbb{R}^{n+1+|M|}$ that satisfy the system (7)-(11). The reformulation cone $\mathrm{RC}_k(R)$ lives in a space of dimension $(n+1+|M|)$. Furthermore, the inequality $\alpha^T x \geq \beta$ is valid for R for any $(\alpha, \beta, w_0, \{w_i\}_{i \in M \setminus \{k\}}) \in \mathrm{RC}_k(R)$, and the reformulation cone contains all valid inequalities for R that strictly dominate $a_{k.}^T x \geq b_k$ on P.

The reformulation cone also contains inequalities that are equivalent to $a_{k.}^T x \geq b_k$ on P. These inequalities can replace $a_{k.}^T x \geq b_k$ in the formulation of MILP, but this will not produce a tighter formulation. We now describe this set of inequalities. Firstly, since $a_{k.}^T x \geq b_k$ is valid for R (because $R \subseteq P$), and the values $\alpha = a_{k.}$, $\beta = b_k$, $w_0 = 1$ and $w_i = 0$ for $i \in M \setminus \{k\}$ satisfy (7)-(10), $\mathrm{RC}_k(R)$ contains the ray $\{(w_0 a_{k.}, w_0 b_k, w_0, 0_{|M|-1}) : w_0 \geq 0\}$. Secondly, inequalities $\alpha^T x \geq \beta$ obtained from

$(w_0 a_{k.})^T x \geq w_0 b_k$ by adding the equalities $-a_{i.}^T x = -b_i$, $i \in M^=$ with scalars $\{w_i\}_{i \in M^=}$ are clearly valid for R. It follows that $\text{RC}_k(R)$ contains the set of $(\alpha, \beta, w_0, \{w_i\}_{i \in M \setminus \{k\}})$ that satisfy

$$\alpha = w_0 a_{k.} - \sum_{i \in M^=} w_i a_{i.}\,, \tag{12}$$

$$\beta = w_0 b_k - \sum_{i \in M^=} w_i b_i\,, \tag{13}$$

$$w_i = 0, \qquad\qquad i \in M^{\geq} \setminus \{k\}\,, \tag{14}$$

$$w_0 \geq 0. \tag{15}$$

We call the set of $(\alpha, \beta, w_0, \{w_i\}_{i \in M \setminus \{k\}})$ that satisfy (11)-(15) the *trivial* reformulation cone. The trivial reformulation cone is denoted $\text{TRC}_k(R)$. For every $(\alpha, \beta, w_0, \{w_i\}_{i \in M \setminus \{k\}}) \in \text{TRC}_k(R)$, replacing $a_{k.}^T x \geq b_k$ with the inequality $\alpha^T x \geq \beta$ does *not* give a tighter formulation of MILP. In the next section, we characterize the case when $\text{TRC}_k(R) = \text{RC}_k(R)$, and this case is related to formulations LP of MILP that are optimal relative to R and sequential strengthening.

2.3 A relationship between constraint replacement and coefficient strengthening

The purpose of this section is the following relationship between constraint replacement and coefficient strengthening.

Theorem 1 *Assuming $P_I \neq \emptyset$, the formulation LP of MILP is optimal relative to R and sequential strengthening $\iff$ for every constraint $a_{k.}^T x \geq b_k$, $k \in M^{\geq}$, we have $RC_k(R) = TRC_k(R)$.*

Theorem 1 shows that, unless the reformulation cone *only* contains inequalities that are equivalent to $a_{k.}^T x \geq b_k$ on P, then there is a coefficient in the constraint $a_{k.}^T x \geq b_k$ that can be strengthened. We note that Theorem 1 implies that for a formulation, which is optimal relative to R and sequential strengthening, there are no valid inequalities for R that strictly dominate a constraint of LP.

Corollary 1 *Assume the formulation LP of MILP is optimal relative to R and sequential strengthening. Then no valid inequality for R strictly dominates an inequality constraint of LP.*

The proof of Theorem 1 is obtained by charactering the rays of the cone $\text{RC}_k(R)$. For an arbitrary ray $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i \in M \setminus \{k\}})$ of $\text{RC}_k(R)$, the inequality $(\alpha^r)^T x \geq \beta^r$ is valid for R by definition. Also, if $w_0^r = 0$, Lemma 1.(ii) shows that $(\alpha^r)^T x \leq \beta^r$ is valid for $P(M \setminus \{k\})$. It follows that the equality $(\alpha^r)^T x = \beta^r$ holds for all $x \in R$ when $w_0^r = 0$. The following lemma characterizes the case when $w_0^r = 0$.

Lemma 2 *Let $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i \in M \setminus \{k\}}) \in RC_k(R)$ satisfy $w_0^r = 0$. We have either*

(i) $R = \emptyset$ (this implies $P_I = \emptyset$), or

(ii) There exists $i \in M^{\geq}$ such that every $x \in R$ satisfies $a_{i.}^T x = b_i$ (the i^{th} surplus variable can be fixed to zero), or

(iii) $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i\in M\setminus\{k\}}) \in TRC_k(R)$
(the inequality $(\alpha^r)^T x \geq \beta^r$ *is equivalent to* $a_{k.}^T x \geq b_k$*).*

Lemma 2 shows that, under the assumption $P_I \neq \emptyset$, if there is a ray $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i\in M\setminus\{k\}})$ of $\mathrm{RC}_k(R)$ that satisfies $w_0^r = 0$, then either the formulation LP of MILP can be improved, or $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i\in M\setminus\{k\}}) \in \mathrm{TRC}_k(R)$. The following lemma considers the case when $w_0^r > 0$. Wlog we can assume $w_0^r = 1$.

Lemma 3 *Let* $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i\in M\setminus\{k\}}) \in RC_k(R)$ *satisfy* $w_0^r = 1$*. We have either*

(i) There exists $l \in M^{\geq} \setminus \{k\}$ *such that* $w_l^r > 0$ *and* $a_{k.}^T x - w_l^r(a_{l.}^T x - b_l) \geq b_k$ *is valid for* R
(the coefficient on s_l *in the* k^{th} *constraint can be improved), or*

(ii) There exists $b'_k > b_k$ *such that* $a_{k.}^T x \geq b'_k$ *is valid for* R
(The right hand side in the k^{th} *constraint can be improved), or*

(iii) $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i\in M\setminus\{k\}}) \in TRC_k(R)$
(the inequality $\alpha^T x \geq \beta$ *is equivalent to* $a_{k.}^T x \geq b_k$*).*

Lemma 2 and Lemma 3 show one direction of Theorem 1, *i.e.*, if there is a ray of $\mathrm{RC}_k(R)$ which is not in $\mathrm{TRC}_k(R)$, then it is possible to improve the formulation LP of MILP with R and sequential strengthening. Lemma 6 in the appendix proves the other direction.

2.4 The optimization problems for strengthening a coefficient relative to the mixed integer hull

We now present the mathematical problems that need to be solved to strengthen a coefficient in the important special case when $R = Conv(P_I)$. The right hand side of an inequality $a_{k.}^T x \geq b_k$, $k \in M^{\geq}$, can be strengthened relative to $Conv(P_I)$ by solving the mixed integer program

$$\min\{a_{k.}^T x : x \in P_I\}.$$

We next discuss how to strengthen the coefficient on a surplus variable $s_l := a_{l.}^T x - b_l \geq 0$ in the constraint $a_{k.}^T x \geq b_k$, where $k, l \in M^{\geq}$ and $k \neq l$. We will demonstrate that the following disjunctive program ($\mathrm{DP}_{k,l}$) solves the problem of whether or not the coefficient on s_l in the constraint $a_{k.}^T x \geq b_k$ can be strengthened.

$$\begin{aligned}
w_{k,l}^{dp} = \inf \quad & a_{k.}^T y - \lambda b_k \\
\text{s.t.} \quad & a_{i.}^T y \geq \lambda b_i, \quad \forall i \in M^{\geq}, && (16)\\
& a_{i.}^T y = \lambda b_i, \quad \forall i \in M^{=}, && (17)\\
& a_{l.}^T y - \lambda b_l = 1, && (18)\\
& \lambda \geq 0, && (19)\\
& y_j \leq d\lambda \quad \vee \quad y_j \geq (d+1)\lambda, \quad \forall (j,d) \in N_I \times \mathbb{Z}. && (20)
\end{aligned}$$

The following proposition shows that, if $\mathrm{DP}_{k,l}$ is infeasible, then the l^{th} constraint is satisfied with equality by all mixed integer solutions. In that case, the formulation LP of MILP can be improved by moving l from $M^{\geq}$ to $M^{=}$.

Proposition 1 *Assume $P_I \neq \emptyset$. The problem $DP_{k,l}$ is feasible $\iff$ there exists $x \in P_I$ such that $a_{l.}^T x > b_l$.*

Now assume $\mathrm{DP}_{k,l}$ is feasible. Since $a_{k.}^T y - \lambda b_k \geq 0$ is part of the formulation of $\mathrm{DP}_{k,l}$, the problem $\mathrm{DP}_{k,l}$ is bounded. Proposition 2 below shows $w_{k,l}^{dp} = w_{k,l}^*(Conv(P_I))$, and that $w_{k,l}^{dp}$ is attained by a feasible solution to $\mathrm{DP}_{k,l}$. It follows that the infimum in the definition of $\mathrm{DP}_{k,l}$ can be replaced by a minimum.

Proposition 2 *Assume $DP_{k,l}$ is feasible and the right hand side of $a_{k.}^T x \geq b_k$ can not be increased.*

(i) $w_{k,l}^{dp} = w_{k,l}^*(Conv(P_I))$.

(ii) The optimal objective value $w_{k,l}^{dp}$ to $DP_{k,l}$ is attained by a feasible solution to $DP_{k,l}$.

It follows that $\mathrm{DP}_{k,l}$ is indeed the optimization problem for finding the optimal coefficient on s_l in the constraint $a_{k.}^T x \geq b_k$. However, we note that the assumption of Proposition 2 that the right hand side of $a_{k.}^T x \geq b_k$ can not be strengthened is necessary for this to be true. This suggests that right hand sides should be strengthened before left hand side coefficients are strengthened. We give an algorithm for solving $\mathrm{DP}_{k,l}$ in the next section.

2.5 A branch and bound algorithm for solving $\mathrm{DP}_{k,l}$

A branch and bound algorithm for strengthening a left hand side coefficient is now presented, *i.e.*, an algorithm for solving $\mathrm{DP}_{k,l}$. This branch-and-bound method is called $\mathrm{B\&B}^{dp}$ in the following for simplicity. We assume the right hand side of the constraint $a_{k.}^T x \geq b_k$ can not be strengthened.

Let $\mathrm{LP}_{k,l}$ denote the LP relaxation of the disjunctive program $\mathrm{DP}_{k,l}$, *i.e.*, the linear program obtained from $\mathrm{DP}_{k,l}$ by eliminating the constraints (20). $\mathrm{B\&B}^{dp}$ starts by solving $\mathrm{LP}_{k,l}$. Let (y^{lp}, λ^{lp}) denote an optimal solution. If (y^{lp}, λ^{lp}) satisfies constraints (20), that is if y_j^{lp}/λ^{lp} is integer for all $j \in N_I$, $\mathrm{DP}_{k,l}$ is solved.

If (y^{lp}, λ^{lp}) does not satisfy (20), there is a disjunction $y_{j'} \leq d'\lambda \vee y_{j'} \geq (d'+1)\lambda$ which is violated by (y^{lp}, λ^{lp}) for some $j' \in N_I$ and $d' = \lfloor y_{j'}^{lp}/\lambda^{lp} \rfloor$. $\mathrm{B\&B}^{dp}$ now creates two subproblems. One subproblem is obtained from $\mathrm{DP}_{k,l}$ by adding the constraint $y_{j'} \leq d'\lambda$ to the formulation, and the other is obtained by adding the constraint $y_{j'} \geq (d'+1)\lambda$. The fact that $\mathrm{B\&B}^{dp}$ branches on disjunctions with two variables is the main difference between $\mathrm{B\&B}^{dp}$ and the usual branch-and-bound method.

The two subproblems are now solved recursively by $\mathrm{B\&B}^{dp}$. This process creates a branch-and-bound tree, where each node of the tree has a disjunctive program associated to it.

We now consider the processing of an arbitrary node v. Define the following sets.

$$S^{\leq} := \{(j,d) \in N_I \times \mathbb{Z} : y_j \leq d\lambda \text{ is enforced at } v\},$$
$$S^{\geq} := \{(j,d) \in N_I \times \mathbb{Z} : y_j \geq (d+1)\lambda \text{ is enforced at } v\},$$

and let S:$= S^{\leq} \cup S^{\geq}$. The disjunctive program $\mathrm{DP}_{k,l}(S)$ that must be solved at node v is given by

$$\begin{aligned} \min \quad & a_{k.}^T y - \lambda b_k \\ \text{s.t.} \quad & (16) - (20), \\ & y_j \leq d\lambda, \qquad \forall (j,d) \in S^{\leq}, \\ & y_j \geq (d+1)\lambda, \qquad \forall (j,d) \in S^{\geq}. \end{aligned}$$

Let $\mathrm{LP}_{k,l}(\mathrm{S})$ denote the linear program obtained from $\mathrm{DP}_{k,l}(\mathrm{S})$ by deleting the disjunctive constraints (20). To process node v, B&B^{dp} first solves $\mathrm{LP}_{k,l}(\mathrm{S})$. If $\mathrm{LP}_{k,l}(\mathrm{S})$ is feasible, let $(y^{lp}(\mathrm{S}), \lambda^{lp}(\mathrm{S}))$ denote an optimal solution, and let $w^{lp}_{k,l}(\mathrm{S})$ denote the optimal objective value. Also, let $w^f_{k,l}$ denote the objective value of the best known solution to $\mathrm{DP}_{k,l}$, where initially $w^f_{k,l} = \infty$. The following four cases can occur.

(1) The problem $\mathrm{LP}_{k,l}(\mathrm{S})$ is infeasible: this implies that $\mathrm{DP}_{k,l}(\mathrm{S})$ is also infeasible. B&B^{dp} then closes node v, *i.e.*, node v is deleted from the list of open nodes.

(2) $\mathrm{LP}_{k,l}(\mathrm{S})$ is feasible, and $w^{lp}_{k,l}(\mathrm{S}) \geq w^f_{k,l}$: this means that no feasible solution to $\mathrm{DP}_{k,l}(\mathrm{S})$ can have a better objective value than the solution with objective value $w^f_{k,l}$. In this case B&B^{dp} closes node v, and node v is said to be fathomed by bound.

(3) $\mathrm{LP}_{k,l}(\mathrm{S})$ is feasible, and $(y^{lp}(\mathrm{S}), \lambda^{lp}(\mathrm{S}))$ satisfies the disjunctive constraints (20): this means $(y^{lp}(\mathrm{S}), \lambda^{lp}(\mathrm{S}))$ is feasible for both $\mathrm{DP}_{k,l}(\mathrm{S})$ and $\mathrm{DP}_{k,l}$. If $w^{lp}_{k,l}(S) < w^f_{k,l}$, the solution $(y^{lp}(\mathrm{S}), \lambda^{lp}(\mathrm{S}))$ is better than the best known solution to $\mathrm{DP}_{k,l}$. B&B^{dp} therefore updates $w^f_{k,l}$ to $w^{lp}_{k,l}(S)$. Since $\mathrm{DP}_{k,l}(\mathrm{S})$ is solved, B&B^{dp} closes node v.

(4) $\mathrm{LP}_{k,l}(\mathrm{S})$ is feasible, and $(y^{lp}(\mathrm{S}), \lambda^{lp}(\mathrm{S}))$ does not satisfy the disjunctive constraints (20): in this case one can choose $(j', d') \in N_I \times \mathbb{Z}$ such that $(y^{lp}(\mathrm{S}), \lambda^{lp}(\mathrm{S}))$ violates the disjunction $y_{j'} \leq \lambda d' \vee y_{j'} \geq \lambda(d'+1)$. B&B^{dp} then creates two nodes $v^{\leq}$ and $v^{\geq}$ that are added to the list of open nodes. The constraints from (20) enforced on $v^{\leq}$ are given by $(S^{\leq} \cup \{(j', d')\}, S^{\geq})$, and the constraints enforced on $v^{\geq}$ are given by $(S^{\leq}, S^{\geq} \cup \{(j', d')\})$.

This finishes the processing of node v. Next B&B^{dp} selects another open node for processing (if an open node is available). This is continued until no more open nodes are available.

The following observations can be used to reduce the computational effort when a number of strengthening problems are solved in sequence. Consider two problems $\mathrm{DP}_{k,l}$ and $\mathrm{DP}_{q,l}$, where $k, l, q \in M^{\geq}$, $k \neq l$, $k \neq q$ and $l \neq q$. Observe that any feasible solution (y^f, λ^f) to $\mathrm{DP}_{k,l}$ is also a feasible solution to $\mathrm{DP}_{q,k}$. In particular, if (y^f, λ^f) is feasible for $\mathrm{DP}_{k,l}$, and $a^T_{q.} y^f = \lambda^f b_q$, then (y^f, λ^f) is an *optimal* solution to $\mathrm{DP}_{q,k}$. In other words, solving the problem $\mathrm{DP}_{k,l}$ also solves other strengthening problems.

Finally note that any lower bound on $w^{dp}_{k,l}$ provides a valid coefficient on s_l in the constraint $a^T_{k.} x \geq b_k$. Hence, as a heuristic, B&B^{dp} can be terminated after enumerating a number of nodes. Also, constraints of $\mathrm{DP}_{k,l}$ can be removed as a heuristic before starting the algorithm, because this ensures a valid lower bound on the value of the original disjunctive program.

3 Effect of Coefficient Strengthening

We now apply coefficient strengthening to a number of MILP instances from the MIPLIB library. The questions we attempt to answer are the following.

(a) Are the MIPLIB instances well formulated, and to what extend does a state-of-the-art pre-processor repair a bad formulation?

(b) Can the coefficients in mixed integer Gomory cuts be strengthened? If so, how significant can the difference be between strengthened cuts and cuts that have not been strengthened?

(c) Can coefficient strengthening be used to solve difficult mixed integer programs?

We use instances from the MIPLIB 3.0 and MIPLIB 2003 libraries in our experiments. MIPLIB 2003 is an updated version of MIPLIB 3.0. To simplify the discussion, we refer to the instances in both libraries as the MIPLIB instances. We use the software of CPLEX for our experiments (version 9.1). Many formulations in MIPLIB contain a large number of coefficients that can be strengthened with the preprocessor of CPLEX 9.1. In order to make a fair comparison with the preprocessor of CPLEX 9.1, and to measure the additional effect of coefficient strengthening, the preprocessor of CPLEX 9.1 is applied to the MIPLIB instances before applying coefficient strengthening. Hence, in the following, when we refer to an instance of MIPLIB, we mean the formulation obtained after applying the preprocessor of CPLEX 9.1.

The remainder of this section is organized as follows. In Section 3.1 we present the algorithm we have implemented for coefficient strengthening. The quality of the MIPLIB formulations is tested in Section 3.2. We attempt to strengthen the coefficients in mixed integer Gomory cuts in Section 3.3. Finally, coefficient strengthening is used to solve two difficult problems in MIPLIB in Section 3.4.

3.1 An algorithm for coefficient strengthening

The algorithm we use for strengthening coefficients is now presented. This algorithm is used in later sections for computational experiments.

We only attempt to strengthen coefficients on binary variables. The reason is that coefficient strengthening is easier for binary variables than for other variables. For a binary variable x_j and a constraint $a_{k.}^T x \geq b_k$, the inequality $a_{k.}^T x + \delta x_j \geq b_k$ is valid for P_I if and only if it is valid for $P_I \cap \{x : x_j = 1\}$. In other words, only one side of the integer disjunction on x_j needs to be considered, and the strengthening problem reduces to a mixed integer program. Furthermore, for most problems in MIPLIB, binary variables are the only integer variables that appear in the formulation.

We do not try to strengthen coefficients in constraints that express bounds on the variables, *i.e.*, constraints of the form $x_j \leq u_j$, or of the form $x_j \geq l_j$, where $l_j < u_j$. Bounds are not treated as other constraints by an LP solver. Changing a coefficient in such a constraint would therefore be similar to adding a constraint.

To improve the coefficient on a binary variable x_j in the constraint $a_{k.}^T x \geq b_k$, where $k \in M^{\geq}$ and $j \in N_I$, the following problem $\text{MIP}_{j,k}$ is given to CPLEX 9.1 and solved.

$$\begin{aligned} \min \quad & a_{k.}^T x - b_k & & \\ \text{s.t.} \quad & a_{i.}^T x = b_i, & \forall i \in M^{=}, & \quad (21) \\ & a_{i.}^T x \geq b_i, & \forall i \in M^{\geq}, & \quad (22) \\ & x_j = 1, & & \quad (23) \\ & x_s \text{ integer}, & \forall s \in N_I \setminus \{j\}. & \quad (24) \end{aligned}$$

Let x^* be an optimal solution to $\text{MIP}_{j,k}$. The strengthened coefficient on x_j is $\delta_{j,k}^* := b_k - a_{k.}^T x^*$. If $\delta_{j,k}^* < 0$, the inequality $a_k^T x + \delta_{j,k}^* x_j \geq b_k$ is stronger than $a_k^T x \geq b_k$. If $\delta_{j,k}^* = 0$, the coefficient on x_j in the constraint $a_k^T x \geq b_k$ can not be strengthened. Observe that, if $x^*_{j'} = 1$ and $a_{k'}^T x^* = b_{k'}$ for some binary variable $j' \in N_I \setminus \{j\}$ and constraint $k' \in M^{\geq} \setminus \{k\}$, then x^* is also an optimal solution to $\text{MIP}_{j',k'}$, and x^* certifies that the coefficient on $x_{j'}$ can not be strengthened in the constraint $a_{k'}^T x \geq b_{k'}$. After solving the problem $\text{MIP}_{j,k}$, we therefore identify all pairs (j', k') of binary variables and constraints that satisfy $x^*_{j'} = 1$ and $a_{k'}^T x^* = b_{k'}$, and the problem $\text{MIP}_{j',k'}$ will not be solved.

We measure the quality of a formulation by the amount of integrality gap that remains. We therefore only try to strengthen coefficients on binary variables that have a positive value in the current LP solution. Strengthening coefficients on variables with a value of zero in the current solution *does* strengthen the formulation. However, it does not change the integrality gap.

The variables are considered sequentially, and *all* strengthening problems are solved for a given variable before considering the next variable. The binary variable that has the largest positive value in the current LP solution is chosen as the next variable (ties broken arbitrarily). We stop the strengthening algorithm when all strengthening problems have been considered once, because all coefficients have then been strengthened as much as possible.

3.2 Strengthening the coefficients of the MIPLIB instances

We now apply coefficient strengthening to the formulations in MIPLIB. For all test problems, we attempt to create strengthened formulations with the following property.

(*) The LP solution to the formulation is an LP solution to a formulation for which no coefficient on *any* binary variable can be strengthened.

We do *not* create formulations for which no coefficient on a binary variable can be strengthened. We only produce the LP solution of such a formulation. There might be coefficients on binary variables with a value of zero in the LP solution that can be strengthened. However, strengthening these coefficients does not change the LP solution, and does therefore not change the amount of integrality gap that is closed.

Table 1 and Table 2 below contain our results for those instances for which we were able to create formulations that satisfy (*). Table 1 contains those instances for which *no* coefficient could be strengthened, and Table 2 contains the instances where *some* coefficients could be strengthened. The second and third columns contain the size of the problems after applying the preprocessor of CPLEX 9.1. The columns headed "Preprocessed LP value" contain the objective values of the LP relaxations. The column headed "Strengthened LP value" in Table 2 contains the objective values after applying the strengthening algorithm. The columns headed "Value of MILP optimum" contain the objective values of the optimal mixed integer solutions. Finally, the column headed "Gap closed" in Table 2 contains the amount of integrality gap that is closed by our algorithm, where the gap closed is defined by (Str. LP - Prep. LP)/(MILP - Prep. LP).

Not all instances in MIPLIB are included in Table 1 and Table 2. The problem flugpl does not involve any binary variables and is therefore excluded from our experiments. Other problems are only described with equalities (for example the instances air04, air05, fiber, markshare1 and markshare2). Since we only attempt to strengthen (structural) inequalities, these problems do not have any coefficients that can be strengthened. There are also problems in MIPLIB that do not have any integrality gap (for instance the problems dsbmip and enigma). Since we measure the quality of a formulation by the amount of integrality gap that remains, we excluded these instances

Problem name	*Number of constraints*	*Number of variables*	*Preprocessed LP value*	*Value of MILP optimum*
10teams	210	1600	897 (=)	904
bell3a	86	100	862117	874375
egout	35	47	242.524	299.001
fixnet6	477	877	3190.04	3981
gt2	28	181	20146.76	21166
khb05250	100	1299	95919464 (=)	106940226
l152lav	97	1988	4656.36 (=)	4722
lseu	28	86	947.96	1120
mas74	13	148	10482.80 (=)	11801.19
mas76	12	148	38893.90 (=)	40005.05
mod008	6	319	290.93 (=)	307
modglob	286	384	19790206 (=)	20099766
opt1217	64	768	-20.02 (=)	-16
pk1	45	86	0 (=)	11
pp08a	133	234	2748.35 (=)	7350
pp08acuts	239	235	5280.61 (=)	7350
qiu	1192	840	-931.639 (=)	-132.873
qnet1o	245	1330	12907.78	16029.69
rgn	24	180	48.799 (=)	82.199
set1ch	423	643	30269.86	49689.50
vpm1	128	188	16.43	20

Table 1: Preprocessed MIPLIB instances that can not be strengthened

as well. Finally, some instances in MIPLIB do not appear in Table 1 and Table 2 because we were unable to provide a formulation that satisfies (*) within a reasonable amount of time.

We first discuss the results in Table 1. For all problems in Table 1, no coefficient on any binary variable that has a positive value in the LP solution can be strengthened. We offer two explanations for this: either these problems are well formulated, or the preprocessor of CPLEX 9.1 strengthens the coefficients that can be strengthened. For the problems 10teams, khb05250, l152lav, mas74, mas76, mod008, modglob, opt1217, pk1, pp08a, pp08acuts, qiu and rgn marked with a "=" sign in Table 1, we note that the integrality gap before and after applying the preprocessor of CPLEX 9.1 remains the same. We suggest this means that these instances are well formulated, and that this explains why no coefficient can be strengthened.

For the remaining problems in Table 1, the preprocessor of CPLEX 9.1 reduced the integrality gap. For some problems, this reduction was quite substantial (the problems egout, fixnet6, gt2, lseu, qnet1o and set1ch are examples of this). It is interesting that the preprocessor of CPLEX 9.1 is able to create formulations that satisfy (*) for these instances, since this is achieved in a very small amount of time.

We next consider the results in Table 2. For all of these problems, some coefficients could be strengthened. For the problems cap6000, misc06, p2756 and vpm2, coefficient strengthening did not close any of the integrality gap. Coefficient strengthening did also not have a large impact on the integrality gap for the instances bell5, dcmulti, gesa2, gesa2o, p0282 and tr12-30. This suggests that these instances are either relatively well formulated, or that the preprocessor of CPLEX 9.1 identifies and modifies the most interesting coefficients that can be strengthened.

For the remaining ten instances, coefficient strengthening significantly reduced the integrality gap. The impact of coefficient strengthening was most dramatic for the problem P0201, where the integrality gap was completely eliminated. In fact, the solution to the strengthened formulation is

Problem name	*Number of constraints*	*Number of variables*	*Preprocessed LP value*	*Strengthened LP value*	*Value of MILP optimum*	Gap closed
bell5	77	94	8341834.36	8343652.39	8699689.24	0.5%
cap6000	2095	5911	-2412601.33	-2412601.33	-2412441	0%
dcmulti	239	515	184034.38	184136.65	188182	2.5%
gen	384	543	58307.89	58334.17	58349.09	63.8%
gesa2	1344	1176	25492512.14	25501347.05	25779856.37	3.1%
gesa2o	1176	1152	18717600.80	18733189.78	19020967.49	5.1%
gesa3	1296	1080	27846449.46	27885122.81	27991042.65	26.7%
gesa3o	1104	1032	12274783.19	12287562.58	12432193.38	8.1%
misc03	95	153	1910	2520.29	3360	42.1%
misc06	461	1317	12841.69	12841.69	12850.86	0%
misc07	211	253	1415	1937.5	2810	37.5%
nsrand-ipx	535	4158	49667.9	50257.6	51200	38.5%
p0033	13	28	2262.55	2428.06	2513	66.1%
p0201	107	183	7155	7615	7615	100%
p0282	160	200	179990.30	180169.64	258401	0.2%
p0548	140	451	4533.81	5275.54	8691	17.8%
p2756	702	2642	2701.67	2701.67	3124	0%
qnet1	363	1417	14274.10	14998.43	16029.69	19.3%
tr12-30	722	1052	13924.2	14324.1	130596	0.2%
vpm2	128	188	11.14	11.14	13.75	0%

Table 2: Preprocessed MIPLIB instances that can be strengthened

integer. Coefficient strengthening therefore solved this problem. Two other interesting examples are the problems misc03 and misc07. For both of these instances, the preprocessor of CPLEX 9.1 did not close any of the integrality gap, whereas roughly 40% of the integrality gap was closed by coefficient strengthening. Finally we note that 38.5% of the integrality gap was eliminated by coefficient strengthening for the problem nsrand-ipx. This is interesting because this is a difficult instance for CPLEX 9.1 to solve. We will investigate this instance in more detail in Section 3.4.

We note that the difficulty of solving the strengthening problems associated with a given instance is not necessarily related to the difficulty of solving the instance. For example, the problems nsrand-ipx and tr12-30 are known to be difficult for CPLEX 9.1. Nevertheless, we were able to solve all the strengthening problems associated with these two instances within a couple of minutes.

Conversely, the instances blend2 and rout are easy problems. However, these instances do not appear in neither Table 1 nor Table 2 because we were unable to solve the associated strengthening problems within a reasonable amount of time.

We conclude that coefficient strengthening can be a useful tool for analyzing the strength of a formulation of a mixed integer program. Our experiments also suggest that it may be possible to improve the performance of the preprocessor of CPLEX 9.1.

3.3 Strengthening the coefficients in mixed integer Gomory cuts

Coefficient strengthening can be applied to any MILP formulation. Coefficient strengthening can therefore also be used on formulations that include cuts. For a given class of cuts, a natural question is whether the coefficients in these cuts can be strengthened. Furthermore, if so, do strengthened cuts lead to significantly stronger formulations than formulations obtained with non-strengthened cuts? We now investigate these questions for mixed integer Gomory (MIG) cuts [8].

Initial experiments that we performed showed that MIG cuts almost always have coefficients that can be strengthened. The purpose of this section is to measure the quality of formulations that can be obtained with strengthened MIG cuts.

We designed the following computational experiment for this purpose. We compare the quality of two formulations obtained from two different cutting plane algorithms. Both cutting plane algorithms use MIG cuts. The difference is that one cutting plane algorithm attempts to strengthen the coefficients in the MIG cuts, whereas the other does not. For simplicity, we call the cutting plane algorithm that applies coefficient strengthening the *strengthened* cutting plane algorithm. The other cutting plane algorithm is called the *pure* cutting plane algorithm.

Both cutting plane algorithms maintain a formulation of the MILP problem. The LP relaxation of the MILP problem is the starting formulation for both algorithms. Five iterations are performed by the two algorithms starting from this initial formulation, where an iteration is defined below. At the start of every iteration of the two cutting plane algorithms, a formulation of the MILP problem is given, and the result of an iteration is a new formulation of the MILP problem.

A *cut pool* is maintained by both cutting plane algorithms. The cut pool is used to store cuts that have been removed from the formulation. The cuts are stored in a cut pool because they might become useful at a later stage of the algorithm. The purpose of the cut pool is to avoid that the size of the formulation becomes too large.

MIG cuts can be derived from an optimal solution x^{lp} to a formulation of the MILP problem (this formulation can contain cuts). Each integer variable with a fractional value in x^{lp} can be used to produce exactly one MIG cut.

The pure cutting plane algorithm can now be described as follows. Every iteration has a starting formulation associated to it, where the LP relaxation of the MILP problem is the starting formulation for the first iteration. The cut pool is initially empty. The following steps are performed in every iteration of the pure cutting plane algorithm.

(1) Let x^{start} be the optimal solution to the current formulation.

    (a) Generate all MIG cuts that can be obtained from x^{start}.

    (b) Find all cuts in the cut pool that are violated by x^{start} (if any).

(2) Add all cuts obtained in step 1 to the current formulation, and re-optimize the corresponding linear program. Let x^{end} be the optimal solution. If x^{end} is integer - STOP.

(3) Find all cuts in the formulation that are *not* satisfied with equality by x^{end}, and move these cuts to the cut pool.

The formulation obtained at the end of an iteration of the pure cutting plane algorithm consists of the original constraints, and the cuts that remain after step 3 has been performed. This formulation is the starting formulation for the next iteration.

The difference between the strengthened cutting plane algorithm and the pure cutting plane algorithm is the following. In the strengthened cutting plane algorithm, the coefficients in the active cuts that remain after step 3 has been performed are strengthened in every iteration. In other words, the strengthened cutting plane algorithm performs the following fourth step in every iteration.

(4) Strengthen the coefficients on the binary variables in the cuts in the current formulation.

The coefficients in the cuts are strengthened with the algorithm described in Section 3.1. Observe that we *only* attempt to strengthen coefficients in cuts. Also, we only consider binary variables that have a positive value in the current LP solution. The question is how much impact step 4 above can have on the quality of the resulting formulation.

We now discuss the computational results. We experienced that the strengthening problems can be difficult to solve for some coefficients in the cuts. This seems to be because the coefficients in MIG cuts can be very fractional. We chose a very high accuracy for the strengthened coefficients to ensure validity (the tolerance was set to 10^{-8}), and this was hard to achieve for some coefficients. For those test problems where we observed that some strengthening problems were difficult to solve, we did not necessarily solve all strengthening problems to optimality. An upper bound on the size of the branch-and-bound tree was enforced on the strengthening problems for these instances.

Table 3 contains the main results of our experiment. All instances in Table 3 (except the problem rout) refer to formulations *after* applying coefficient strengthening to the original formulations. For all test problems, the strengthened cutting plane algorithm closed more integrality gap than the pure cutting plane algorithm. Among all MIPLIB instances, we only included those instances in Table 3 where the difference between the two algorithms was most significant. More precisely, Table 3 contains those instances where the difference in the amount of integrality gap closed by the two algorithms was more than 5%. We note that this implies that problems for which both cutting plane algorithms closed more than 95% of the integrality gap are excluded from Table 3. This was the case for the instances 10teams, egout, gt2, p0033, p0548 and p2756. In particular, the pure cutting plane algorithm (and the strengthened cutting plane algorithm) produced an integer solution after one iteration for the problem p0033.

The first five columns of Table 3 have the same meaning as described earlier for Table 1 and Table 2. The column headed "Gap closed pure" (resp. "Gap closed strengthened") contains the amount of integrality gap that was closed after five iterations of the pure (resp. strengthened) cutting plane algorithm. For some problems, it was necessary to enforce a bound on the size of the branch-and-bound trees created when solving the strengthening problems, and this bound is given in the column headed "Node bound". Finally, the ratio between the amount of integrality gap closed by the strengthened cutting plane algorithm and the amount of integrality gap closed by the pure cutting plane algorithm is given in the column headed "Ratio".

The results in Table 3 demonstrate that strengthened mixed integer Gomory cuts can be substantially stronger than mixed integer Gomory cuts that have not been strengthened. For eleven of the instances in Table 3, the strengthened cutting plane algorithm closed more than twice as much of the integrality gap than the pure cutting plane algorithm. This was most impressive for the problem misc07, where the strengthened cutting plane algorithm closed a factor of 17.8 more of the integrality gap than the pure cutting plane algorithm.

We have only tested the effect of coefficient strengthening on mixed integer Gomory cuts. It would be interesting to investigate the effect of coefficient strengthening when several classes of cuts are used in combination.

3.4 Solving two difficult MIPLIB instances with coefficient strengthening

We now use coefficient strengthening to solve the problems nsrand-ipx and roll3000 of MIPLIB 2003. Table 4 contains the main results. These problems have only recently been solved to optimality (see [10] for a discussion on how to use Xpress-MP to solve these problems).

Problem name	*Number of constraints*	*Number of variables*	*Strengthened LP value*	*MILP value*	*Gap closed Pure*	*Gap closed Strengthened*	*Node bound*	*Ratio*
bell5	77	94	8343652.39	8699689.24	25.0%	76.6%	-	3.1
dcmulti	239	515	184034.38	188182	60.8%	76.2%	-	1.3
fiber	290	1049	186320	394147.58	85.7%	96.4%	10000	1.1
gen	384	543	58334.17	58349.09	25.7%	36.1%	-	1.4
l152lav	97	1988	4656.36	4722	20.0%	49.9%	1000	2.5
lseu	28	86	947.96	1120	41.0%	79.9%	-	1.9
mas74	13	148	10482.80	11801.19	8.2%	33.5%	100000	4.1
mas76	12	148	38893.90	40005.05	7.1%	42.7%	-	6.0
misc03	95	153	2520.29	3360	16.6%	91.2%	-	5.5
misc07	211	253	1937.5	2810	3.4%	60.5%	-	17.8
mod008	6	319	290.93	307	43.2%	65.5%	-	1.5
p0282	160	200	180169.64	258401	14.5%	58.7%	-	4.0
qiu	1192	840	-931.639	-132.873	7.9%	20.8%	1000	2.6
qnet1	363	1417	14998.43	16029.69	26.8%	59.3%	100	2.2
qnet1o	245	1330	12907.78	16029.69	47.4%	53.0%	100	1.1
rgn	24	180	48.799	82.199	33.1%	69.3%	-	2.1
rout(NS)	290	555	-1393.39	-1297.69	8.2%	36.7%	100	4.5
vpm2	128	188	11.14	13.75	34.9%	39.8%	-	1.1

Table 3: Strengthening 5 rounds of mixed integer Gomory cuts

We first discuss how we solve the problem nsrand-ipx. We start with creating the formulation of this instance obtained in Section 3.2. The LP value of the strengthened formulation of this instance is 50257.6 (recall that strengthening the coefficients in the initial formulation of nsrand-ipx closed 38.5% of the initial integrality gap). For this formulation, no coefficient on a variable that has a positive value in the optimal solution of the LP relaxation can be strengthened. The next step we perform on this formulation is to also strengthen the coefficients on the variables that have a value of zero in the optimal solution to the LP relaxation. This does not change the amount of integrality gap that is closed, but does tighten the formulation. As a result, we obtain a formulation of nsrand-ipx for which *no* coefficient of any binary variable can be strengthened (a node bound was not needed).

The pure cutting plane algorithm was then applied for five iterations to give the final formulation. The cuts that were inactive at the end of the cutting plane algorithm were removed from the formulation. The final number of constraints (after adding the cuts) and the objective value of the LP relaxation of the resulting formulation are given in Table 4.

This gives the final formulation that we attempt to solve with CPLEX 9.1. The amount of time used to construct this formulation was roughly two hours. The formulation was given to CPLEX 9.1 with the option "strong branching" and solved. The number of branch-and-bound nodes needed to solve this formulation with this approach and the amount of time needed are given in the last two columns of Table 4. As can be seen in the table, the amount of time used was quite substantial. However, we can mention that CPLEX 9.1 creates a branch-and-bound tree with more than 4 million nodes when solving the problem mas74, even though it does not take more than an hour to solve the problem. Hence, the main reason for the difference in time to solve these two instances is the size of the problems, and *not* the size of the branch-and-bound tree.

We now explain how we solved the problem roll3000. It was not possible to strengthen the coefficients in this formulation as much as possible, because it was too time consuming. Instead, the following simple heuristic for strengthening a coefficient was used. To strengthen the coefficient

Problem name	*Original number of constraints*	*Original number of variables*	*Original LP value*	*MILP value*	*Final number of constraints*	*Final LP value*	*Number of nodes*	*Total time*
nsrand-ipx	535	4158	48880	51200	557	50466.55	171500	42h 33m 10s
roll3000	960	1170	11098.05	12890	1529	11965.64	1485100	486h 28m 37s

Table 4: Computational results for the problems nsrand-ipx and roll3000

on the binary variable x_j in the constraint $a_{k.}^T x \geq b_k$, we solved the linear program $\min\{a_{k.}^T x : x \in P \text{ and } x_j = 1\}$. If the optimal solution x^* satisfies $a_{k.}^T x^* > b_k$, the coefficient $a_{k,j}$ on x_j can be strengthened to $b_k - \sum_{l \in N \setminus \{j\}} a_{k,l} x_l^*$.

The initial formulation is strengthened by strengthening the coefficients on *all* binary variables, *i.e.*, coefficients on variables with a value of zero in the LP solution are also strengthened. As mentioned earlier, this does not change the LP solution, but does improve the quality of the formulation. Also, since a heuristic strengthening procedure is applied, a second pass through the variables might provide further improvements. We therefore iterate the procedure until no coefficients can be strengthened.

After strengthening the initial formulation, ten iterations of the strengthened cutting plane algorithm are performed, where we used the relaxation $R_j = P \cap \{x : x_j \in \{0,1\}\}$ (instead of the mixed integer hull) to strengthen the coefficients on x_j. To improve the quality of the resulting formulation, the following modifications were made to the strengthened cutting plane algorithm. Firstly, coefficients on binary variables that have a value of zero in the LP solution are also strengthened. This increased the computational effort substantially, but proved to be crucial for obtaining a formulation that could be solved with CPLEX 9.1. Secondly, instead of simply deleting the cuts that were not active at the current LP solution, we only deleted cuts that were redundant for the formulation. Given a formulation $\{x \in \mathbb{R}^n : a_{i.}^T x \geq b_i, i \in M\}$ of a mixed integer program, the inequality $a_{k.}^T x \geq b_k$ is redundant for the formulation, if the optimal objective value to the linear program $\min\{a_{k.}^T x : a_{i.}^T x \geq b_i,\ i \in M \setminus \{k\}\}$ is at least b_k. After adding a round of MIG cuts to the formulation, every cut was tested for redundancy in the formulation. This was also an expensive operation, but it was important that all non-redundant cuts were present in the final formulation.

The size and quality of the final formulation are shown in the last row of Table 4. CPLEX 9.1. used 486 hours to solve this formulation to optimality by examining roughly 1.5 million nodes with "strong branching".

4 Strengthening a scheduling formulation

We now use coefficient strengthening to obtain an improved formulation of a specific optimization problem. More precisely, starting from an initial MILP model of the problem, we first produce an improved formulation for some specific instances by using the strengthening procedure. Then we analyze the strengthened formulations to produce an improved MILP model which is valid for all data instances of the problem. We illustrate this on a specific continuous time scheduling problem involving both batch tasks and continuous tasks, which is typical of process and chemical industries. The starting point is an initial model of this scheduling problem. We then generate some (small) instances of this model and strengthen their coefficients. Finally we analyze the strengthened formulations in order to understand how coefficients can be strengthened for general data. In other words, our approach is to use the strengthening procedure as a tool to build tight formulations for a certain problem class.

As a very simplified example, consider the production facility whose process flows are represented in Figure 1 . This production process produces a single product and consists of several stages:

1. An initial product is produced in batches in two identical reactors. The production process of each reactor is characterized by its duration $p = 3$ $[h]$ and its batch size $B = 8$ $[m^3]$, defined as the quantity of the product obtained at the end of a batch. The reactors can process simultaneously.

2. At the end of a batch, the product is discharged from the reactor into a buffer. A new batch can then be started in the reactor. The buffer is characterized by its capacity $\overline{S} = 15$ $[m^3]$, which is the maximum quantity of the product it can contain. Both reactors feed the same buffer.

3. The product is then discharged from the buffer and it is used to perform a continuous task which outputs the finished product. The continuous task is characterized by its minimum and maximum process rates, respectively $\underline{\rho} = 1$ and $\overline{\rho} = 10$ $[m^3/h]$, defined as the minimum and maximum quantity of product processed per unit of time.

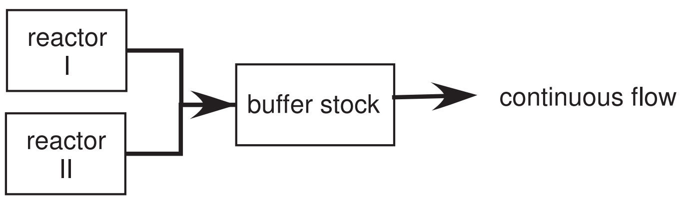

Figure 1: A simple instance of process flows

We refer to Pochet and Warichet [14] for extensions of this simple scheduling problem that involves more stages in the process flow, and in which batch tasks are decomposed into subtasks to model the utilization of scarce resources shared by the different reactors.

The purpose of the short term scheduling problem is to build a production schedule (start times for batch tasks, process rate over time for the continuous task) that maximizes the total production of the finished product. The production schedule can not violate the capacity restrictions for the reservoir.

Traditionally, this scheduling problem is formulated by dividing the scheduling horizon into discrete time periods of uniform length. The drawback of this discrete time formulation is usually that the time period length must be very small in order to accurately model the start times and sequence of events (e.g. start time of a batch task), even though the number of events occurring during the whole time horizon is typically very small. Consequently, the size of the formulation

becomes very large, which makes it very difficult to solve real life instances to optimality.
To overcome this difficulty, continuous time formulations have been proposed in which the scheduling horizon is divided into a number of time slots (time periods with a non-uniform duration). The duration of each time slot is a decision variable, and the end of each time slot corresponds to an event where the status of the process is changed (e.g. start or end of a batch). Moreover, in order to reduce the length of the planning horizon and the size of the resulting formulation, cyclic models where the same schedule is repeated indefinitely have been introduced.

We present here a continuous time formulation for the cyclic scheduling problem corresponding to the simple scheduling instance defines above. This formulation is inspired by Schilling and Pantelides [15]. We also refer to Wu and Ierapetritou [16] for another formulation.

In our instance, the objective is to maximize the total quantity produced by the continuous task over a single cycle (in this instance, the cycle is composed of $T = 4$ time slots) minus the operating costs. The latter are defined as μ times the cycle duration, where the parameter μ represents the fixed operating cost per unit of time that typically comes from the linearization of productivity, see Pochet and Warichet [14].

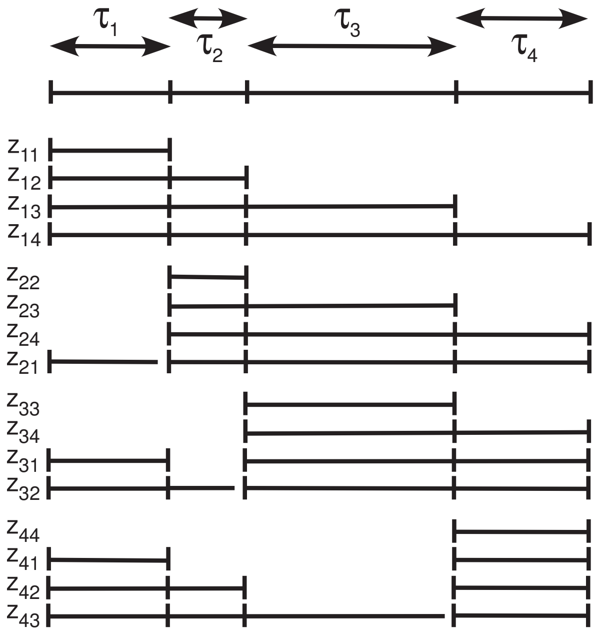

Figure 2: Cyclic batch variables with $T = 4$

To formulate the problem, we use an index $t = 1, \cdots, T$ for the time slots, and the continuous variables $\tau_t \in \mathbb{R}_+$, $1 \leq t \leq T$, to represent the durations of time slots. The variables $z_{kl} \in \{0, 1\}$, $1 \leq k, l \leq T$, model the batch tasks, where $z_{kl} = 1$ when there is a batch task in a reactor that runs from time slot k up to l. The definition of the batch variables z_{kl} is illustrated in Figure 2. In particular, when $z_{kl} = 1$ and $k > l$, the batch runs over time slots k up to T and also time slots 1 up to l in the next cycle. The set $A(t) \subset \{1, \cdots, T\} \times \{1, \cdots, T\}$ denotes the set of batches that are active during time slot t. For instance, $A(2) = \{(1,2), (1,3), (1,4), (2,2), (2,3), (2,4), (2,1), (3,2), (4,2), (4,3)\}$

(see Figure 2).

The other variables, defined for each time slot $t = 1, \cdots, T$, are $q_t \in \mathbb{R}_+$ for the quantity produced by the continuous task, $E_t \in \mathbb{R}_+$ for the inventory level in the reservoir at the end of time slot t, before feeding in the batch just finished, and $S_t \in \mathbb{R}_+$ for the inventory level in the reservoir at the start of time slot t, after feeding in the batch finished at the end of time slot $t-1$. The starting formulation of this cyclic scheduling problem is the following.

$$
\begin{aligned}
\max \quad & \sum_{t=1}^{T} q_t \; - \; \mu\Big(\sum_{t=1}^{T} \tau_t\Big) & & (25)\\
\text{s.t.} \quad & \sum_{(k,l)\in A(t)} z_{kl} \leq 2, & 1 \leq t \leq T, \quad & (26)\\
& \bigoplus_{l=k}^{k-1} z_{kl} \leq 1, & 1 \leq k \leq T, \quad & (27)\\
& \bigoplus_{k=l+1}^{l} z_{kl} \leq 1, & 1 \leq l \leq T, \quad & (28)\\
& \bigoplus_{t=k}^{l} \tau_t \geq p\, z_{kl}, & 1 \leq k,\, l \leq T, \quad & (29)\\
& \bigoplus_{t=k}^{l} \tau_t \leq p\, z_{kl} \; + M\,(1 - z_{kl}), & 1 \leq k,\, l \leq T, \quad & (30)\\
& \underline{\rho}\, \tau_t \leq q_t \leq \overline{\rho}\, \tau_t, & 1 \leq t \leq T, \quad & (31)\\
& 0 \leq S_t \leq \overline{S}, & 1 \leq t \leq T, \quad & (32)\\
& 0 \leq E_t \leq \overline{S}, & 1 \leq t \leq T, \quad & (33)\\
& 0 \leq \tau_t \leq p, & 1 \leq t \leq T, \quad & (34)\\
& E_t = S_t - q_t, & 1 \leq t \leq T, \quad & (35)\\
& S_{t+1} = E_t + B\,\Big(\bigoplus_{k=t+1}^{t} z_{kt}\Big), & 1 \leq t \leq T, \quad & (36)\\
& z_{kl} \in \{0, 1\}, & 1 \leq k,\, l \leq T, \quad & (37)
\end{aligned}
$$

where $\bigoplus$ denotes the cyclic sum, *i.e.*, $\bigoplus_{v=k}^{l} \alpha_v = \sum_{v=k}^{l} \alpha_v$ if $k \leq l$, and $\bigoplus_{v=k}^{l} \alpha_v = \sum_{v=k}^{T} \alpha_v + \sum_{v=1}^{l} \alpha_v$ if $k > l$. Constraints (26) ensure that at most 2 reactors are busy at any time. Constraints (27)-(28) model the restriction that events cannot occur at the same time, *i.e.*, at most one batch can start (resp. finish) at each time slot. Constraints (29)-(30) ensure that $\bigoplus_{t=k}^{l} \tau_t = p$ when $z_{kl} = 1$, where $M = pT = 12$ is an upper bound on the length of a cycle. Constraints (31) express the variable lower and upper bounds on continuous production. Constraints (32)-(34) are the simple bounds on the inventory levels and time slot durations. Finally, constraints (35)-(36) are mass balance constraints for the reservoir.

Unfortunately this smaller formulation compared to the discrete time formulation is weak (The MIP is hard to solve for commercial MIP solvers, in the case of real size instances). We strengthened the right hand sides and the coefficients on all the 0-1 variables as much as possible (using the mixed integer hull as the relaxation R). When one applies the strengthening procedure directly on this formulation and particular instance, one gets an integral linear programming relaxation, and the MIP problem is solved without any branching. This remains true for larger values of T. We checked up to $T = 20$.
Moreover, we were able to analyze, explain and generalize all the improved coefficients produced by coefficient strengthening, as a function of the parameters B, p, $\overline{\rho}$, $\underline{\rho}$ and $\overline{S}$). This therefore provides a tighter formulation for general data and more complex problems, see Pochet and Warichet [14].

As an illustration of the effect of strengthening, we now show two improved constraints. The first is constraint (30) for $(k, l) = (4, 2)$, $p = 3$ and $M = 12$ given by:

$$\tau_4 + \tau_1 + \tau_2 \leq 3\, z_{42} \; + 12\, (1 - z_{42}) = 12 - 9\, z_{42}$$

which was improved to the following inequality:

$$\tau_4 + \tau_1 + \tau_2 \leq 9 - 6\, z_{42} - 3\, z_{12} \; .$$

A first improvement is to reduce the value of the big M to $3p = 9$. This generates a RHS equal to $(9 - 6\, z_{42})$. The RHS can be further reduced by observing that, if $z_{12} = 1$, then $z_{42} = 0$ (by (28)) and $(\tau_4 + \tau_1 + \tau_2) \leq 6$ (because $(\tau_1 + \tau_2) = 3$ from (29)-(30) and $\tau_4 \leq 3$ from (34)). This and similar explanations allowed us to generalize the strengthening of constraints (30).

The second example is constraint (29) for $(k, l) = (2, 4)$ and $p = 3$ given by:

$$\tau_2 + \tau_3 + \tau_4 \geq 3\, z_{24}$$

which was improved to the following inequality:

$$\tau_2 + \tau_3 + \tau_4 \geq 3\, (z_{22} + z_{23} + z_{24}) + 3\, z_{44} + 0.1\, z_{34} \; .$$

The term $3\, (z_{22} + z_{23} + z_{24})$ comes from the fact that $z_{22} + z_{23} + z_{24} \leq 1$ by (27) and, if $z_{22} + z_{23} + z_{24} = 1$, then $\tau_2 + \tau_3 + \tau_4 \geq 3$ by (29)-(30) and $\tau_t \geq 0$ for all t. For instance, $z_{22} = 1$ implies $\tau_2 + \tau_3 + \tau_4 \geq \tau_2 = 3$.
The term $3\, z_{44}$ can be explained in a similar way as above, i.e. using (28)-(30). If $z_{44} = 1$, then $\tau_4 = p = 3$, and if, in addition, $z_{22} + z_{23} + z_{24} = 1$, then we must have $z_{24} = 0$, $z_{22} + z_{23} = 1$, $\tau_2 + \tau_3 \geq p = 3$, and hence $\tau_2 + \tau_3 + \tau_4 \geq 6$.
The explanation of the coefficient $0.1\, z_{34}$ is more involved. If $z_{34} = 1$, then $(\tau_3 + \tau_4) = 3$ from (29)-(30), $z_{24} = z_{44} = 0$ from (28), and from (27) we have

1. either $z_{22} = z_{23} = 0$ and the inequality is valid because $(\tau_2 + \tau_3 + \tau_4) \geq (\tau_3 + \tau_4) = 3 \geq 0.1$,
2. or $z_{22} = 1$ and the inequality is valid because $(\tau_2) + (\tau_3 + \tau_4) = 3 + 3 \geq 3.1$,
3. or $z_{23} = 1$ and the inequality is valid because
   (a) $z_{23} = 1$ implies $S_4 \geq B = 8$ (one batch is finished at the end of time slot 3, and put in inventory at the beginning of time slot 4),

(b) $z_{34} = 1$ implies $E_4 \leq \overline{S} - B = 15 - 8 = 7$ (one batch is finished at the end of time slot 4, and there must be enough storage space available to stock it at the start of time slot 1),

(c) $E_4 = S_4 - q_4$ by (35) implies $q_4 \geq 1$ (at least 1 unit of product must be consumed during time slot 4),

(d) constraint (31) implies $\tau_4 \geq \frac{1}{\overline{\rho}} = 0.1$,

(e) and therefore $(\tau_2 + \tau_3) + (\tau_4) \geq 3 + 0.1$.

So we see that, in this example, the improved coefficient of 0.1 on z_{34} in the second example is given by $\frac{2B-\overline{S}}{\overline{\rho}}$. This shows the way of possible generalizations.

Similar results have been obtained for other scheduling problems of the same type. In all cases, the strengthening procedure suggested improved formulations. This indicates that coefficient strengthening can be a powerful tool for constructing tight formulations for practical problems.

5 Concluding Remarks

We conclude that coefficient strengthening can be a useful tool for analyzing the strength of a formulation of a mixed integer program. First, our experiments suggest that it may be possible to improve the performance of the preprocessor of CPLEX 9.1, and to tighten the cuts produced by commercial MIP solvers. Next, we have illustrated how strengthening can be used to reveal properties of a specific production scheduling MIP problem, and thereby provide information on how to construct tighter formulations of the problem.

It might be useful for practitioners of integer programming to have a tool for coefficient strengthening. This tool could be very useful in modeling practical problems by testing the strength of original formulations and by identifying coefficients that can be strengthened.

References

[1] E. Balas. Intersection cuts - a new type of cutting planes for integer programming. *Operations Research*, 19:19–39, 1971.

[2] E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.

[3] R.E. Bixby, S. Ceria, C.M. McZeal, and M.W.P. Savelsbergh. An updated mixed integer programming library: MIPLIB 3.0. *Technical report TR98-03, Department of Computational and Applied Mathematics, Rice University*, 1998.

[4] G. Bradley, P. Hammer, and L. Wolsey. Coefficient reduction for inequalities in 0-1 variables. *Mathematical Programming*, 7:263–282, 1974.

[5] A.L. Brearley, G. Mitra, and H.P. Williams. An Analysis of Mathematical Programs Prior to Applying the Simplex Method. *Mathematical Programming*, 7:263–282, 1973.

[6] W. Cook, R. Kannan, and A. Schrijver. Chvátal closures for mixed integer programs. *Mathematical Programming*, 47:155–174, 1990.

[7] CPLEX optimizer version 8.0, ILOG.

[8] R.E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, The Rand Corporation, Santa Monica, CA, 1960.

[9] K. Hoffman and M. Padberg. Improving Representation of Zero-One Linear Programs for Branch-and-Cut. *ORSA Journal of Computing*, 3:121–134, 1991.

[10] R. Laundy, M. Perregaard, G. Tavares, H. Tipi, and A. Vazacopoulos. Solving hard integer programming problems with Xpress-MP: A MIPLIB 2003 case study. *Rutcor Reseach Report*, 2007.

[11] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley, 1988.

[12] G.L. Nemhauser and L.A. Wolsey. A recursive procedure for generating all cuts for 0-1 mixed integer programs. *Mathematical Programming*, 46:379–390, 1990.

[13] M.W.P. Nemhauser. Preprocessing and Probing Techniques for Mixed Integer Programming Problems. *ORSA Journal of Computing*, 6:445–454, 1994.

[14] Y. Pochet and F. Warichet. A tighter continuous time formulation for the cyclic scheduling of a mixed plant. *CORE Discussion Paper, UCL, Belgium*, 2007.

[15] G. Schilling and C.C. Pantelides. Optimal periodic scheduling of multipurpose plants. *Computers Chem. Engng*, 23:635–655, 1999.

[16] D. Wu and M. Ierapetritou. Cyclic short-term scheduling of multiproduct batch plants using continuous time representation. *Computers Chem. Engng*, 28:2271–2286, 2004.

Appendix. Proofs

PROOF OF LEMMA 1. We have that $\alpha^T x \leq \beta$ is valid for $P(M \setminus \{k\}) \iff \max\{\alpha^T x : x \in P(M \setminus \{k\})\} \leq \beta \iff$ (from LPduality) $\min\{-\sum_{i\in M\setminus\{k\}} w_i b_i : \alpha = -\sum_{i\in M\setminus\{k\}} w_i a_{i.}, w_i \geq 0 \text{ for } i \in M^{\geq} \setminus \{k\}\} \geq \beta \iff$ there exists a solution to (7) $-$ (10) in which $w_0 = 0$. This shows (ii).

We now show (i). First suppose $\alpha^T x \geq \beta$ dominates $a_{k.}^T x \geq b_k$ on P. Consider the linear program (LP$'$)

$$\begin{aligned}
\min\ & a_{k.}^T x \\
\text{s.t.}\ & a_{i.}^T x \geq b_i, \qquad \forall i \in M^{\geq} \setminus \{k\} \qquad (w_i)\\
& a_{i.}^T x = b_i, \qquad \forall i \in M^{=}, \qquad (w_i)\\
& \alpha^T x \geq \beta. \qquad (w_0)
\end{aligned}$$

Since $P(M\setminus\{k\}) \cap \{x \in \mathbb{R}^n : \alpha^T x \geq \beta\} \neq \emptyset$, the problem LP$'$ is feasible. Further, since $\alpha^T x \geq \beta$ dominates $a_{k.}^T x \geq b_k$ on P, LP$'$ is bounded from below by b_k. By linear programming duality, there exists w_0' and $\{w_i'\}_{i\in M\setminus\{k\}}$ such that $\sum_{i\in M\setminus\{k\}} w_i' b_i + w_0'\beta \geq b_k$ and $\sum_{i\in M\setminus\{k\}} w_i' a_{i.} + w_0'\alpha = a_{k.}$, where $w_0' \geq 0$ and $w_i' \geq 0$ for $i \in M^{\geq} \setminus \{k\}$. If $w_0' = 0$, then $a_{k.}^T x = \sum_{i\in M\setminus\{k\}} w_i' a_{i.}^T x \geq$

$\sum_{i\in M\setminus\{k\}} w'_i b_i \geq b_k$ for any $x \in P(M \setminus \{k\})$. In other words, if $w'_0 = 0$, then $a_{k.}^T x \geq b_k$ is redundant for P. We therefore have $w'_0 > 0$. Defining $w_0 := \frac{1}{w'_0}$ and $w_i := \frac{w'_i}{w'_0}$ gives w_0 and $\{w_i\}_{i\in M\setminus\{k\}}$ that satisfy (7) – (10).

Conversely, suppose $\{w_i\}_{i\in M\setminus\{k\}}$ and $w_0 > 0$ satisfy (7) – (10). For any $x' \in P(M \setminus \{k\})$ such that $\alpha^T x' \geq \beta$, we have $w_0 a_{k.}^T x' \geq \alpha^T x' + \sum_{i\in M\setminus\{k\}} w_i a_{i.}^T x' \geq \beta + \sum_{i\in M\setminus\{k\}} w_i b_i \geq w_0 b_k$. Since $w_0 > 0$, it follows that $\alpha^T x \geq \beta$ dominates $a_{k.}^T x \geq b_k$ on P.

□

PROOF OF PROPOSITION 1. First suppose there exists $x^I \in P_I$ such that $a_{l.}^T x^I > b_l$. Then $(y^I, \lambda^I) := (\frac{x^I}{s_l^I}, \frac{1}{s_l^I})$ is feasible for DP$_{k,l}$, where $s_l^I := a_{l.}^T x^I - b_l$ denotes the value of the surplus variable in the l^{th} constraint. Note that the fact that $x^I = \frac{y^I}{\lambda^I}$ is integer implies that (20) is satisfied. Hence, if there exists $x^I \in P_I$ satisfying $a_{l.}^T x^I > b_l$, then DP$_{k,l}$ is feasible.

Now suppose DP$_{k,l}$ is feasible. Let (y^D, λ^D) be an arbitrary feasible solution to DP$_{k,l}$. There are two cases. Either it is possible to choose (y^D, λ^D) such that $\lambda^D > 0$, or every feasible solution (y^D, λ^D) to DP$_{k,l}$ satisfies $\lambda^D = 0$. If it is possible to choose (y^D, λ^D) such that $\lambda^D > 0$, then $x^D := \frac{x^D}{\lambda^D} \in P_I$ and $a_{l.}^T x^D = \frac{(1+\lambda^D b_l)}{\lambda^D} = \frac{1}{\lambda^D} + b_l > b_l$. It therefore only remains to consider the case when every feasible solution (y^D, λ^D) to DP$_{k,l}$ satisfies $\lambda^D = 0$. This means the problem DP$_{k,l}$ reduces to the linear program

$$
\begin{aligned}
\min \quad & a_{k.}^T y \\
\text{s.t.} \quad & a_{i.}^T y \geq 0, \quad \forall i \in M^{\geq}, && (38)\\
& a_{i.}^T y = 0, \quad \forall i \in M^{=}, && (39)\\
& a_{l.}^T y = 1, && (40)\\
& y_j = 0, \quad \forall j \in N_I. && (41)
\end{aligned}
$$

Since we assumed DP$_{k,l}$ is feasible, there exists $y^r \in \mathbb{R}^n$ that satisfies (38)-(41). This implies that, given any $x^I \in P_I$, we have $x^I + y^r \in P_I$ and $a_{l.}^T(x^I + y^r) > b_l$.

□

To prove Proposition 2, we first show that $w_{k,l}^{dp}$ provides a valid coefficient for the variable $s_l = a_{l.}^T x - b_l$ in the constraint $a_{k.}^T x \geq b_k$.

Lemma 4 *Assume DP$_{k,l}$ is feasible. The inequality $a_{k.}^T x - w_{k,l}^{dp}(a_{l.}^T x - b_l) \geq b_k$ is valid for P_I, where $w_{k,l}^{dp}$ denote the optimal objective value of DP$_{k,l}$.*

PROOF. Let $x^I \in P_I$ be arbitrary. If $a_{l.}^T x^I = b_l$, we have $a_{k.}^T x^I - w_{k,l}^{dp}(a_{l.}^T x^I - b_l) \geq b_k$, so assume $a_{l.}^T x^I > b_l$. Define $\lambda^I := \frac{1}{a_{l.}^T x^I - b_l}$ and $y^I := \lambda^I x^I$. Clearly (y^I, λ^I) is feasible for DP$_{k,l}$, so we have $w_{k,l}^{dp} \leq a_{k.}^T y^I - \lambda^I b_k$. Multiplying with $a_{l.}^T x^I - b_l > 0$ on both sides of this inequality gives $w_{k,l}^{dp}(a_{l.}^T x^I - b_l) \leq a_{k.}^T x^I - b_k$, which implies $a_{k.}^T x^I - w_{k,l}^{dp}(a_{l.}^T x^I - b_l) \geq b_k$.

□

Lemma 4 shows the relation $w^*_{k,l}(Conv(P_I)) \geq w^{dp}_{k,l} \geq 0$. To prove prove Proposition 2, we also need the following lemma that concerns certificates for when a coefficient can not be improved.

Lemma 5 *Let R be a polyhedral relaxation of P_I satisfying $P_I \subseteq R \subseteq P$. Let $k, l \in M^{\geq}$, $k \neq l$, and assume $w^*_{k,l}(R) = 0$. Then either (i) or (ii) holds.*

(i) $b^*_k(R) > b_k$ (the right hand side of $a^T_{k.}x \geq b_k$ can be increased).

(ii) $\exists x^R \in R$ s.t. $a^T_{k.}x^R = b_k$ and $a^T_{l.}x^R > b_l$ (there is a certificate $x^R \in R$ showing $w^*_{k,l}(R) = 0$).

PROOF. Assume (i) does not hold, i.e. we have $b^*_k(R) = b_k$. Let $x^C \in R$ be a certificate satisfying $a^T_{k.}x^C = b_k$. We will prove there exists $x^R \in R$ satisfying $a^T_{k.}x^R = b_k$ and $a^T_{l.}x^R > b_l$, i.e. a certificate showing $w^*_{k,l}(R) = 0$.

Let the extreme points of R be $\{v^r\}_{r \in V^R}$, and let the extreme rays be $\{e^s\}_{s \in E^R}$, where V^R and E^R are finite index sets. We have $R = Conv(\{v^r\}_{r \in V^R}) + Cone(\{e^s\}_{s \in E^R})$. An inequality $\alpha^T x \geq \beta$ is valid for R if and only if $\alpha^T v^r \geq \beta$ for all $r \in V^R$, and $\alpha^T e^s \geq \beta$ for all $s \in E^R$. It follows that $w^*_{k,l}(R)$ is the optimal objective value to the linear program

$$\begin{array}{llll} \max & w_{k,l} & & \\ \text{s.t.} & w_{k,l}(a^T_{l.}v^r - b_l) \leq & a^T_{k.}v^r - b_k, & \forall r \in V^R, \\ & w_{k,l}(a^T_{l.}e^s) \quad \leq & a^T_{k.}e^s, & \forall s \in E^R. \end{array}$$

We have $a^T_{k.}v^r - b_k \geq 0$, $a^T_{l.}v^r - b_l \geq 0$, $a^T_{k.}e^s \geq 0$ and $a^T_{l.}e^s \geq 0$ for all $(r, s) \in V^R \times E^R$, since $R \subseteq P$. Furthermore, since $w^*_{k,l}(R) = 0$, we have either (i) There exists $r \in V^R$ such that $a^T_{k.}v^r - b_k = 0$ and $a^T_{l.}v^r - b_l > 0$, or (ii) There exists $s \in E^R$ such that $a^T_{k.}e^s = 0$ and $a^T_{l.}e^s > 0$. If (i) holds, let $x^R := v^r$, and if (ii) holds, let $x^R := x^C + e^s$. We have $x^R \in R$, $a^T_{k.}x^R = b_k$ and $a^T_{l.}x^R > b_l$.

□

We are now ready to give the proof of Proposition 2.

PROOF OF PROPOSITION 2. Assume $\mathrm{DP}_{k,l}$ is feasible, and that the right hand side of $a^T_{k.}x \geq b_k$ can not be improved. Then there exists $x^I \in P_I$ satisfying $a^T_{k.}x^I = b_k$. Also, let $\{(y^t, \lambda^t)\}_{t \in \mathbb{N}}$ be a sequence of feasible solutions to $\mathrm{DP}_{k,l}$ that satisfies $\lim_{t \to \infty}(a^T_{k.}y^t - \lambda^t b_k) = w^{dp}_{k,l}$. By extracting a convergent subsequence, we can assume either $\lambda^t = 0$ for all $t \in \mathbb{N}$, or $\lambda^t > 0$ for all $t \in \mathbb{N}$.

First suppose $\lambda^t = 0$ for all $t \in \mathbb{N}$. Then for every $t \in \mathbb{N}$, y^t is a solution to the system (42)-(45).

$$a^T_{i.}y \geq 0, \quad \forall i \in M^{\geq}, \tag{42}$$

$$a^T_{i.}y = 0, \quad \forall i \in M^{=}, \tag{43}$$

$$a^T_{l.}y = 1, \tag{44}$$

$$y_j = 0, \quad \forall j \in N_I. \tag{45}$$

Since $a_{k.}^T y$ is continuous, $\lim_{t\to\infty} a_{k.}^T y^t = w_{k,l}^{dp}$ and the set (42)-(45) is closed, there exists $y^* \in \mathbb{R}^n$ satisfying (42)-(45) and $a_{k.}^T y^* = w_{k,l}^{dp}$. Hence $(y^*, 0)$ is both feasible and optimal for $\text{DP}_{k,l}$, which shows (ii). In addition, we have $z^I := x^I + y^* \in P_I$ and $a_{l.}^T z^I = a_{l.}^T x^I + a_{l.}^T y^* = a_{l.}^T x^I + 1 > b_l$.

First suppose $a_{k.}^T z^I = b_k$. Then $a_{k.}^T y^* = w_{k,l}^{dp} = 0$. Using Lemma 5.(ii) then gives $w_{k,l}^*(Conv(P_I)) = 0$, and applying Lemma 4 gives $0 = w_{k,l}^*(Conv(P_I)) \geq w_{k,l}^{dp} \geq 0$, which shows (i).

Now suppose $a_{k.}^T z^I > b_k$. This implies $a_{k.}^T y^* = w_{k,l}^{dp} > 0$. Observe that we must have $a_{l.}^T x^I = b_l$, since otherwise the inequality $a_{k.}^T x - w_{k,l}^{dp}(a_{l.}^T x - b_l) \geq b_k$ would cut off x^I, which contradicts Lemma 4. Also observe that Lemma 4 implies $w_{k,l}^*(Conv(P_I)) \geq w_{k,l}^{dp}$. Now, the inequality $a_{k.}^T x - (w_{k,l}^{dp} + \epsilon)(a_{l.}^T x - b_l) \geq b_k$ is not valid for P_I for any $\epsilon > 0$, since $a_{k.}^T z^I - (w_{k,l}^{dp} + \epsilon)(a_{l.}^T z^I - b_l) = b_k + w_{k,l}^{dp} - (w_{k,l}^{dp} + \epsilon)(a_{l.}^T x^I + a_{l.}^T y^* - b_l) = b_k + w_{k,l}^{dp} - (w_{k,l}^{dp} + \epsilon) < b_k$. It follows that $w_{k,l}^{dp}$ is the largest possible coecfficient on $s_l := a_{l.}^T x - b_l$, which means $w_{k,l}^*(Conv(P_I)) = w_{k,l}^{dp}$ by definition, and therefore (i) holds.

The final case to consider is when $\lambda^t > 0$ for all $t \in \mathbb{N}$. We first show this implies (i), i.e. that $w_{k,l}^*(Conv(P_I)) = w_{k,l}^{dp}$. This is done by showing $a_{k.}^T x - (w_{k,l}^{dp} + \epsilon)(a_{l.}^T x - b_l) \geq b_k$ is *not* valid for P_I for any $\epsilon > 0$. Let $\epsilon > 0$ be arbitrary. Since $\lim_{t\to\infty}(a_{k.}^T y^t - \lambda^t b_k) = w_{k,l}^{dp}$, there exists $t^* \in \mathbb{N}$ such that $a_{k.}^T y^{t^*} - \lambda^{t^*} b_k < w_{k,l}^{dp} + \epsilon$. Define $x^* := \frac{y^{t^*}}{\lambda^{t^*}}$. We have $x^* \in P_I$ and $\lambda^{t^*} = \frac{1}{a_{l.}^T x^* - b_l}$. Multiplying with $a_{l.}^T x^* - b_l > 0$ on both sides of the inequality $w_{k,l}^{dp} + \epsilon > a_{k.}^T y^{t^*} - \lambda^{t^*} b_k$ gives $(a_{l.}^T x^* - b_l)(w_{k,l}^{dp} + \epsilon) > a_{k.}^T x^* - b_k$, which implies $a_{k.}^T x^* - (w_{k,l}^{dp} + \epsilon)(a_{l.}^T x^* - b_l) < b_k$. It follows that $w_{k,l}^{dp} = w_{k,l}^*(Conv(P_I))$.

We now show (ii), *i.e.*, that the optimal objective value $w_{k,l}^{dp} = w_{k,l}^*(Conv(P_I))$ is obtained by a feasible solution to $\text{DP}_{k,l}$. Since the coefficient on $(a_{l.}^T x - b_l)$ in $a_{k.}^T x - w_{k,l}^*(Conv(P_I))(a_{l.}^T x - b_l) \geq b_k$ can *not* be improved, Lemma 5.(ii) implies that there exists $x^{k,l} \in P_I$ that satisfies $a_{l.}^T x^{k,l} > b_l$ and $a_{k.}^T x^{k,l} - w_{k,l}^*(Conv(P_I))\ (a_{l.}^T x^{k,l} - b_l) = b_k$. Define $\lambda^{k,l} := \frac{1}{a_{l.}^T x^{k,l} - b_l}$ and $y^{k,l} := \lambda^{k,l} x^{k,l}$. We have that $(y^{k,l}, \lambda^{k,l})$ is feasible for $\text{DP}_{k,l}$ and $a_{k.}^T y^{k,l} - \lambda^{k,l} b_k = w_{k,l}^*(Conv(P_I))$.

□

PROOF OF COROLLARY 1. Suppose, for a contradiction, that the formulation LP of MILP is optimal relative to R and sequential strengthening, and that the valid inequality $(\alpha')^T x \geq \beta'$ for R strictly dominates the inequality constraint $a_{k.}^T x \geq b_k$ on P. Then Lemma 1.(i) shows that there exists $w_0' > 0$ and $\{w_i'\}_{i\in M\setminus\{k\}}$ such that (7)-(10) are satisfied. In other words, we have $(\alpha', \beta', w_0', \{w_i'\}_{i\in M\setminus\{k\}}) \in \text{RC}_k(R)$. By scaling, if necessary, we may assume $w_0' = 1$. Applying Theorem 1 gives $\text{RC}_k(R) = \text{TRC}_k(R)$. However, this implies $(\alpha')^T x \geq \beta'$ is equivalent to $a_{k.}^T x \geq b_k$ on P.

□

PROOF OF LEMMA 2. We have $\alpha^r = -\sum_{i\in M\setminus\{k\}} w_i^r a_{i.}$, $\beta^r \geq -\sum_{i\in M\setminus\{k\}} w_i^r b_i$ and $(\alpha^r)^T x = \beta^r$ for all $x \in R$. Let $W^S := \{i \in M^{\geq} \setminus \{k\} : w_i^r > 0\}$. There are three cases.

(a) $\beta^r > -\sum_{i\in M\setminus\{k\}} w_i^r b_i$: then for every $x \in R$, we have $(\alpha^r)^T x = -\sum_{i\in M\setminus\{k\}} w_i^r a_{i.}^T x \leq -\sum_{i\in M\setminus\{k\}} w_i^r b_i < \beta^r$, which implies $R = \emptyset$.

(b) $\beta^r = -\sum_{i\in M\setminus\{k\}} w_i^r b_i$ and $W^S \neq \emptyset$: If $a_{i.}^T x' > b_i$ for some $x' \in R$ and $i \in W^S$, then $(\alpha^r)^T x' = -\sum_{i\in M\setminus\{k\}} w_i^r a_{i.}^T x' < -\sum_{i\in M\setminus\{k\}} w_i^r b_i = \beta^r$. It follows that $a_{i.}^T x = b_i$ for all $x \in R$ and $i \in W^S$, which means all surplus variables s_i, $i \in W^S$, can be fixed to zero.

(c) $\beta^r = -\sum_{i\in M\setminus\{k\}} w_i^r b_i$ and $W^S = \emptyset$: This implies $w_i^r = 0$ for all $i \in M^{\geq} \setminus \{k\}$, $\alpha^r = -\sum_{i\in M^=} w_i^r a_{i.}$ and $\beta^r = -\sum_{i\in M^=} w_i^r a_{i.}$, which implies $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i\in M\setminus\{k\}}) \in \mathrm{TRC}_k(R)$.

□

PROOF OF LEMMA 3. By using (7), $(\alpha^r)^T x \geq \beta^r$ can be written as $a_{k.}^T x - \sum_{i\in M\setminus\{k\}} w_i^r a_{i.}^T x \geq \beta^r$. Observe that, since $\beta^r \geq b_k - \sum_{i\in M\setminus\{k\}} w_i^r b_i$, the inequality $(\alpha^r)^T x \geq \beta^r$ dominates the inequality $a_{k.}^T x - \sum_{i\in M\setminus\{k\}} w_i^r (a_{i.}^T x - b_i) \geq b_k$. Furthermore, since every $x \in R$ satisfies $a_{i.}^T x = b_i$ for all $i \in M^=$, we have that the inequality

$$a_{k.}^T x - \sum_{i\in M^{\geq}\setminus\{k\}} w_i^r (a_{i.}^T x - b_i) \geq b_k \tag{46}$$

is valid for R. Assume there exists $l \in M \setminus \{k\}$ such that $w_l^r > 0$. Then the inequality $a_{k.}^T x - w_l^r (a_{l.}^T x - b_l) \geq b_k$ is dominated by (46) on R, and therefore $a_{k.}^T x - w_l^r (a_{l.}^T x - b_l) \geq b_k$ is valid for R, which implies (ii).

For the remaining cases we can assume $w_i^r = 0$ for all $i \in M^{\geq} \setminus \{k\}$. This implies that $\alpha^r = a_{k.} - \sum_{i\in M^=} w_i^r a_{i.}$ and $\beta^r \geq b_k - \sum_{i\in M^=} w_i^r b_i$. If $\beta^r = b_k - \sum_{i\in M^=} w_i^r b_i$, then $(\alpha^r, \beta^r, w_0^r, \{w_i^r\}_{i\in M\setminus\{k\}}) \in \mathrm{TRC}_k(R)$, which implies (iii). So suppose $\beta^r > b_k - \sum_{i\in M^=} w_i^r b_i$. This implies that every $x \in R$ satisfies $a_{k.}^T x = (\alpha^r)^T x + \sum_{i\in M^=} w_i^r a_{i.}^T x \geq \beta^r + \sum_{i\in M^=} w_i^r b_i > b_k$. It follows that the optimal objective value b'_k of the linear program $\min\{a_{k.}^T x : x \in R\}$ satisfies $b'_k > b_k$, which implies (iii).

□

We now prove the missing direction of Theorem 1.

Lemma 6 *Suppose $RC_k(R) = TRC_k(R)$ for every $k \in M^{\geq}$. Then the formulation LP of MILP is optimal relative to R and sequential strengthening.*

PROOF. The proof is by contradiction. Suppose $\mathrm{RC}_k(R) = \mathrm{TRC}_k(R)$ for every $k \in M^{\geq}$. Assume there exists $k, l \in M^{\geq}$, $k \neq l$, such that $w_{k,l}^*(R) = \max\{w_{k,l} : a_{k.}^T x - w_{k,l}(a_{l.}^T x - b_l) \geq b_k$ is valid for R$\} > 0$. Define $\alpha' := a_{k.} - w_{k,l}^*(R) a_{l.}$, $\beta' := b_k - w_{k,l}^*(R) b_l$, $w'_0 := 1$, $w'_l := w_{k,l}^*(R) > 0$ and $w'_i = 0$ for $i \in M \setminus \{k, l\}$. Then we have $(\alpha', \beta', w'_0, \{w'_i\}_{i\in M\setminus\{k\}}) \in \mathrm{RC}_k(R) \setminus \mathrm{TRC}_k(R)$.

Similarly if $b_k^*(R) = \max\{\beta : a_{k.}^T x \geq \beta$ is valid for R$\} > b_k$, then $(\alpha'', \beta'', w''_0, \{w''_i\}_{i\in M\setminus\{k\}}) \in \mathrm{RC}_k(R) \setminus \mathrm{TRC}_k(R)$, where $\alpha'' := a_{k.}$, $\beta'' := b_k^*$, $w''_0 := 1$ and $w''_i := 0$ for $i \in M \setminus \{k\}$.

□