Improved Rank-One-Based Relaxations and Bound Tightening Techniques for the Pooling Problem

Mosayeb Jalilian* Burak Kocuk†

June 18, 2023

Abstract

The pooling problem is a classical NP-hard problem in the chemical process and petroleum industries. This problem is modeled as a nonlinear, nonconvex network flow problem in which raw materials with different specifications are blended in some intermediate tanks, and mixed again to obtain the final products with desired specifications. The analysis of the pooling problem is quite an active research area, and different exact formulations, relaxations and restrictions are proposed. In this paper, we focus on a recently proposed rank-one-based formulation of the pooling problem. In particular, we study a recurring substructure in this formulation defined by the set of nonnegative, rank-one matrices with bounded row sums, column sums, and the overall sum. We show that the convex hull of this set is second-order cone representable. In addition, we propose an improved compact-size polyhedral outer-approximation and families of valid inequalities for this set. We further strengthen these convexification approaches with the help of various bound tightening techniques specialized to the instances of the pooling problem. Our computational experiments show that the newly proposed polyhedral outer-approximation can improve upon the traditional linear programming relaxations of the pooling problem in terms of the dual bound. Furthermore, bound tightening techniques reduce the computational time spent on both the exact, linear programming and mixed-integer linear programming relaxations.

1 Introduction

The classical blending problem that appears in many industrial settings involves determining the optimal blend of raw materials to produce a certain quantity of end products with minimum cost. This problem determines the proportions of the raw materials used in different products considering the incoming raw materials' specifications. The blending problem is polynomially solvable since it can be modeled as a compact-size linear program (LP).

When the raw materials are blended in intermediate tanks and then mixed again to form the end products, the problem becomes considerably more challenging to solve due to its nonconvex nature. This problem is known as the *pooling problem* and is one of the main problems in the chemical process and petroleum industries. The problem involves three types of tanks: inputs or sources to store raw materials, pools or intermediates to blend incoming flow streams and create new compositions, and outputs or terminals to store the final products. There are two classes of pooling problems based on the links among the different tanks. The standard pooling problem

*mosayeb.jalilian@neoma-bs.fr, Faculty of Supply Chain and Operations Management, NEOMA Business School, 76130 Mont-Saint-Aignan, France. This research was conducted when the first author was a master's student at Sabancı University.

†burakkocuk@sabanciuniv.edu, Industrial Engineering Program, Sabancı University, Istanbul, Turkey 34956.

has no flow stream among the pools, and the flow streams are source-to-terminal, source-to-pool, and pool-to-terminal. A typical *standard pooling problem* instance is shown in Figure 1. On the other hand, in the generalized pooling problem, flow streams between the pools are allowed, which make the problem even more challenging. This class was introduced by Audet et al. (2004), and an instance of a simple generalized pooling problem is shown in Figure 2.

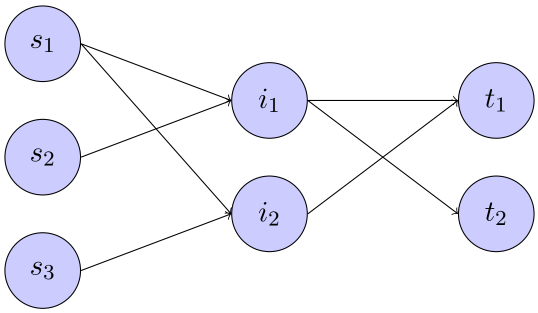

Figure 1: A standard pooling problem instance.

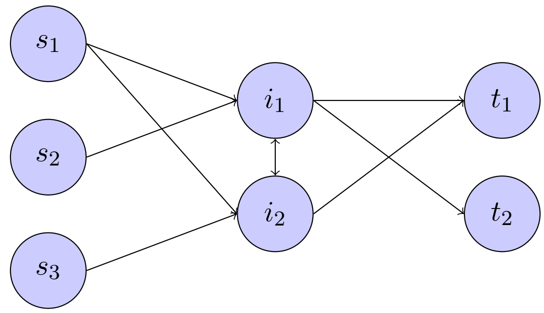

Figure 2: A generalized pooling problem instance.

The analysis of the pooling problem has become an active research area since its introduction by Haverly (1978). The nonconvexity of the problem arises due to keeping track of specifications throughout the network, leading to the potential existence of multiple local optima (Alfaki and Haugland, 2013a). Therefore, researchers have developed various exact formulations, relaxations, and heuristics to solve the problem.

The *P-formulation* was introduced by Haverly (1978), which modeled the problem using flow and pool attribute quality variables. The authors used the Alternating Method to solve the problem recursively, where an LP model was generated using an estimation of the pool qualities. More recently, Boland et al. (2015) studied a problem consisting of multi-period variables which arises in the mining industry as a special case of the generalized pooling problem based on the *P-formulation*.

Another formulation, the *Q-formulation*, was proposed by Ben-Tal et al. (1994), which used variables representing the relative proportions of pool input flows instead of the flow variables of pools in the *P-formulation*. The authors derived a general principle that can reduce or eliminate the duality gap of a nonconvex program and its Lagrangian dual in some special cases by partitioning the feasible set. They used this principle to compute a near-optimal solution that provides a primal bound for three different versions of the instance produced by Haverly (1978).

Additionally, Audet et al. (2004) proposed a hybrid formulation, which consists of the quality variable from the *P-formulation* and the proportion variable from the *Q-formulation* in addition to the flow variables. Tawarmalani and Sahinidis (2002) added some valid constraints to express mass balances across pools, creating the *PQ-formulation*. This formulation has proportion variables corresponding to sources and flow variables along the arcs between pools and terminals (Alfaki and Haugland, 2013b). The authors relaxed the new constraints by the convex and concave envelopes (Al-Khayyal and Falk, 1983) and proved the dominance of their results using convexification and disjunctive programming.

Furthermore, Alfaki and Haugland (2013b) introduced a model called the *TP-formulation* consisting of the proportion variables corresponding to the terminals and flow variables along with the arcs between sources and pools. They claimed that combining these two proportions (sources and terminals) leads to a new model referred to as the *STP-formulation* in which the full benefit is achieved. Boland et al. (2016) extended these approaches for the generalized pooling problem through *source-based* and *terminal-based* multi-commodity flow formulations. Grothey and McKinnon (2020) introduced the *QQ-formulation*, which only uses proportion variables to solve real-world instances in the animal feed mix industry.

It is worth mentioning that some studies have elaborated on the complexity of the pooling problem. While proving the NP-hardness of the pooling problem, Alfaki and Haugland (2013a) showed that the problem preserves NP-hardness even if there exists only one pool. Baltean-Lugojan and Misener (2018) demonstrated the strongly-polynomial solutions and the NP-hardness of the pooling problem by parameterizing the objective function concerning pool concentrations. Meanwhile, the problem remains NP-hard even by having only one quality constraint at each pool or when the number of sources and terminals are no more than two (Haugland, 2016), but there exists a pseudo-polynomial algorithm to solve the problem (Haugland and Hendrix, 2016). On the other hand, the pooling problem could be polynomial-time solvable if there exists a bounded number of sources (Haugland and Hendrix, 2016; Boland et al., 2017). Moreover, having only one source or one terminal makes the problem polynomially solvable since it can be formulated in the compact form as a linear program (Haugland, 2016).

As can be seen above, the pooling problem is NP-Hard in general, and challenging to solve in practice. This has motivated the researchers to develop relaxations and restrictions for the problem to obtain dual and primal bounds. The LP relaxations based on the McCormick envelopes (McCormick, 1976) have been widely used in the literature to solve the pooling problem (Foulds et al., 1992; Tawarmalani and Sahinidis, 2002; Alfaki and Haugland, 2013a; Boland et al., 2015; Dey et al., 2020). In addition, mixed-integer programming (MIP) models have been developed to generate high-quality bounds as well (Adhya et al., 1999; Tomasgard et al., 2007; Pham et al., 2009; Alfaki and Haugland, 2011; Dey and Gupte, 2015; Haugland and Hendrix, 2016; Gupte et al., 2017, 2019). Furthermore, Marandi et al. (2018) conducted a numerical evaluation on the standard pooling problem instances by applying the sum-of-squares hierarchy (Lasserre et al., 2017) via solving semi-definite programs to construct lower bounds. Although this method has promising results in small instances, the scale of larger instances remains an issue and higher levels of the hierarchy become computationally expensive.

Dey et al. (2020) proposed a new formulation for the pooling problem in which the bilinear constraints are replaced with rank-one constraints on the decomposed flow matrix variables related to a pool. This allowed the authors to develop new relaxations for the pooling problem where the rank-one constraint with side constraints is relaxed. For example, they proved that the convex hull of the set of nonnegative, rank-one matrices with bounded row (or column) sums and the overall sum is polyhedrally representable. This translates to a nice interpretation for the pooling problem in which the bounds on row (resp. column) sums can be treated as the bounds on the incoming (resp. outgoing) arcs to a pool and the overall bound can be seen as the bound of the overall flow on the pool. We note that investigating the convex hulls of rank-one matrices carries significant implications for optimization, finding relevance in various domains such as machine learning, data analysis, and signal processing (see Burer and Kılınç-Karzan (2017); Burer and Letchford (2009); Gupte et al. (2020); Li and Vittal (2017); Rahman and Mahajan (2019); Dey et al. (2020)). In this paper, we also follow this approach and prove that the convex hull of the set of nonnegative, rank-one matrices with bounded row sums, column sums, and the overall sum is second-order cone representable.

While there have been some notable advancements in the field, most research has focused on the standard pooling problem and has been limited to small to medium problem instances. Moreover, the state-of-the-art solutions do not perform well while the flow streams among the pools are allowed. However, multi-period network flow problems, such as the mining problem associated with large-scale data, can be formulated as a special case of the general pooling problem.

We aim to address the above challenges in our paper, which offers both theoretical and methodological contributions to the pooling problem literature. From the theoretical aspect, we prove that the convex hull of the set of nonnegative, rank-one matrices with bounded row sums, column sums, and the overall sum is second-order cone representable. Although the size of this representation is exponential, it helps us to develop new strong LP relaxations than the well-known PQ- and TP-relaxations, and valid inequalities based on the reformulation and linearization technique (RLT). From the methodological aspect, we focus on improving both the time and quality of the exact and relaxed models via bound tightening. To improve the lower and upper bounds on the capacities of the nodes and arcs, we use the optimization-based bound tightening (OBBT) technique. In addition, we develop a novel bound-tightening method that leverages the special structure of the mining instances, which are large-scale real-world problems that can be converted to generalized pooling problems. By implementing our method in a few simple steps, we can significantly improve the quality of the dual bounds and obtain the solution in a more reasonable time using the exact formulation.

The rest of this paper is organized as follows: In Section 2, we prove that the convex hull of the set of nonnegative, rank-one matrices with bounded row sums, column sums and overall sum is second-order cone representable. In addition, we develop polyhedral outer-approximations of this complicated convex hull and propose valid inequalities using RLT. These developments will be the basis of our analysis in the succeeding sections. In Section 3, we describe the pooling problem formally and provide multi-commodity flow formulations in detail. Then, we review different polyhedral and MIP-based relaxations for these formulations. In Section 4, we present new LP relaxations we have developed. Moreover, we discuss the OBBT technique and how we can use it for the pooling problem. We also provide the details our tailored bound-tightening method to improve the bounds on the capacity of the arcs and nodes of the time-indexed pooling problem which arises in the mining industry. We present the settings of our different experiments and their computational results in Section 5. Finally, we will have some concluding remarks in Section 6.

Notation: We denote the set of integers $1, \dots, n$ as $[n]$. We will use the notation $\cdot$ when a bound is relaxed. e_k is the k-th unit vector, e is the vector of ones.

2 Main Results

Let us define the following polyhedral set

$$\mathcal{T}(l,u,l',u',L,U) := \Big\{ X \in \mathbb{R}_+^{m\times n} : l_i \le \sum_{j=1}^{n} x_{ij} \le u_i,\ i \in [m],\ l'_j \le \sum_{i=1}^{m} x_{ij} \le u'_j,\ j \in [n], \quad L \le \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \le U \Big\}, \tag{1}$$

where $l,u \in \mathbb{R}_+^m$, $l',u' \in \mathbb{R}_+^n$ and $L,U \in \mathbb{R}_+$ such that $u \ge l$, $u' \ge l'$ and $U \ge L$. Without loss of generality, we assume that $u > 0$ and $u' > 0$ (otherwise, we can simplify the analysis by deleting the row i with $u_i = 0$ and column j with $u'_j = 0$). In this section, we will focus on the study of the nonconvex set

$$\tilde{\mathcal{T}}(l,u,l',u',L,U) := \left\{ X \in \mathcal{T}(l,u,l',u',L,U) :\ \mathrm{rank}(X) \le 1 \right\}. \tag{2}$$

This nonconvex set appears as a substructure in the pooling problem. As an illustration, consider pool 1 in Figure 1. Let $X \in \mathbb{R}_+^{2\times 2}$ represent the decomposed flow variables, that is, x_{ij} is the amount of flow originated at node i and terminated at node j; $i \in \{s_1, s_2\}$, $j \in \{t_1, t_2\}$. In this case, the sum of row i (resp. column j) entries of matrix X is the incoming flow to (resp. outgoing flow from) this pool from source i (resp. to terminal j). Similarly, the sum of overall entries of X is the total flow at the pool. Due to the special structure of the pooling problem, we require $\text{rank}(X) \leq 1$, as this guarantees that the outgoing flow from the pool will have identical specifications (see Dey et al. (2020) for details).

Remark 1. *The matrix X has a slightly different interpretation in the case of generalized pooling problem. See Figure 3 and related discussions.*

We study nonconvex set $\tilde{\mathcal{T}}$ in three steps: In the first step, we prove that its convex hull is second-order cone representable in Section 2.1. This is an improvement over Dey et al. (2020), which showed that the convex hull of the set of nonnegative, rank-one matrices with bounded row sums or column sums, and overall sum is polyhedrally representable (we call this set as *column-wise relaxation* or *row-wise relaxation*). Unfortunately, the size of our second-order cone representation is exponential in the size of the matrix dimensions. This has motivated us to find a compact-size outer-approximation of the convex hull. In the second step, we obtain such a polyhedral outer-approximation in Section 2.2, which is stronger than the intersection of column-wise and row-wise relaxations from Dey et al. (2020). In the third and final step, we use RLT to further strengthen the polyhedral outer-approximation obtained in the second step with the addition of valid inequalities in Section 2.3.

2.1 Second-Order Cone Representable Convex Hull

In this section, we will prove the following theorem:

Theorem 1. $\text{conv}(\tilde{\mathcal{T}}(l, u, l', u', L, U))$ *is second-order cone representable.*

Before proving Theorem 1, we remind the reader that it has been recently proven that the set $\text{conv}(\tilde{\mathcal{T}}(l, u, l', u', L, U))$ has a polynomial-size polyhedral representation when either row bounds or column bounds are relaxed (Dey et al., 2020).

Theorem 2 (Dey et al. (2020))**.** *We have the following extended formulations:*

- *The row-wise formulation is* $\text{conv}(\tilde{\mathcal{T}}(l, u, \cdot, \cdot, L, U)) = \left\{X \in \mathbb{R}_+^{m\times n} : \exists t \in \mathbb{R}_+^n : (3)\right\}$, *where*

$$l_i t_j \leq x_{ij} \leq u_i t_j,\ i \in [m], j \in [n],\ L t_j \leq \sum_{i=1}^{m} x_{ij} \leq U t_j,\ j \in [n],\ \sum_{j=1}^{n} t_j = 1. \tag{3}$$

- *The column-wise formulation is* $\text{conv}(\tilde{\mathcal{T}}(\cdot, \cdot, l', u', L, U)) = \left\{X \in \mathbb{R}_+^{m\times n} : \exists t' \in \mathbb{R}_+^m : (4)\right\}$, *where*

$$l'_j t'_i \leq x_{ij} \leq u'_j t'_i,\ i \in [m], j \in [n],\ L t'_i \leq \sum_{j=1}^{n} x_{ij} \leq U t'_i,\ i \in [m],\ \sum_{i=1}^{m} t'_i = 1. \tag{4}$$

We need the following lemma in the proof of Theorem 1.

Lemma 1. *Let X be an extreme point of $\tilde{\mathcal{T}}(l, u, l', u', L, U)$. Then,*

- $\#_{row} := \left|\{i \in [m] : l_i < \sum_{j=1}^{n} x_{ij} < u_i\}\right| \leq 1.$
- $\#_{col} := \left|\{j \in [n] : l'_j < \sum_{i=1}^{m} x_{ij} < u'_j\}\right| \leq 1.$

Proof. We only prove the first statement since the proof of the second statement is similar. Since $\text{rank}(X) \leq 1$ and $X \geq 0$, there exist two non-zero vectors $y \in \mathbb{R}_+^{n_1}$ and $z \in \mathbb{R}_+^{n_2}$ such that $X = yz^\top$. By contradiction, suppose that $\#_{row} > 1$. Without loss of generality, let us assume that $l_i < \sum_{j=1}^n x_{ij} < u_i$ for $i = 1, 2$, which implies that $y_1 > 0$ and $y_2 > 0$. Now, let us consider the following two points:

$$X^\pm = y^\pm z^\top \text{ where } y^\pm = y \pm \epsilon e_1 \mp \epsilon e_2.$$

We have some observations: Firstly, the sum of the entries of y, y^+ and y^- vectors is the same since $e^\top y = e^\top y^+ = e^\top y^-$. Secondly, the row sums (except the first two) of X, X^+, and X^- matrices are the same since $e_i^\top(yz^\top)e = e_i^\top(y^+z^\top)e = e_i^\top(y^-z^\top)e$ for $i \geq 3$. Thirdly, all the column sums of X, X^+ and X^- matrices are the same since $e^\top(yz^\top)e_j = e^\top(y^+z^\top)e_j = e^\top(y^-z^\top)e_j$ for $j \geq 1$. Fourthly, all the overall sums of X, X^+ and X^- matrices are the same since $e^\top(yz^\top)e = e^\top(y^+z^\top)e = e^\top(y^-z^\top)e$.

Now, since all the row sums except the first two and all the column sums are unchanged, and the row sum bounds are not tight for the first two rows, we can find small enough $\epsilon > 0$ such that both X^+ and X^- belong to $\tilde{\mathcal{T}}(l, u, l', u', L, U)$ for some small enough $\epsilon > 0$. Hence, since $X = \frac{1}{2}X^+ + \frac{1}{2}X^-$ cannot be an extreme point, we reach a contradiction to the fact that $\#_{row} > 1$. □

Now, we are finally ready to prove Theorem 1.

Proof of Theorem 1. Since $\tilde{\mathcal{T}}(l, u, l', u', L, U)$ is compact, $\text{conv}(\tilde{\mathcal{T}}(l, u, l', u', L, U))$ can be obtained as the convex hull of its extreme points.

Let us consider a set, denoted by $\mathcal{S}_{i,j}(\{b_{i'}\}_{i' \neq i}, \{b'_{j'}\}_{j' \neq j})$, in which all the row bounds and column bounds (except the i-th row and j-th column) are equal to $b_{i'} \in \{l_{i'}, u_{i'}\}$, $i' \neq i$ and $b'_j \in \{l'_{j'}, u'_{j'}\}$, $j' \neq j$. Assuming $\mathcal{S}_{i,j}(\{b_{i'}\}_{i' \neq i}, \{b'_{j'}\}_{j' \neq j}) \neq \emptyset$, let X belong to this set. Note that X is an extreme point of $\tilde{\mathcal{T}}(l, u, l', u', L, U)$ due to Lemma 1. Since $\text{rank}(X) \leq 1$ and $X \geq 0$, there exist two non-zero vectors $y \in \mathbb{R}_+^{n_1}$ and $z \in \mathbb{R}_+^{n_2}$ such that $X = yz^\top$.

Note that we have $y_{i'} \sum_{j=1}^n z_j = b_{i'}$, $i' \neq i$ and $z_{j'} \sum_{i=1}^m y_i = b'_{j'}$, $j' \neq j$. The rest of the proof involves considering three cases:

Case 1: Suppose that there exist $I \neq i$ and $J \neq j$ such that $b_I > 0$ and $b'_J > 0$. Then, we obtain the following relations:

$$y_{i'} = \frac{b_{i'}}{b_I} y_I, \; i' \neq i \quad \text{and} \quad z_{j'} = \frac{b'_{j'}}{b'_J} z_J, \; j' \neq j.$$

As a shorthand notation, we define

$$B := \sum_{i' \neq i} \frac{b_{i'}}{b_I}, \quad B' := \sum_{j' \neq j} \frac{b'_{j'}}{b'_J}.$$

Considering the i-th row, j-th column and overall bounds, we obtain the following set of equations in y and z,

$$\begin{aligned}
&y_I(z_j + B'z_J) = b_I, \; z_J(y_i + By_I) = b'_J \\
&y_i(z_j + B'z_J) \in [l_i, u_i], \; z_j(y_i + By_I) \in [l'_j, u'_j], \; (y_i + By_I)(z_j + B'z_J) \in [L, U] \\
&y_i, y_I, z_j, z_J \geq 0,
\end{aligned}$$

which can be translated to X variables as follows:

$$\begin{aligned}
&x_{ij}x_{IJ} = x_{iJ}x_{Ij} \\
&x_{Ij} + B'x_{IJ} = b_I,\ x_{iJ} + Bx_{IJ} = b'_J \\
&x_{ij} + B'x_{iJ} \in [l_i, u_i],\ x_{ij} + Bx_{Ij} \in [l'_j, u'_j],\ x_{ij} + B'x_{iJ} + Bx_{Ij} + BB'x_{IJ} \in [L, U] \\
&x_{ij}, x_{IJ}, x_{iJ}, x_{Ij} \geq 0.
\end{aligned} \tag{5}$$

The set defined by (5) is the intersection of a quadratic equation with a polytope, and its convex hull is known to be a second-order cone representable set (Santana and Dey, 2020), which we denote by $\mathcal{Q}_{ij}$. Hence, we conclude that the convex hull of $\mathcal{S}_{i,j}(\{b_{i'}\}_{i'\neq i}, \{b'_{j'}\}_{j'\neq j})$ is the following second-order cone representable set:

$$\left\{ X \in \mathbb{R}_+^{m\times n} : (x_{ij}, x_{IJ}, x_{iJ}, x_{Ij}) \in \mathcal{Q}_{ij}, \sum_{j'=1}^{n} x_{ij} = b_{i'}\ i' \in [m] \setminus \{i\},\ \sum_{i'=1}^{m} x_{i'j} = b'_{j'}\ j' \in [n] \setminus \{j\} \right\}.$$

Case 2: Suppose that $b_{i'} = 0$ for all $i' \neq i$, meaning that all rows of X (except possibly for the i-th one) are zero vectors. Then, we conclude that $\mathcal{S}_{i,j}(\{b_{i'}\}_{i'\neq i}, \{b'_{j'}\}_{j'\neq j})$ is the following polyhedral set:

$$\left\{ X \in \mathbb{R}_+^{m\times n} : l_i \leq \sum_{j'=1}^{n} x_{ij'} \leq u_i,\ l'_{j'} \leq x_{ij'} \leq u'_{j'}\ j' \in [n],\ x_{i'j'} = 0\ i' \in [m] \setminus \{i\}, j' \in [n] \right\}.$$

Case 3: Suppose that $b'_{j'} = 0$ for all $j' \neq j$, meaning that all columns of X (except possibly for the j-th one) are zero vectors. Then, $\mathcal{S}_{i,j}(\{b_{i'}\}_{i'\neq i}, \{b'_{j'}\}_{j'\neq j})$ is the following polyhedral set:

$$\left\{ X \in \mathbb{R}_+^{m\times n} : l'_j \leq \sum_{i'=1}^{m} x_{i'j} \leq u'_j,\ l_{i'} \leq x_{i'j} \leq u_{i'}\ i' \in [m],\ x_{i'j'} = 0\ i' \in [m], j' \in [n] \setminus \{j\}. \right\}.$$

In all cases, we conclude that the convex hull of $\mathcal{S}_{i,j}(\{b_{i'}\}_{i'\neq i}, \{b'_{j'}\}_{j'\neq j}$ is second-order cone representable. Finally, by using the relation

$$\text{conv}(\tilde{\mathcal{T}}(l, u, l', u', L, U)) = \text{conv}\left(\bigcup_{i\in[m], j\in[n]} \bigcup_{b_{i'}\in\{l_{i'}, u_{i'}\}, b'_{j'}\in\{l'_{j'}, u'_{j'}\}} \text{conv}\big(\mathcal{S}_{i,j}(\{b_{i'}\}_{i'\neq i}, \{b'_{j'}\}_{j'\neq j})\big) \right),$$

and utilizing the fact that the convex hull of the union of compact second-order cone representable sets is again second-order cone representable (Ben-Tal and Nemirovski, 2001), we prove the statement of the theorem. □

2.2 Polyhedral Outer-approximations

We proved that $\text{conv}(\tilde{\mathcal{T}}(l, u, l', u', L, U))$ is second-order cone representable in Theorem 1. However, its exact representation might be quite large. Instead, we will develop some outer approximations of that set in this section.

2.2.1 A Straightforward Polyhedral Outer-approximation

A straightforward outer-approximation can be obtained using Theorem 2 as follows:

$$\mathcal{T}^1(l,u,l',u',L,U) := \operatorname{conv}(\tilde{\mathcal{T}}(l,u,\cdot,\cdot,L,U)) \cap \operatorname{conv}(\tilde{\mathcal{T}}(\cdot,\cdot,l',u',L,U)). \tag{6}$$

Clearly, the following relation holds,

$$\operatorname{conv}(\tilde{\mathcal{T}}(l,u,l',u',L,U)) \subseteq \mathcal{T}^1(l,u,l',u',L,U)$$

The set $\mathcal{T}^1(l,u,l',u',L,U)$ is the intersection of the row-wise and column-wise extended formulations derived in the previous section. Here, the variable t_j (t'_i) represents the ratio of the column sum j (row sum i) to the overall sum. We note that, due to the rank condition, t_j (t'_i) also represents the ratio of the entry x_{ij} to the row sum i (column sum j).

We now present some results related to the set $\mathcal{T}^1(l,u,l',u',L,U)$. To help us with the illustration, let us define the set

$$\tilde{\mathcal{T}}^1(l,u,l',u',L,U) := \Bigg\{ X \in \mathbb{R}_+^{m\times n} : \exists t \in \mathbb{R}_+^n, t' \in \mathbb{R}_+^m : \\ (3)-(4), \\ \operatorname{rank}(X) \le 1, \\ x_{ij} = t_j \sum_{j'=1}^{n} x_{ij'} = t'_i \sum_{i'=1}^{m} x_{i'j},\ i \in [m], j \in [n] \Bigg\}.$$

Proposition 1. *$\mathcal{T}(l,u,l',u',L,U) \supseteq \mathcal{T}^1(l,u,l',u',L,U)$.*

Proof. Let $X \in \mathcal{T}^1(l,u,l',u',L,U)$ and t,t' satisfy the constraints in the description of equation (6).

Summing each side of the inequality $Lt_j \le \sum_{i=1}^m x_{ij} \le Ut_j$ over j and using $\sum_{j=1}^n t_j = 1$ yield $L \le \sum_{j=1}^n \sum_{i=1}^m x_{ij} \le U$.

Summing each side of the inequality $l_i t_j \le x_{ij} \le u_i t_j$ over j and using $\sum_{j=1}^n t_j = 1$ yield $l_i \le \sum_{j=1}^n x_{ij} \le u_i$ for each $i \in [m]$.

Summing each side of the inequality $Lt'_i \le \sum_{j=1}^n x_{ij} \le Ut'_i$ over i and using $\sum_{i=1}^m t'_i = 1$ yield $L \le \sum_{i=1}^m \sum_{j=1}^n x_{ij} \le U$.

Summing each side of the inequality $l'_j t'_i \le x_{ij} \le u'_j t'_i$ over i and using $\sum_{i=1}^m t'_i = 1$ yield $l'_j \le \sum_{i=1}^m x_{ij} \le u'_j$ for each $j \in [n]$.

Hence, we conclude that $X \in \mathcal{T}(l,u,l',u',L,U)$. □

Proposition 2. *$\tilde{\mathcal{T}}(l,u,l',u',L,U) = \tilde{\mathcal{T}}^1(l,u,l',u',L,U)$.*

Proof. $\tilde{\mathcal{T}}(l,u,l',u',L,U) \supseteq \tilde{\mathcal{T}}^1(l,u,l',u',L,U)$: It follows directly from Proposition 1.

$\tilde{\mathcal{T}}(l,u,l',u',L,U) \subseteq \tilde{\mathcal{T}}^1(l,u,l',u',L,U)$: Let $X \in \tilde{\mathcal{T}}(l,u,l',u',L,U)$. If $\operatorname{rank}(X) = 0$, meaning that $X = 0$, then we can simply set $t_1 = 1$ and $t'_1 = 1$. Since all the lower bounds have to be zero in this case (otherwise, $X = 0$ would not have been feasible), it is trivial to see that $X \in \tilde{\mathcal{T}}^1(l,u,l',u',L,U)$.

On the other hand, if $\operatorname{rank}(X) = 1$, then we set

$$t_j := \frac{x_{ij}}{\sum_{j'=1}^n x_{ij'}} \quad \text{and} \quad t'_i := \frac{x_{ij}}{\sum_{i'=1}^m x_{i'j}} \qquad i \in [m], j \in [n].$$

Observe that these definitions are well-defined since $\operatorname{rank}(X) = 1$. We trivially obtain $\sum_{j=1}^{n} t_j = 1$ and $\sum_{i=1}^{m} t'_i = 1$.

Multiplying each side of the inequality $L \le \sum_{i=1}^{m}\sum_{j'=1}^{n} x_{ij'} \le U$ by t_j and replacing $\sum_{j'=1}^{n} x_{ij'} t_j$ by x_{ij} yield $Lt_j \le \sum_{i=1}^{m} x_{ij} \le Ut_j$ for each $j \in [n]$.

Multiplying each side of the inequality $l_i \le \sum_{j'=1}^{n} x_{ij'} \le u_i$ by t_j and replacing $\sum_{j'=1}^{n} x_{ij'} t_j$ by x_{ij} yield $l_i t_j \le x_{ij} \le u_i t_j$ for each $i \in [m]$, $j \in [n]$.

Multiplying each side of the inequality $L \le \sum_{j=1}^{n}\sum_{i'=1}^{m} x_{i'j} \le U$ by t'_i and replacing $\sum_{i'=1}^{m} x_{i'j} t'_i$ by x_{ij} yield $Lt'_i \le \sum_{j=1}^{n} x_{ij} \le Ut'_i$ for each $i \in [m]$.

Multiplying each side of the inequality $l'_j \le \sum_{i'=1}^{m} x_{i'j} \le u'_j$ by t'_i and replacing $\sum_{i'=1}^{m} x_{ij'} t'_i$ by x_{ij} yield $l'_j t'_i \le x_{ij} \le u'_j t'_i$ for each $i \in [m]$, $j \in [n]$.

Hence, we conclude that $X \in \tilde{\mathcal{T}}^1(l, u, l', u', L, U)$. □

2.2.2 A Stronger Polyhedral Outer-approximation

Since the convex relaxation $\mathcal{T}^1(l, u, l', u', L, U)$ is the intersection of the row-wise and column-wise extended formulations, it is natural to think of another extended formulation that considers rows and columns simultaneously. We now propose a stronger polyhedral outer approximation.

Let us define $\mathcal{T}^2(l, u, l', u', L, U) := \{X \in \mathbb{R}_+^{m\times n} : \exists R \in \mathbb{R}_+^{m\times n} : (7)\}$ where

$$\begin{aligned} &l_i \sum_{i'=1}^{m} r_{i'j} \le x_{ij} \le u_i \sum_{i'=1}^{m} r_{i'j},\ i \in [m], j \in [n],\ Lr_{ij} \le x_{ij} \le Ur_{ij},\ i \in [m], j \in [n], \\ &l'_j \sum_{j'=1}^{n} r_{ij'} \le x_{ij} \le u'_j \sum_{j'=1}^{n} r_{ij'},\ i \in [m], j \in [n],\ \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij} = 1, \end{aligned} \tag{7}$$

and

$$\tilde{\mathcal{T}}^2(l, u, l', u', L, U) := \Bigg\{X \in \mathbb{R}_+^{m\times n} : \exists R \in \mathbb{R}_+^{m\times n} : (7),\ \operatorname{rank}(X) \le 1, \\ x_{ij} = r_{ij} \sum_{i'=1}^{m}\sum_{j'=1}^{n} x_{i'j'},\ i \in [m], j \in [n]\Bigg\} \tag{8}$$

The variable r_{ij} represents the ratio of the entry x_{ij} to the overall sum. Intuitively, the relationships between the r variables appeared in (7), and the t, t' variables appeared in (3)–(4) are given as

$$r_{ij} = t'_i t_j,\ t'_i = \sum_{j'=1}^{n} r_{ij'} \text{ and } t_j = \sum_{i'=1}^{n} r_{i'j}.$$

Now, we will compare the relaxed extended formulations $T^1(l, u, l', u', L, U)$ and $T^2(l, u, l', u', L, U)$:

Proposition 3. *$\mathcal{T}^1(l, u, l', u', L, U) \supseteq \mathcal{T}^2(l, u, l', u', L, U)$.*

Proof. Let $X \in \mathcal{T}^2(l, u, l', u', L, U)$ and R satisfy the constraints in equation (7). Set

$$t_j := \sum_{i'=1}^{m} r_{i'j} \quad \text{and} \quad t'_i := \sum_{k'=1}^{m} r_{ij'} \qquad i \in [m], j \in [n].$$

By construction, we have

$$\sum_{j=1}^{n} t_j = 1 \quad \text{and} \quad \sum_{i=1}^{m} t'_i = 1.$$

Also,

$$l_i \sum_{i'=1}^{m} r_{i'j} \le x_{ij} \le u_i \sum_{i'=1}^{m} r_{i'j} \implies l_i t_j \le x_{ij} \le u_i t_j,$$

$$l'_i \sum_{k'=1}^{m} r_{ij'} \le x_{ij} \le u'_i \sum_{k'=1}^{m} r_{ij'} \implies l'_i t'_i \le x_{ij} \le u'_i t'_i.$$

Consider the inequality $Lr_{ij} \le x_{ij} \le Ur_{ij}$;
Summing each side of it over i and using $t_j = \sum_{i=1}^{m} r_{ij}$ yield $Lt_j \le \sum_{i=1}^{m} x_{ij} \le Ut_j, \quad \forall j \in [n]$.
Summing each side of it over j and using $t'_i = \sum_{k=1}^{m} r_{ij}$ yield $Lt'_i \le \sum_{j=1}^{n} x_{ij} \le Ut'_i, \quad \forall i \in [m]$.
Hence, we conclude that $X \in \mathcal{T}^1(l, u, l', u', L, U)$. □

Proposition 4. $\tilde{\mathcal{T}}(l, u, l', u', L, U) = \tilde{\mathcal{T}}^2(l, u, l', u', L, U)$.

Proof. $\tilde{\mathcal{T}}(l, u, l', u', L, U) \supseteq \tilde{\mathcal{T}}^2(l, u, l', u', L, U)$: It follows directly from Propositions 1 and 3.

$\tilde{\mathcal{T}}(l, u, l', u', L, U) \subseteq \tilde{\mathcal{T}}^2(l, u, l', u', L, U)$: Let $X \in \tilde{\mathcal{T}}(l, u, l', u', L, U)$. If $\mathrm{rank}(X) = 0$, meaning that $X = 0$, then we can simply set $r_{11} = 1$. Since all the lower bounds have to be zero in this case (otherwise, $X = 0$ would not have been feasible), it is trivial to see that $X \in \tilde{\mathcal{T}}^2(l, u, l', u', L, U)$. On the other hand, if $\mathrm{rank}(X) = 1$, then we set

$$r_{ij} := \frac{x_{ij}}{\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}} \qquad i \in [m], j \in [n].$$

We trivially obtain $\sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} = 1$. Since $\mathrm{rank}(X) = 1$, we also have

$$\sum_{i'=1}^{m} r_{i'j} = \frac{\sum_{i'=1}^{m} x_{i'j}}{\sum_{j'=1}^{n} \sum_{i'=1}^{m} x_{i'j'}} = \frac{x_{ij}}{\sum_{j'=1}^{n} x_{ij'}} \quad \text{and} \quad \sum_{j'=1}^{n} r_{ij'} = \frac{\sum_{j'=1}^{n} x_{ij'}}{\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}} = \frac{x_{ij}}{\sum_{i'=1}^{m} x_{i'j}},$$

for each $i \in [m], j \in [n]$.

Multiplying each side of the inequality $L \le \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'} \le U$ by r_{ij} and get x_{ij} instead of $\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'} r_{ij}$ yield $Lr_{ij} \le x_{ij} \le Ur_{ij}$ for each $i \in [m], j \in [n]$.

Multiplying each side of the inequality $l_i \le \sum_{j'=1}^{n} x_{ij'} \le u_i$ by $\sum_{i'=1}^{m} r_{i'j}$ and get x_{ij} instead of $\sum_{j'=1}^{n} x_{ij'} \sum_{i'=1}^{m} r_{i'j}$ yield $l_i \sum_{i'=1}^{m} r_{i'j} \le x_{ij} \le u_i \sum_{i'=1}^{m} r_{i'j}$ for each $i \in [m]$.

Multiplying each side of the inequality $l'_j \le \sum_{j'=1}^{n} x_{i'j} \le u'_j$ by $\sum_{j'=1}^{n} r_{ij'}$ and get x_{ij} instead of $\sum_{i'=1}^{m} x_{i'j} \sum_{j'=1}^{n} r_{ij'}$ yield $l'_j \sum_{i'=1}^{m} r_{ij'} \le x_{ij} \le u'_j \sum_{i'=1}^{m} r_{ij'}$ for each $j \in [n]$.

Hence, we conclude that $X \in \tilde{\mathcal{T}}^2(l, u, l', u', L, U)$. □

We conclude that both the sets $\mathcal{T}^1(l, u, l', u', L, U)$ and $\mathcal{T}^2(l, u, l', u', L, U)$ are outer approximations for $\mathrm{conv}(\mathcal{T}(l, u, l', u', L, U))$, but $\mathcal{T}^2(l, u, l', u', L, U)$ yields a stronger relaxation than $\mathcal{T}^1(l, u, l', u', L, U)$. On the other hand, the extended formulation $\mathcal{T}^1(l, u, l', u', L, U)$ requires $m+n$ many additional variables while we need mn many extra variables for $\mathcal{T}^2(l, u, l', u', L, U)$.

2.3 Valid Inequalities Obtained by RLT

In this section, we strengthen the polyhedral outer-approximation of $\mathrm{conv}(\tilde{\mathcal{T}}(l, u, l', u', L, U))$ obtained in the previous section by using RLT. Assume that $L > 0$ and let us define the following set of inequalities

$$l'_j/U \le t_j \le u'_j/L \qquad j \in [n] \tag{9a}$$

$$l_i/U \le t'_i \le u_i/L \qquad i \in [m] \tag{9b}$$

$$l_i t_j \le u'_j t'_i \qquad i \in [m], j \in [n] \tag{9c}$$

$$l'_j t'_i \le u_i t_j \qquad i \in [m], j \in [n], \tag{9d}$$

and consider the set

$$\tilde{\mathcal{R}}(l, u, l', u', L, U) := \Big\{ X \in \mathbb{R}_+^{m\times n} : \exists t \in \mathbb{R}_+^n, t' \in \mathbb{R}_+^m, R \in \mathbb{R}_+^{m\times n} : \\ (9), \\ \sum_{j=1}^{n} t_j = 1, \\ \sum_{i=1}^{m} t'_i = 1, \\ r_{ij} = t'_i t_j,\ i \in [m], j \in [n], \\ x_{ij} = r_{ij} \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'},\ i \in [m], j \in [n] \Big\}.$$

Proposition 5. *We have* $\tilde{\mathcal{R}}(l, u, l', u', L, U) \supseteq \tilde{\mathcal{T}}(l, u, l', u', L, U)$*.*

Proof. Let $X \in \tilde{\mathcal{T}}_{m,n}$. If $\operatorname{rank}(X) = 0$, meaning that $X = 0$, then we can simply set $t_1 = 1$ and $t'_1 = 1$, and thus $r_{11} = 1$. In this case, all the lower bounds have to be zero (otherwise, $X = 0$ would not have been feasible), and it is trivial to see that $X \in \tilde{\mathcal{R}}(l, u, l', u', L, U)$. On the other hand, if $\operatorname{rank}(X) = 1$, then we set $r_{ij} := \frac{x_{ij}}{\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}}$ for $i \in [m]$ and for $j \in [n]$, $t'_i := \sum_{j'=1}^{n} r_{ij'}$ for $i \in [m]$ and $t_j := \sum_{i'=1}^{m} r_{i'j}$ for $j \in [n]$. We trivially obtain $\sum_{j=1}^{n} t_j = 1$, $\sum_{i=1}^{m} t'_i = 1$, $r_{ij} = t'_i t_j$ and $x_{ij} = r_{ij} \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$ for $i \in [m]$, $j \in [n]$.

Dividing each side of the inequality $l_i \le \sum_{j=1}^{n} x_{ij} \le u_i$ by $\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$ and using the definition of t'_i as above yield the inequality

$$\frac{l_i}{\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}} \le t'_i \le \frac{u_i}{\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}} \implies l_i/U \le t'_i \le u_i/L, \quad i \in [m].$$

Dividing each side of the inequality $l'_j \le \sum_{j=1}^{n} x_{ij} \le u'_j$ by $\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$ and using the definition of t_j as above yield the inequality

$$\frac{l'_j}{\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}} \le t_j \le \frac{u'_j}{\sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}} \implies l'_j/U \le t_j \le u'_j/L, \quad j \in [n].$$

The above derivation also shows that we have

$$\frac{l_i}{t'_i} \le \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'} \le \frac{u_i}{t'_i},\ i \in [m] \quad \text{and} \quad \frac{l'_j}{t_j} \le \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'} \le \frac{u'_j}{t_j},\ j \in [n],$$

from which we deduce that $l_i t_j \le u'_j t'_i$ and $l'_j t'_i \le u_i t_j$ for $i \in [m]$, $j \in [n]$.

Hence, we prove that $X \in \tilde{\mathcal{R}}(l, u, l', u', L, U)$. □

We will obtain valid inequalities for the nonconvex set $\tilde{\mathcal{R}}(l, u, l', u', L, U)$ using the RLT approach. Note that any such valid inequality is also valid for the set of our interest, $\tilde{\mathcal{T}}(l, u, l', u', L, U)$ due to Proposition 5. We will apply the following procedure to obtain such inequalities:

(i) Transform the inequalities (9) into the form less-than-or-equal type with 0 right hand side.

(ii) Multiply the resulting inequalities to obtain bilinear expressions in t and t', and convert them into inequalities in r:

(a) Replace the term $t'_i t_j$ with r_{ij}.
(b) Replace the term t'_i with $\sum_{j'=1}^{n} r_{ij'}$.
(c) Replace the term t_j with $\sum_{i'=1}^{m} r_{i'j}$.

(iii) Obtain inequalities in x variables by multiplying $\sum_{i'=1}^{m} \sum_{j=1}^{n} x_{i'j'}$ with the inequalities obtain in the previous step in r variables:

(a) Replace the term $r_{ij} \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$ with x_{ij}.
(b) Replace the term $\sum_{i'=1}^{m} r_{i'j} \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$ with $\sum_{i'=1}^{m} x_{i'j}$.
(c) Replace the term $\sum_{j'=1}^{n} r_{ij'} \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$ with $\sum_{j'=1}^{n} x_{ij'}$.

Multiplying (9a) **and** (9b): These are precisely the McCormick envelopes applied to $r_{ij} = t'_i t_j$. These linear inequalities in r variables are given as follows:

$$r_{ij} \geq (l_i/U) \sum_{i'=1}^{m} r_{i'j} + (l'_j/U) \sum_{j'=1}^{n} r_{ij'} - (l_i l'_j)/(U^2)$$
$$r_{ij} \leq (l_i/U) \sum_{i'=1}^{m} r_{i'j} + (u'_j/L) \sum_{j'=1}^{n} r_{ij'} - (l_i u'_j)/(UL)$$
$$r_{ij} \leq (u_i/L) \sum_{i'=1}^{m} r_{i'j} + (l'_j/U) \sum_{j'=1}^{n} r_{ij'} - (u_i l'_j)/(UL)$$
$$r_{ij} \geq (u_i/L) \sum_{i'=1}^{m} r_{i'j} + (u'_j/L) \sum_{j'=1}^{n} r_{ij'} - (u_i u'_j)/(L^2)$$

The linear inequalities in x variables are obtained as follows:

$$x_{ij} \geq (l_i/U) \sum_{i'=1}^{m} x_{i'j} + (l'_j/U) \sum_{j'=1}^{n} x_{ij'} - (l_i l'_j)/(U^2) \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$$
$$x_{ij} \leq (l_i/U) \sum_{i'=1}^{m} x_{i'j} + (u'_j/L) \sum_{j'=1}^{n} x_{ij'} - (l_i u'_j)/(UL) \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$$
$$x_{ij} \leq (u_i/L) \sum_{i'=1}^{m} x_{i'j} + (l'_j/U) \sum_{j'=1}^{n} x_{ij'} - (u_i l'_j)/(UL) \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$$
$$x_{ij} \geq (u_i/L) \sum_{i'=1}^{m} x_{i'j} + (u'_j/L) \sum_{j'=1}^{n} x_{ij'} - (u_i u'_j)/(L^2) \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$$

Multiplying (9c) **and** (9d) **for the same** (i, j) **pair**: The second-order cone representable inequalities in r variables are given as follows:

$$l_i u_i \left(\sum_{i'=1}^{m} r_{i'j} \right)^2 + l'_j u'_j \left(\sum_{j'=1}^{n} r_{ij'} \right)^2 \leq (l_i l'_j + u_i u'_j) r_{ij}$$

The inequalities in x variables, which are again second-order cone representable, are obtained as follows:

$$l_i u_i \left(\sum_{i'=1}^{m} x_{i'j} \right)^2 + l'_j u'_j \left(\sum_{j'=1}^{n} x_{ij'} \right)^2 \leq (l_i l'_j + u_i u'_j) x_{ij} \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$$

Multiplying (9a) **and** (9c) **for the same** (i, j) **pair**: We obtain two types of inequalities. The first type is second-order cone representable. These inequalities in r variables are as follows:

$$l_i \left(\sum_{i'=1}^{m} r_{i'j} \right)^2 \leq (l_i l'_j / U) \sum_{i'=1}^{m} r_{i'j} - (l'_j u'_j / U) \sum_{j'=1}^{n} r_{ij'} + u'_j r_{ij}.$$

The inequalities in x variables are obtained as follows:

$$l_i \left(\sum_{i'=1}^{m} x_{i'j} \right)^2 \leq \left[(l_i l'_j / U) \sum_{i'=1}^{m} x_{i'j} - (l'_j u'_j / U) \sum_{j'=1}^{n} x_{ij'} + u'_j x_{ij} \right] \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'}$$

The second type is a reverse-convex inequality

$$u_i t_j^2 \geq (u_i l'_j / U) \sum_{i'=1}^{m} r_{i'j} - (l'^2_j / U) \sum_{j'=1}^{n} r_{ij'} + l'_j r_{ij}.$$

We can over-approximate the left-hand side to obtain the following valid linear inequality in r variables:

$$u_i \left[(u'_j / L + l'_j / U) \sum_{i'=1}^{m} r_{i'j} - (u'_j l'_j)/(UL) \right] \geq (u_i l'_j / U) \sum_{i'=1}^{m} r_{i'j} - (l'^2_j / U) \sum_{j'=1}^{n} r_{ij'} + l'_j r_{ij}. \tag{10}$$

The corresponding inequalities in x variables are obtained as follows:

$$\begin{aligned} u_i \left[(u'_j / L + l'_j / U) \sum_{i'=1}^{m} x_{i'j} - (u'_j l'_j)/(UL) \sum_{i'=1}^{m} \sum_{j'=1}^{n} x_{i'j'} \right] \geq \\ (u_i l'_j / U) \sum_{i'=1}^{m} x_{i'j} - (l'^2_j / U) \sum_{j'=1}^{n} x_{ij'} + l'_j x_{ij}. \end{aligned} \tag{11}$$

We note that similar valid inequalities can be obtained if the inequalities (9a) and (9d), (9b) and (9d), or (9b) and (9d) are multiplied.

3 The Pooling Problem

In this section, we are going to describe the pooling problem formally, and present its the well-known and recently developed exact formulations and different types of relaxations.

3.1 Problem Definition and Notation

Let $G = (N, A)$ represent a graph with the node set N and the arc set A. Moreover, let S, I, T, and K denote the set of sources (inputs), intermediates (pools), terminals (outputs), and specifications respectively. Then, in the pooling problem, we have $N = S \cup I \cup T$. For the standard pooling problem , we have $A \subseteq (S \times (I \cup T)) \cup (I \cup T)$, while in the general pooling problem, we have $A \subseteq (S \times (I \cup T)) \cup (I \times (I \cup T))$. This definition clearly shows that in the generalized version, we may have flow streams among the pools. In this notation, S_i is the set of source nodes from which there is a path to node i and T_i is the set of terminal nodes to which there is a path from node i. The set of nodes to which there is an arc from node i and the set of nodes from which there is an arc to node i are denoted by N_i^+ and N_i^- respectively. In this notation, the unit cost of using arc (i, j) is shown by C_{ij} and the specification k of source s by λ_k^s. We may also have some lower and upper bounds for the desired specification k at terminal t denoted by $[\underline{\mu}_k^t, \bar{\mu}_k^t]$, the capacity of node i denoted by $[L_i, U_i]$, and capacity of arc (i, j) denoted by $[l_{ij}, u_{ij}]$. A summary of all the notation, which we have mostly adapted from Dey et al. (2020), can be found in Table 1.

Table 1: Notations of the source-based rank formulation

Indices	s	source (or input), $s = 1, \dots, S$
	i	intermediate (or pool), $i = 1, \dots, I$
	t	terminal (or output), $t = 1, \dots, T$
	k	specification, $k = 1, \dots, K$
Sets	S_i	the set of source nodes from which there is a path to node i
	T_i	the set of terminal nodes to which there is a path from node i
	N_i^+	the set of nodes to which there is an arc from node i
	N_i^-	the set of nodes from which there is an arc from node i
Variables	f_{ij}	the amount of flow from node i to node j
	x_{ij}^s	the amount of flow on arc (i, j) originated at the source $s \in S_i$
	q_i^s	the fraction of flow at pool i originated at source s
Parameters	C_{ij}	cost of sending unit flow over arc (i, j)
	λ_k^s	the specification k of source s
	$[\underline{\mu}_k^t, \bar{\mu}_k^t]$	the desired interval for specification k of terminal t
	$[L_i, U_i]$	lower bound and upper bound of the capacity of node i
	$[l_{ij}, u_{ij}]$	lower bound and upper bound of the capacity of arc (i, j)

3.2 Source-Based Rank Formulation

In this section, we review the source-based multi-commodity flow formulation for the generalized pooling problem developed in Alfaki and Haugland (2013a). This formulation consists of the proportion variables corresponding to sources and the flow variables along with the arcs between pools and terminals. This formulation was later investigated and presented in Dey et al. (2020). These authors have convexified the nonconvex constraint in different ways. We will use their formulation and go over their proposed methods in the next sections.

3.2.1 Mathematical Model

In this section, we will review the *Source-Based* multi-commodity flow formulation. An explanation of the mathematical model and an introduction to different relaxations and restrictions will follow.

$$\min \sum_{i \in I} \sum_{s \in S_i} \sum_{j \in N_i^+} C_{si} x_{ij}^s - \sum_{i \in I} \sum_{j \in N_i^+} C_{ij} f_{ij} \tag{12}$$

$$
\begin{aligned}
\text{s.t. } & L_i \le \sum_{j \in N_i^-} f_{ji} \le U_i && \forall i \in I \cup T && (13)\\
& L_i \le \sum_{j \in N_i^+} f_{ij} \le U_i && \forall i \in S && (14)\\
& l_{ij} \le f_{ij} \le u_{ij} && \forall (i,j) \in A \cup \{(s,i) : i \in I, s \in S_i\} && (15)\\
& \sum_{j \in N_{si}^-} x_{ji}^s = \sum_{j \in N_i^+} x_{ij}^s && \forall i \in I, \forall s \in S_i && (16)\\
& \sum_{s \in S_i} x_{ij}^s = f_{ij} && \forall (i,j) \in A && (17)\\
& \sum_{j \in N_i^+} x_{ij}^s = f_{si} && \forall i \in I, \forall s \in S_i && (18)\\
& \underline{\mu}_k^t \sum_{j \in N_t^-} f_{jt} \le \sum_{j \in N_t^-} \sum_{s \in S_j} \lambda_k^s x_{jt}^s \le \bar{\mu}_k^t \sum_{j \in N_t^-} f_{jt} && \forall t \in T, \forall k \in K && (19)\\
& x_{ij}^s = q_i^s f_{ij} && \forall (i,j) \in A, \forall s \in S_i && (20)\\
& x_{ij}^s \ge 0 && \forall (i,j) \in A, \forall s \in S_i && (21)\\
& q_i^s \ge 0 && \forall i \in I, \forall s \in S_i. && (22)
\end{aligned}
$$

The objective function (12) minimizes the cost of sending the raw materials to the outputs through some pools. In this equation, the sum of x_{ij}^s variables can be replaced by f_{ij} and the objective will be as follows:

$$\min \sum_{(i,j) \in A} C_{ij} f_{ij}$$

In this case, we should consider the cost of purchasing raw materials as positive and the profit of selling outputs as negative. In addition, the cost of sending the raw material directly to the output is the difference between its cost and the revenue from selling it which could be either positive or negative.

Constraint (13) imposes bounds on the capacity of pools and terminals while in constraint (14), we have these bounds for the sources. In constraint (15), flows on different arcs are limited to be in an interval, and constraint (16) is the flow conservation, which guarantees that all the flows coming into pools go out of them. We define the set N_{si}^- used in equation (16) as

$$N_{si}^- = \{j \in N_i^- : j \notin S \setminus s \text{ and } s \in S_j\}.$$

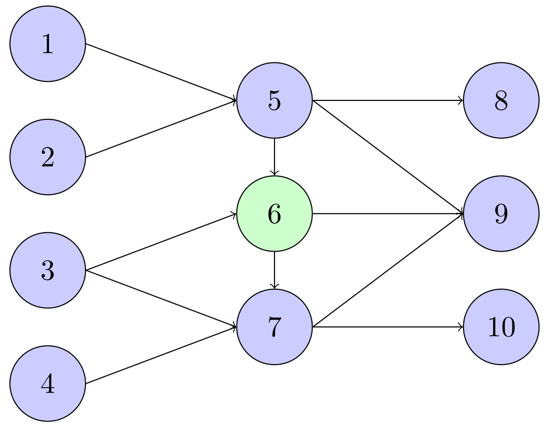

Figure 3: A sample generalized pooling problem instance.

Figure 3 shows a fictitious sample of a general pooling problem, which will be our running example in this section. Let us write constraint (16) for pool $i = 6$ and its source $s = 1$:

$$N^-_{1,6} = \{5\},\ N^+_6 = \{7, 9\} \implies x^1_{5,6} = x^1_{5,7} + x^1_{5,9}$$

In fact, this constraint ensures that the incoming flow to a pool from each of its source nodes equals the outgoing flow from it originated at the same source.

Equations (17) and (18) ensure the flow decomposition to be performed precisely while if the link (s, i) does not exist for $i \in I$ and $s \in S_i$ but we will call f_{si} as a *ghost flow* following Dey et al. (2020). Let us have a look at Figure 3 and try to write equation (18) for $i = 6$ and $s = 1$:

$$x^1_{6,7} + x^1_{6,9} = f_{1,6}$$

Constraint (19) is to meet the specification requirements at terminals. Constraint (20), which is the nonconvex bilinear constraint, calculates the fraction of flow at pool i originated at source s on each arc (i, j). Finally, we have the nonnegativity of the variables in constraints (21) and (22).

It is worth mentioning that the *Source-Based* formulation is equivalent to the *PQ*-formulation.

3.3 Polyhedral Relaxations

The proposed exact formulation of the pooling problem is nonconvex. Now, we will present LP relaxations of this model. Our starting point will be the work of Dey et al. (2020) in which the bilinear constraint in the *Source-Based* formulation (20) is rewritten as a set of rank restrictions on a matrix consisting of decomposed flow variables x^s_{ij} as follows:

$$\text{rank}\left([x^s_{ij}]_{(s,j)\in S_i \times N^+_i}\right) \leq 1 \qquad \forall i \in I. \tag{23}$$

As an example, consider pool $i = 6$ in Figure 3. It is easy to see that the following relation:

$$\begin{bmatrix} x^1_{6,7} & x^1_{6,9} \\ x^2_{6,7} & x^2_{6,9} \\ x^3_{6,7} & x^3_{6,9} \end{bmatrix} = \begin{bmatrix} q^1_6 \\ q^2_6 \\ q^3_6 \end{bmatrix} \times \begin{bmatrix} f_{6,7} & f_{6,9} \end{bmatrix}$$

This example demonstrates the logic of the rank-one constraint as the matrix on the left-hand side is the bilinear constraint can be written as the product of a column vector and a row vector.

The rank constraint (23) can be convexified in different ways. Below, we present some LP-based relaxations in detail.

3.3.1 Column-Wise Relaxation

Let us consider constraints (14), (15) (in which f_{ij} is substituted by its equivalent values from (18) and (17) respectively), and the bilinear constraint (20) (replaced with its equivalent *rank* constraint (23)) as a set. According to Theorem 2 and equation (3) we can define the *column-wise* relaxation for the *source-based* formulation for pool i as below:

$$\mathcal{F}_1^{\mathcal{S}(i)} := \left\{ [x_{ij}^s]_{(s,j)\in S_i\times N_i^+} \in \operatorname{conv}(\tilde{\mathcal{T}}(l_i, u_i, \cdot, \cdot, L_i, U_i)) \right\}. \tag{24}$$

This relaxation restricts the column-sum of the decomposed flow variables' matrices for all $i \in I$ and is equivalent to the McCormick relaxation of the PQ-formulation (Dey et al., 2020).

As an illustration, let us consider $i = 6$ in Figure 3 as a pool for which we will implement the column-wise extended relaxation. The associated sets for this pool are as follows:

$$S_6 = \{1, 2, 3\}, N_6^+ = \{7, 9\} \tag{25}$$

Then, in the following matrix, we have a row for each element of S_6 and a column for each element in N_6^+. The bound of each column is the bound of an outgoing arc from the corresponding pool and the overall bound is the pool bound. This instance shows how we impose column-sum bounds on the matrix of decomposed flow variables for each pool.

$$\begin{array}{cccc} & u_{6,7} & u_{6,9} & U_6 \\ \left[x_{ij}^s\right]_{(s,j)} = & \multicolumn{2}{c}{\begin{pmatrix} x_{6,7}^1 & x_{6,9}^1 \\ x_{6,7}^2 & x_{6,9}^2 \\ x_{6,7}^3 & x_{6,9}^3 \end{pmatrix}} & \\ L_6 & l_{6,7} & l_{6,9} & \end{array}$$

3.3.2 Row-Wise Relaxation

Analogous to the *column-wise* relaxation, the *row-wise* relaxation can be defined based on Theorem 2 and equation (4). This relaxation, which restricts the row-sum of the decomposed flow variables' matrices for all $i \in I$, is defined as follows:

$$\mathcal{F}_2^{\mathcal{S}(i)} := \left\{ [x_{ij}^s]_{(s,j)\in S_i\times N_i^+} \in \operatorname{conv}(\tilde{\mathcal{T}}(\cdot, \cdot, l_i', u_i', L_i, U_i)) \right\}. \tag{26}$$

Considering pool 6 from Figure 3 and the associated sets defined in equations (25), the following matrix has a row for each element in S_6 and a column for each element of N_6^+. The bounds imposed on the summation of each row are the bounds of incoming arcs (including the *ghost flows*) to pool $i = 6$ and the overall bound is the pool's capacity bounds.

$$\left[x_{ij}^s\right]_{(s,j)} = \begin{array}{c} \\ l_{1,6} \\ l_{2,6} \\ l_{3,6} \\ L_6 \end{array} \begin{array}{c} \\ \begin{pmatrix} x_{6,7}^1 & x_{6,9}^1 \\ x_{6,7}^2 & x_{6,9}^2 \\ x_{6,7}^3 & x_{6,9}^3 \end{pmatrix} \\ \\ \end{array} \begin{array}{c} U_6 \\ u_{1,6} \\ u_{2,6} \\ u_{3,6} \\ \\ \end{array}$$

This matrix shows how we impose row bounds on the matrix of the decomposed flow variables.

3.3.3 Intersection of Row-Wise and Column-Wise Relaxations

We can use the intersection of the row-wise and column-wise relaxations as a new method to relax the nonlinear constraint of the pooling problem (equation (6)). As we saw in Section 2.2.1, this relaxation is at least as good as both of the previous ones but increases the scale of the problem. We use the extended formulations of the row-wise and column-wise relaxations to implement it and define it for all $i \in I$ as follows:

$$\mathcal{F}_3^{\mathcal{S}(i)} = \mathcal{F}_1^{\mathcal{S}(i)} \cap \mathcal{F}_2^{\mathcal{S}(i)}$$

3.4 Mixed-Integer Programming Approximations

One of the other ways to deal with the bilinear constraint (20), is to utilize discretization methods. Gupte et al. (2017) have classified the discretization methods proposed for the pooling problem into two different categories: i) forcing some variables to take certain prespecified values from their domain which applies to each bilinear program, and ii) discretizing the amount of flow at each pool which was proposed by Dey and Gupte (2015) for the first time and results in a "network flow MILP restriction" by exploiting the pooling problem's structure. Both of these strategies convert the pooling problem to an MILP and give an approximation of the problem. In this section, we will use the discretization methods described in Dey et al. (2020), which focus on the first strategy. We use them in the *Source-Based* formulation to obtain inner and outer approximations of the pooling problem.

In this section, we try to find an outer approximation (relaxation) by discretizing the proportion variables q as follows:

$$q_j = \sum_{h=1}^{H} 2^{-h} z_{jh} + \gamma_j,$$

where $H \in \mathbb{Z}_{++}$ is the level of discretization, z_{ih} are binary variables, and γ_i is a continuous non-negative variable upper-bounded by 2^{-H}. Now we define x^s_{ij} as follows:

$$x^s_{ij} = \left(\sum_{j' \in N_i^+} x^s_{ij'}\right)\left(\sum_{h=1}^{H} 2^{-h} z_{jh} + \gamma_j\right) \qquad \forall i \in I, \forall s \in S_i, \forall j \in N_i^+.$$

Let $\alpha_{sjh} := (\sum_{j' \in N_i^+} x^s_{ij'}) z_{jh}$ and $\beta_{ij} := (\sum_{j' \in N_i^+} x^s_{ij'})\gamma_j$, then, by using the McCormick envelopes, we obtain the following outer-approximation for all $i \in I,\ s \in S_i,$ and $j \in N_i^+$:

$$\bar{\mathcal{D}}^{row}_{(|S_i|,|N_i^+|,H)}([l_{si}]_s, [u_{si}]_s) :=$$
$$\{x \in \mathbb{R}_+^{S_i \times N_i^+} | (\alpha, \beta, \gamma, z) \in \mathbb{R}^{S_i \times N_i^+ \times H} \times \mathbb{R}^{S_i \times N_i^+} \times \mathbb{R}^{N_i^+} \times \{0,1\}^{N_i^+ \times H} :$$

$$l_{si} z_{jh} \le \alpha_{sjh} \le u_{si} z_{jh} \qquad \forall j \in N_i^+, \forall h \in [H], \tag{27}$$

$$u_{si} z_{jh} + \sum_{j' \in N_i^+} x^s_{ij'} - u_{si} \le l_{si} z_{jh} + \sum_{j' \in N_i^+} x^s_{ij'} - l_{si} \qquad \forall j \in N_i^+, \forall h \in [H], \tag{28}$$

$$l_{si}\gamma_j \le \beta_{sj} \le u_{si}\gamma_j \qquad \forall j \in N_i^+, \tag{29}$$

$$\left.\begin{aligned} & u_{si}\gamma_j + 2^{-H}(\sum_{j' \in N_i^+} x^s_{ij'} - u_{si}) \le \beta_{sj} \\ & \beta_{sj} \le l_{si}\gamma_j + 2^{-H}(\sum_{j' \in N_i^+} x^s_{ij'} - l_{si}) \end{aligned}\right. \qquad \forall j \in N_i^+, \tag{30}$$

$$l_{si} \leq \sum_{j \in N_i^+} x_{ij}^s \leq u_{si} \tag{31}$$

$$x_{ij}^s = \sum_{h=1}^{H} 2^{-h} \alpha_{sjh} + \beta_{sj} \qquad \forall j \in N_i^+\}. \tag{32}$$

Dey et al. (2020) showed that $\bar{\mathcal{D}}^{row}_{(|S_i|,|N_i^+|,H)}$ is a relaxation of the source-based rank formulation.

We can analogously define the outer approximation denoted as $\bar{\mathcal{D}}^{col}_{(|S_i|,|N_I^+|,H)}([l_{ij}]_j, [u_{ij}]_j), \forall i \in I$ by restricting the sum of each column in the decomposed flow variable matrices and as well.

Analogous to the *Source-Based* formulation, we have the *Terminal-Based* formulation consisting of the proportion variables corresponding to the terminals and the flow variables along with the arcs between sources and pools. This model was first introduced in Alfaki and Haugland (2013a) as the *TP*-formulation and was later utilized in Dey et al. (2020).

4 Solution Approach

In this section, we develop a new LP relaxation that considers imposing bounds on the row-sum and the column-sum of the decomposed flow variable matrices simultaneously (Section 4.1.1). We also provided the technical details of obtaining new valid inequalities by the RLT in Section 2.3, which we will use in the computations. In addition, we discuss how to utilize the OBBT technique to improve the bounds of the arcs and nodes of the generalized pooling problem instances of the literature (Section 4.3.1). Moreover, we propose a simple and computationally cheap bound tightening method to improve the bounds of the mining problem as a special case of the generalized pooling problem with the "time-indexed" feature (Section 4.3.2). It is important to note that this section represents a novel application of the concepts and techniques introduced in Section 2 to the pooling problem, showcasing the versatility and effectiveness of our approach.

4.1 New Relaxations and Valid Inequalities

We have discussed the well-known relaxations of the literature in Section 3. In this section, we will present the relaxations and valid inequalities we have developed to improve the quality of the dual bounds.

4.1.1 New Linear Programming Relaxations

We have reviewed the row-wise and column-wise extended relaxations for the *Source-Based* and *Terminal-Based* multi-commodity flow formulations presented by Dey et al. (2020) in Section 3.3. Also, we discussed that a stronger relaxation can be obtained by intersecting these two aforementioned relaxations, which may increase the size of the problem. In Section 2.2.2, we showed that to have an even stronger relaxation, we can consider imposing bounds on the row-sum and column-sum of a matrix consisting of the decomposed flow variables of each pool simultaneously. We call this relaxation *Row-Column* and define it as the following.

$$\mathcal{F}_4^{\mathcal{S}(i)} := \left\{ [x_{ij}^s]_{(s,j) \in S_i \times N_i^+} \in \mathcal{T}^2(l_i, u_i, l_i', u_i', L_i, U_i) \right\}. \tag{33}$$

This relaxation convexifies the set $\tilde{\mathcal{T}}^2(l_i, u_i, l_i', u_i', L_i, U_i)$ defined in the equation (8) which can be considered as the intersection of the bilinear constraint (i.e., its equivalent *rank* constraint) and the capacity constraints that we have in the *Source-Based* formulation for each pool $i \in I$.

Proposition 1 shows the *row-column* relaxation is at least as good as the intersection of the row-wise and the column-wise relaxations. Therefore, we have the following in which the left-hand side relationship can be strict:

$$\mathcal{F}_4^{\mathcal{S}(i)} \subseteq \mathcal{F}_3^{\mathcal{S}(i)} := \mathcal{F}_1^{\mathcal{S}(i)} \cap \mathcal{F}_2^{\mathcal{S}(i)}$$

4.2 Valid Inequalities

In Section 2.3, we applied the RLT to derive new valid inequalities to strengthen the relaxations. Now, we will exemplify how these inequalities are adaptable to the context and the notations of the pooling problem. For brevity, we will use equations (10) and (11) below as examples although all the inequalities derived in Section 2.3 are applicable in principle.

- The inequalities (10) in r variables can be written as follows:

$$u_{si}\Big[(u_{ij}/L_i + l_{ij}/U_i)\sum_{s'\in S_i} r_{ij}^{s'} - (u_{ij}l_{ij})/(U_iL_i)\Big] \geq (u_{si}l_{ij}/U_i)\sum_{s'\in S_i} r_{ij}^{s'} - (l_{ij}^2/U_i)\sum_{j'\in N_i^+} r_{ij'}^{s} + l_{ij}r_{ij}^{s}$$

- The corresponding inequalities (11) in x variables are as follows:

$$u_{si}\Big[(u_{ij}/L_i + l_{ij}/U_i)\sum_{s'\in S_i} x_{ij}^{s'} - (u_{ij}l_{ij})/(U_iL_i)\sum_{s'\in S_i}\sum_{j'\in N_i^+} x_{ij'}^{s'}\Big] \geq (u_{si}l_{ij}/U_i)\sum_{s'\in S_i} x_{ij}^{s'} - (l_{ij}^2/U_i)\sum_{j'\in N_i^+} x_{ij'}^{s} + l_{ij}x_{ij}^{s}$$

Our experiments involve simultaneously adding valid inequalities both in x and r variables. Specifically, we denote the valid inequalities resulting from the multiplication of (9a) and (9b) as $\mathcal{V}_{ab}$, while the ones obtained by multiplying (9a) and (9c) are labeled as $\mathcal{V}_{ac}$. We have restricted ourselves to these families of inequalities since the other inequalities either have a negligible effect on the overall results or cause numerical issues. Detailed results of our experiments with the addition of valid inequalities can be found in Section 5.2.4.

4.3 Bound Tightening

4.3.1 Optimization-Based Bound Tightening

In global optimization, one of the valuable tools to reduce the variables' domain is to execute OBBT (Coffrin et al., 2015; Puranik and Sahinidis, 2017; Bynum et al., 2018). Let $\underline{z}$ and $\bar{z}$ represent the lower and upper bound of our multi-commodity flow problems, which can be obtained from a relaxation and any primal solution, respectively. Then to find the bounds of arcs and different nodes, we optimize the following objective functions over a linear programming relaxation of the original problem. We need to take into account that the original objective function should be between $\underline{z}$ and $\bar{z}$. Therefore, we add this as a constraint to the new problem.

- To find the lower bound (upper bound) of each arc $(i, j) \in A$, we minimize (maximize) the corresponding flow variable (f_{ij}).

- To generate a lower bound (upper bound) for each source node s, we minimize (maximize) the summation of outgoing flows from that node ($\sum_{i\in N_s^+} f_{si}$).
- For each pool i, to find a lower bound (upper bound), we minimize (maximize) the summation of incoming flows ($\sum_{j\in N_i^-} f_{ji}$) or the summation of outgoing flows from that node ($\sum_{j\in N_i^+} f_{ij}$).
- Finally, to improve the lower bound (upper bound) of each terminal node t, we minimize (maximize) the summation of all the incoming flows to that terminal ($\sum_{i\in N_t^-} f_{it}$).

4.3.2 Bound Tightening for the Time-indexed Pooling Problem

A special case of the generalized pooling problem which arises in the mining industry is investigated in Boland et al. (2015). In this case, the raw material (supply) with certain specifications comes into stockpile $p = \{1, \ldots, P\}$ at time $\tau \in \mathcal{T}_p^s$. On the other hand, demand for the final product with the desired specifications is placed at time $\tau \in \mathcal{T}^t$. Any violations of the output specifications from the customer's desired ones will cause a "contractually agreed" penalty, and the objective function is to minimize this penalty. The prescription of converting this problem to a general pooling problem by Boland et al. (2015) is as follows.

- **Input Nodes**: Create the input node s_p^τ for each supply coming into stockpile p at time $\tau \in \mathcal{T}_p^s$.
- **Pool Nodes**: Create the pool node i_p^τ for each supply coming into stockpile p at time $\tau \in \mathcal{T}_p^s$.
- **Output Nodes**: Create the output node j^τ for each demand at time $\tau \in \mathcal{T}^t$.
- **Input-to-Pool Arcs**: Create an arc from input node s_p^τ to pool node i_p^τ for each supply coming into stockpile p at time $\tau \in \mathcal{T}_p^s$.
- **Pool-to-Pool Arcs**: Create an arc from i_p^τ to $i_p^{\tau'}$ where $\tau' \in \mathcal{T}_p^s$ is the time of the "immediate successor" supply of the one at time $\tau \in \mathcal{T}_p^s$ coming to the stockpile p.
- **Pool-to-Output Arcs**: Create an arc from i_p^τ to $j^{\tau''}$ where $\tau'' \in \mathcal{T}^t$ is the time of the "immediate successor" demand of the supply s_p^τ coming to stockpile p at time $\tau \in \mathcal{T}_p^s$.

We also add one extra pool with time $\tau = \infty$ for the supply surplus of each stockpile whose summation is all being directed to an extra output node that we add. The amount of incoming flow to the last output equals the summation of all the supplies minus the summation of all the demands. Figure 4 shows a mining problem instance.

Since the mining problem is mostly large-scale and has some unique features, such as being time-indexed, which increases the size of the problem even more, it may not be a good idea to perform Optimization-Based Bound Tightening on it. Therefore, we propose some simple and cheap methods to improve the bounds of this problem. Let us consider Figure 4 as a part of a mining problem instance which is formulated as a general pooling problem in which the nodes are placed vertically to reflect the time of supply/demand (the supply/demand node's names represent the ordering of their time). In addition, there are two stockpiles in this example, and we can see the input and pool nodes are aligned in two lines to show which nodes are from the same stockpile (even supply and pool nodes are from stockpile 1 and odds are from stockpile 2). Algorithm 1 shows how we can improve the bounds of different nodes and arcs of this instance.

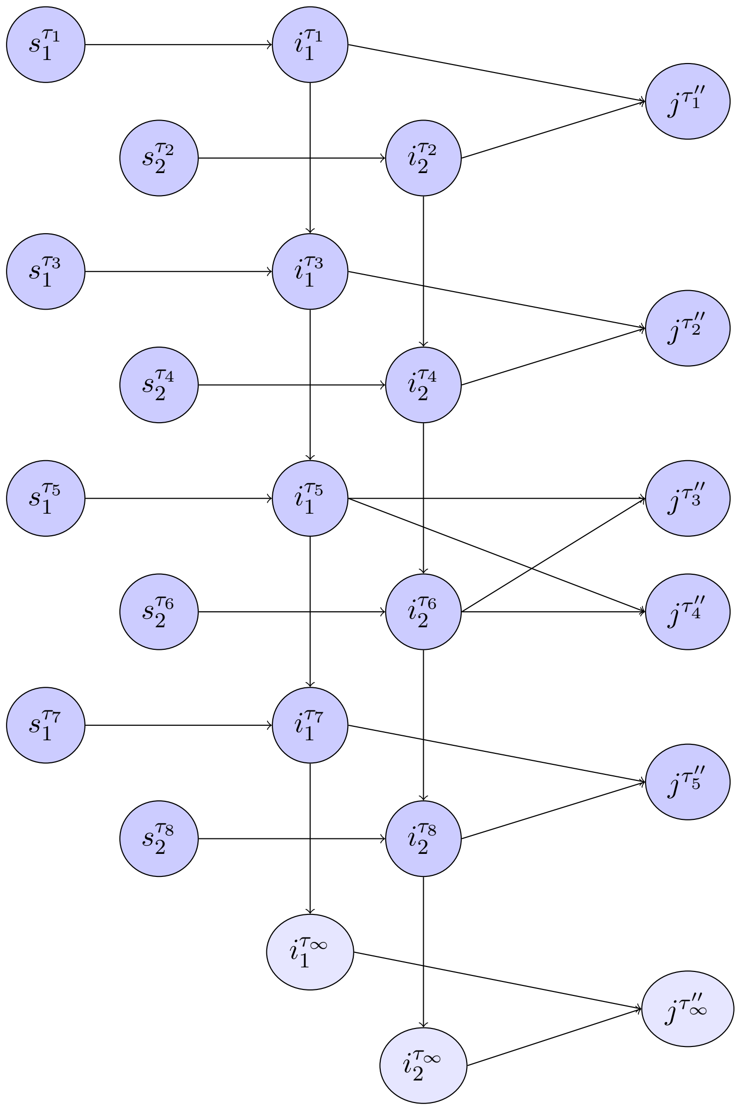

Figure 4: A Mining Problem Instance

Algorithm 1 Bound Tightening for the Mining Problem

1: Bounds of each input node and its outgoing arc equals its amount of supply;

$$L_s = U_s = l_{si} = u_{si} = q_s \qquad \forall s \in S, i \in N_s^+.$$

2: Bounds of each output node equals its amount of demand;

$$L_t = U_t = d_t \qquad \forall t \in T.$$

3: The lower bound of each pool-to-terminal arc is improved as:

$$l_{it} = \max\{L_t - \sum_{p \in P_i} U_p, 0\} \qquad \forall (i,t) \in A, i \in I, t \in T.$$

The upper bound of this arc is improved as:

$$u_{it} = \min\{U_i, U_t\} \qquad \forall (i,t) \in A, i \in I, t \in T.$$

4: The lower bound of each pool-to-pool arc is calculated by:

$$l_{ij} = \max\{L_i - \sum_{(i,t) \in A, t \in T} U_t, 0\} \qquad \forall (i,j) \in A, i,j \in I,$$

To improve the upper bound of this arc, we calculate the following:

$$u_{ij} = U_i - \sum_{(i,t) \in A, t \in T} l_{it} \qquad \forall (i,j) \in A, i,j \in I,$$

In addition to the amount of supply surplus after meeting each demand. The upper bound of each pool-to-pool arc equals the minimum between the supply surplus and u_{ij}.

5: The bounds of each pool equals the summation of the bounds of all its incoming arcs;

$$L_i = \sum_{j \in N_i^-} l_{ji} \text{ and } U_i = \sum_{j \in N_i^-} u_{ji} \qquad \forall i \in I.$$

5 Computations

In this section, we present the results of our experiments on two different sets of generalized pooling problem instances. First, we consider 13 well-known standard pooling problem instances from the literature (Haverly, 1978; Adhya et al., 1999; Foulds et al., 1992; Ben-Tal et al., 1994). We have generalized them by adding the arcs (i,j) and (j,i) for each pair of pools $i,j \subseteq I$, where $i \neq j$ (Alfaki and Haugland, 2013a). Second, we use the data of the real-world mining problem instances based on the work of Boland et al. (2015). In what follows, we report the results of the exact methods based on the original *Source-Based* and *Terminal-Based* rank formulations as well as the outcomes of the experiments with different types of relaxations, restrictions, and valid inequalities which we discussed in Sections 3 and 4. We perform all the experiments with and without the bound tightening and report the results separately. Table 2 shows the methods and notations we use.

Table 2: Computational Methods

Method		Notation	
		Source-Based	*Terminal-Based*
LP Relaxations	*Column-wise*	$\mathcal{F}_1^{\mathcal{S}}$	$\mathcal{F}_2^{\mathcal{T}}$
	Row-wise	$\mathcal{F}_2^{\mathcal{S}}$	$\mathcal{F}_1^{\mathcal{T}}$
	Row-wise ∩ *Column-wise*	$\mathcal{F}_3^{\mathcal{S}}$	$\mathcal{F}_3^{\mathcal{T}}$
	*Row-column**	$\mathcal{F}_4^{\mathcal{S}}$	$\mathcal{F}_4^{\mathcal{T}}$
MIP Relaxations	Discretizing q considering X's column-sum	$\mathcal{M}_1^{\mathcal{S}}(H)$	$\mathcal{M}_2^{\mathcal{T}}(H)$
	Discretizing q considering X's row-sum	$\mathcal{M}_2^{\mathcal{S}}(H)$	$\mathcal{M}_1^{\mathcal{T}}(H)$
Valid Inequalities	Obtained by multiplication of (9a) and (9b)*	$\mathcal{V}_{ab}^{\mathcal{S}}$	$\mathcal{V}_{ab}^{\mathcal{T}}$
	Obtained by multiplication of (9a) and (9c)*	$\mathcal{V}_{ac}^{\mathcal{S}}$	$\mathcal{V}_{ac}^{\mathcal{T}}$

* Developed in this paper

All the experiments are implemented in Python 3.7, and optimization problems are solved by Gurobi 9.1.1 on an Intel(R) 3.7 GHz processor and 64 GB RAM workstation. The time limit for each experiment is set to one hour. Also, we have utilized the Python `JobLib` package to perform OBBT in parallel for each pair of $(i, j) \in A$ and node $i \in N$.

5.1 Literature Instances

In this section, we focus on the instances from the literature and perform different experiments. Table 13 shows the total number of input, pool, and output nodes as well as the total number of arcs and specifications in each of the generalized instances from the literature.

5.1.1 Exact Formulations

First, we solve these instances with the original *Source-Based* and *Terminal-Based* formulations. Table 3 shows these results (additional details in Appendix, Tables 14 and 15).

Table 3: Literature Instances: Exact Formulations

Formulation	Gurobi without OBBT		Gurobi with OBBT		
	Time	%O-Gap	Prep. Time*	Time	%O-Gap
Source-Based	1117.10	0.66%	18.62	185.45	0.00%
Terminal-Based	**277.43**	0.67%	18.62	**2.65**	0.00%

* Preprocessing Time

This table presents the running time and the optimality gap (O-Gap) when using the Gurobi solver to obtain the 'Exact' solution for the literature instances. It compares the results with and without the OBBT technique. The preprocessing time refers to the overall time taken to compute the bounds for the original objective value plus the OBBT processing time.

In OBBT, to obtain a lower bound for the objective value, we have solved the original *Terminal-Based* formulation in which we have relaxed the bilinear constraint and refer to it as the Multi-Commodity Flow (MCF) formulation in the rest of the paper. In addition, to get an upper bound, we have solved the restriction $\mathcal{G}_2^{\mathcal{T}}(H = 3)$ developed by Dey et al. (2020) for all the instances.

According to the table, the OBBT technique helps to improve the running time and the optimality gap for most of the instances for both formulations. In terms of the running time, the *Terminal-Based* formulation performs better than the *Source-based* formulation even without the bound tightening. Additionally, in the *Terminal-Based* formulation, the Gurobi is able to close the gap in a much shorter time than the previous formulation.

Generally, we can say that performing the OBBT technique on the generalized version of literature instances is advantageous on average, and the time and optimality gap improvements are more significant for the *Terminal-Based* formulation.

5.1.2 Linear Programming Relaxations

In this section, we investigate the performance of different relaxations we have explained for the pooling problem instances of the literature. Also, we report the results of these methods with their original bounds and with the improved bounds to evaluate the effect of OBBT on these outer approximations.

Table 4: Literature Instances **without OBBT** (LP Relaxations)

Formulation	$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_3$		$\mathcal{F}_4$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Source-Based	0.04	15.65%	0.03	15.72%	0.15	15.65%	0.15	15.65%
Terminal-Based	**0.02**	15.65%	**0.02**	**14.60%**	**0.03**	**14.53%**	**0.03**	**14.53%**

Table 4 shows the results of the LP relaxations without the bound improvements (additional details in Appendix, Tables 16 and 17). This table indicates the running time and the duality gap (D-Gap). To calculate this gap, we consider the objective values of the 'Exact' obtained by 'Gurobi with OBBT' with 0.00% optimality gap as the upper bound (UB) and the bounds obtained by the relaxations as the lower bound (LB) and use the following equation:

$$\text{Gap} = \frac{UB - LB}{|UB|} \times 100 \tag{34}$$

In general, the average running time of the LP relaxations is much smaller than 'Exact'. Without performing OBBT, all the LP relaxations of both formulations yield almost the same duality gap while the *Terminal-Based* formulation is able to give slightly better duality gap percentages. As we can see, $\mathcal{F}_3$ is better than $\mathcal{F}_1$ and $\mathcal{F}_2$, and interestingly it is equal to $\mathcal{F}_4$ here.

Table 5: Literature Instances **with OBBT** (LP Relaxations)

Formulation	$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_3$		$\mathcal{F}_4$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Source-Based	0.05	7.99%	**0.01**	**4.96%**	**0.02**	4.36%	0.03	4.35%
Terminal-Based	**0.01**	**4.96%**	0.02	7.99%	0.03	4.36%	0.03	4.35%

In Table 5, we can see the results of the LP relaxations of the *Source-Based* and *Terminal-Based* formulations with OBBT (additional details in Appendix, Tables 18 and 19). According to the results, the *column-wise* relaxations of the *Source-Based* and *Terminal-Based* formulations ($\mathcal{F}_1^{\mathcal{S}}$ and $\mathcal{F}_2^{\mathcal{T}}$, respectively) have the same performance while solving the literature instances of the pooling problem and give the duality gap percentage of 7.99 on average. Identically, the *row-wise* relaxations of these two formulations ($\mathcal{F}_2^{\mathcal{S}}$ and $\mathcal{F}_1^{\mathcal{T}}$) give the same solution qualities with the average duality gap percentage of 4.96. In addition, the intersection of the *row-wise* and *column-wise* ($\mathcal{F}_3$) and the *row-column* relaxation ($\mathcal{F}_4$) perform better than the previous LP relaxations which just consider imposing bounds on the row-sum or the column-sum. Moreover, $\mathcal{F}_3$ and $\mathcal{F}_4$ are interestingly equal for this data set.

We can realize that performing OBBT on the literature instances before utilizing the LP relaxations to solve them improves the duality gap significantly and this improvement is more significant than that of the 'Exact' solutions.

5.1.3 Discretization Relaxations

In this section, we evaluate the performance of the MIP relaxations in solving the generalized version of the literature instances. We compare the results of these methods before and after applying the OBBT technique while discretizing the variable q. Dey et al. (2020) have investigated the impact of the different discretization levels $H = 1, \ldots, 5$ and shown that a good choice for the pooling problem that balances accuracy and computational effort is $H = 3$. Therefore, we have considered the same level to run the experiments using MIP relaxations and restrictions.

Table 6: Literature Instances: MIP Relaxations ($H = 3$)

OBBT	$\mathcal{M}_1^{\mathcal{S}}(H)$		$\mathcal{M}_2^{\mathcal{S}}(H)$		$\mathcal{M}_1^{\mathcal{T}}(H)$		$\mathcal{M}_2^{\mathcal{T}}(H)$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
No	17.38	8.35%	136.38	0.84%	**0.12**	0.61%	0.14	0.14%
Yes	**0.29**	**0.24%**	**4.06**	**0.11%**	0.14	**0.24%**	0.14	**0.11%**

Table 6 shows the results of the different discretization relaxations for the pooling problem instances of the literature with and without OBBT (additional details in Appendix, Tables 20 and 21). In case of having no OBBT, $\mathcal{M}_1^{\mathcal{S}}(H)$ cannot perform as well as the others in terms of the solution quality and gives an average gap percentage of 8.35 for all the instances. In terms of the running time, the MIP relaxations of the *Source-Based* formulation are not as strong as those of the *Terminal-Based* formulation and take more time to solve the problems. This difference is more significant when comparing $\mathcal{M}_2^{\mathcal{S}}(H)$ to the others.

However, we observe that OBBT helps the MIP relaxations of the *Source-Based* formulation to have remarkable improvements in terms of the running time and the duality gap on average as well (additional details in Appendix, Table 21).

Regarding the use of discretization relaxations for the literature instances, OBBT does not make remarkable improvements for the relaxations of the *Terminal-Based* formulation since it has a good performance already. Utilizing this bound improvement method is more beneficial for the relaxations of the *Source-Based* formulation. It helps the model to obtain better bounds in a shorter time.

Generally, MIP relaxations are stronger than the LP methods on average, but they are computationally more expensive and need more time to reach high-quality dual bounds.

5.2 Mining Instances

In this section, we report the results of applying different methods we have described previously to solve real-world cases of the mining problem. We have converted these problems to the generalized pooling problem by the instructions in Section 4.3.2. This set consists of yearly, half-yearly, and quarterly planning time horizons. Table 24 shows the characteristics of the different instances in this dataset. This table shows the number of sources, pools, and terminals plus the overall number of arcs. The number of source-to-pool arcs is indicated by $|ASI|$, and we have used similar notations for pool-to-pool and pool-to-terminal arcs. The number of specifications is the same for all the instances. Supply of the raw materials is coming to two stockpiles indicated by $SP1$ and $SP2$ at different time points.

The supplies of the raw materials must be blended in the pools and mixed again in the output points to meet the demand amount with certain specification requirements. There are four specifications; ash, moisture, sulfur, and volatile, which should not violate the maximum preferable amount specified by the customers. Otherwise, the supplier will be penalized by a contractually agreed amount, and the objective is to minimize this penalty.

5.2.1 Exact Formulations

Table 7 shows the 'Exact' objective value of solving the mining instances (additional details in Appendix, Tables 25 and 26).

Table 7: Mining Instances: Exact Formulations

Formulation	Gurobi		Gurobi with Bounds	
	Time	%O-Gap	Time	%O-Gap
Source-Based	1384.84	3.13%	1767.06	2.79%
Terminal-Based	1690.87	**1.25%**	1520.46	**0.88%**

According Table 7, while using the *Source-Based* formulation, the optimality gap and the running time of the Gurobi are 3.13% and 1457.08 on average, respectively. On the other hand, the *Terminal-Based* formulation finds the solutions with the optimality gap of 1.25% in the running time of 1690.87 on average. Additionally, the average optimality gap of the *Source-Based* formulation, while having updated bounds, has decreased. Moreover, updating the bounds of the problem has a positive impact on the running time and the optimality gap of Gurobi while experimenting with the *Terminal-Based* formulation.

5.2.2 Linear Programming Relaxations

In this section, we utilize the LP relaxations to deal with the generalized pooling problem counterpart of the mining problem instances.

Boland et al. (2015) have used the McCormick envelopes to obtain dual bounds of the mining instances. They have also modeled and solved the mining problem (in addition to its generalized pooling problem counterpart) and reported the best-known primal bounds for these instances. The authors have calculated the duality gap of their relaxation based on the primal bounds they have obtained. Since their primal bounds are cheaper and to have comparable results, we have reported their gap in the tables consisting of the results of our relaxations and calculated the duality gap of the results based on their primal bounds.

Table 8: Mining Instances **without Bounds**: LP Relaxations

Formulation	D-Gap	$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_3$		$\mathcal{F}_4$	
	(Boland)	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Source-Based	19%	3.37	7.18%	3.77	7.18%	**3.06**	7.18%	**1.91**	7.18%
Terminal-Based	19%	**2.53**	7.18%	**2.25**	7.18%	25.24	7.18%	5.16	7.18%

Table 8 shows the results of the LP relaxations of the *Source-Based* and *Terminal-Based* formulations without performing the bound tightening method (additional details in Appendix, Tables 27 and 28). The results show that without performing the bound improvement methods, all the LP relaxations of the two multi-commodity flow formulations have the same performance and give identical bounds. These bounds are significantly stronger than those reported by Boland et al. (2015). This may be true for the mining problem as a special case of the generalized pooling problem in which the supply and the demand are placed at different points of time.

We have followed the steps defined in Section 4.3.2 to improve the bounds of different nodes and arcs in the mining problem. These steps are cheap and simple, and since they do not require solving optimization problems, they have negligible preprocessing time. Thus, unlike OBBT, we do not report the preprocessing time in this case.

Table 9: Mining Instances **with Bounds** (LP Relaxations)

Formulation	D-Gap	$\mathcal{F}_1$		$\mathcal{F}_2$		$\mathcal{F}_3$		$\mathcal{F}_4$	
	(Boland)	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Source-Based	19%	3.55	5.12%	3.92	**4.98%**	3.99	4.01%	2.75	3.94%
Terminal-Based	19%	**2.14**	**3.38%**	**2.60**	6.32%	**2.40**	**3.05%**	2.92	**2.98%**

Table 9 indicates that performing the bound tightening method improves the quality of the LP relaxations (detailed results in Tables 29 and 30). As we can see, $\mathcal{F}_3^{\mathcal{S}}$ and $\mathcal{F}_4^{\mathcal{S}}$ give better dual bounds. Recall that our proposed LP relaxation $\mathcal{F}_4^{\mathcal{S}}$ considers bounds on the row-sum and the column-sum of the decomposed flow variables matrices of the pools simultaneously. As we can see from the table, this relaxation outperforms the others and gives stronger dual bounds.

In addition, the duality gap of $\mathcal{F}_3^{\mathcal{T}}$ and $\mathcal{F}_4^{\mathcal{T}}$ are 1% better than those of the *Source-Based* formulation respectively. Therefore, we can say that in both cases of using the updated bounds and without them, the *row-column* relaxation is the best choice to obtain the dual bounds of the mining instances since it is at least as good as the others and gives better dual bounds while using the updated bounds.

5.2.3 Discretization Relaxations

We evaluate the performance of the discretization methods to obtain outer approximations of the mining problems in this section. Table 10 shows the results of different MIP relaxations, which discretize the variable q at the discretization level $H = 3$ for the mining problems (detailed results in Tables 31 and 32). As discussed previously, these MIP methods are generally stronger than LP relaxations, but they need much more time to give high-quality solutions. The results confirm this fact, and we can see that the running time of the discretization relaxations for the mining problems is not as short as those of the LP relaxations, but the duality gap they give is better. To calculate this duality gap, similar to the LP relaxations, we have considered the best primal bounds reported by Boland et al. (2015). Meanwhile, without the updated bounds, $\mathcal{M}_1^{\mathcal{S}}(H)$ performs better than the others in terms of the solution quality and average duality gap.

Table 10: Mining Instances (MIP Relaxations ($H = 3$))

Bounds	$\mathcal{M}_1^{\mathcal{S}}(H)$		$\mathcal{M}_2^{\mathcal{S}}(H)$		$\mathcal{M}_1^{\mathcal{T}}(H)$		$\mathcal{M}_2^{\mathcal{T}}(H)$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
No	1162.50	2.15%	1211.92	3.73%	1889.89	3.33%	1010.25	2.53%
Yes	1167.92	**1.41%**	1339.23	**2.48%**	**1064.11**	**1.06%**	1141.55	**2.18%**

The results of the MIP relaxations in conjunction with the bound tightening indicate that improving the bounds of arcs and nodes of the mining problem has a positive impact on the dual bound obtained by the discretization relaxations. The discretization method $\mathcal{M}_1^{\mathcal{T}}(H)$ has not only improved the duality gap significantly but also the running time is much less than the case of having no updated bounds. The rest of the methods have improved the relaxation quality by making use of the bound tightening method within almost the same amount of average running time. For some instances, the MIP relaxations are running out of the time limit of one hour, which is the cause of the large average of time needed to obtain a high-quality dual bound.

5.2.4 Valid Inequalities

We developed some valid inequalities in Section 2.3, which have the lower bounds of the pools as the denominator of a fraction. These lower bounds exist in the mining problem, and we can improve them. However, for the literature instances, they do not exist generally. Therefore, we only evaluate the new valid inequalities' performance with the mining instances. In this section, we aim to evaluate the performance of adding the valid inequalities to the *row-wise*, *column-wise*, and their intersection as well as the *row-column* relaxations of the *Source-Based* and *Terminal-Based* formulations separately. We report the results of the valid inequalities $\mathcal{V}_{ab}$ and $\mathcal{V}_{ac}$ since they have reasonable performance in the mining instances.

Table 11: Average Running Time and Duality Gap with Bound Tightening (*Source-Based*: LP+Valid Inequalities)

Valid Ineq.	$\mathcal{F}_1^{\mathcal{S}}$		$\mathcal{F}_2^{\mathcal{S}}$		$\mathcal{F}_3^{\mathcal{S}}$		$\mathcal{F}_4^{\mathcal{S}}$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
-	3.55	5.12%	3.92	4.98%	3.99	4.01%	2.75	3.94%
$\mathcal{V}_{ab}^{\mathcal{S}}$	2.60	**4.70%**	2.51	**4.54%**	2.79	**3.92%**	4.32	**3.88%**
$\mathcal{V}_{ac}^{\mathcal{S}}$	2.22	5.04%	2.40	4.97%	2.17	4.00%	4.10	3.93%

Table 11 shows the running time and the duality gap of adding the valid inequalities to the LP relaxations of the *Source-Based* formulation (additional details in Appendix, Tables 33 and 35). We have used the updated bounds of the arcs and nodes obtained by the proposed bound-tightening method. We observe that the best performance of the relaxations is achieved while adding the $\mathcal{V}_{ab}^{\mathcal{S}}$. However, adding the $\mathcal{V}_{ac}^{\mathcal{S}}$ has a positive effect on improving the duality gap of the *Source-Based* formulation. Furthermore, as we expected, the *row-column* relaxation also gives the highest quality dual bounds in this case.

To obtain high-quality dual bounds for the generalized pooling problem counterpart of the mining problem instances, we can make use of this addition as it can yield less duality gap than the LP relaxations in the same average amount of running time.

Table 12: Average Running Time and Duality Gap with Bound Tightening (*Terminal-Based*: LP+Valid Inequalities)

Valid Ineq.	$\mathcal{F}_1^{\mathcal{T}}$		$\mathcal{F}_2^{\mathcal{T}}$		$\mathcal{F}_3^{\mathcal{T}}$		$\mathcal{F}_4^{\mathcal{T}}$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
-	2.14	3.38%	2.60	6.32%	2.40	3.05%	2.92	2.98%
$\mathcal{V}_{ab}^{\mathcal{T}}$	3.43	**3.26%**	3.00	**4.38%**	4.18	**3.03%**	6.70	**2.96%**
$\mathcal{V}_{ac}^{\mathcal{T}}$	1.81	3.38%	2.10	6.32%	1.54	3.05%	4.45	2.98%

Table 12 summarizes the results of LP relaxations of the *Terminal-Based* formulation while we add valid inequalities to them (additional details in Appendix, Tables 34 and 36). As shown in the table, the best dual bound of the relaxations is obtained in addition to $\mathcal{V}_{ab}^{\mathcal{T}}$. For the *Terminal-Based* formulation, adding $\mathcal{V}_{ac}^{\mathcal{T}}$ does not improve the dual bound quality, and it remains the same as the case without any added valid inequalities. In three different cases shown in the table, the best dual bounds are obtained while using the *row-column* relaxation.

6 Conclusion

In this paper, we focused on the pooling problem, a challenging nonlinear and nonconvex network flow problem. Our analysis focused on a recently proposed rank-one-based formulation of the problem. Firstly, we proved that the convex hull of a recurring substructure in the formulation defined as the set of nonnegative, rank-one matrices with bounded row sums, column sums, and the overall sum is second-other cone representable. Secondly, we introduced novel linear programming relaxations based on this analysis, which outperform existing approaches. Thirdly, we derived valid inequalities using the Reformulation Linearization Technique to further strengthen the dual bounds. Finally, to improve the bounds on node and arc capacities, we utilized Optimization-Based Bound Tightening for generic problem instances, and a simple and cost-effective bound-tightening method tailored for time-indexed pooling problem instances. Computational experiments on some benchmark instances showed that our approach has the potential of producing accurate and efficient results thanks to the improved formulations and bound tightening techniques.

Acknowledgments

The authors thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for supporting this work (project number: 119M855).

References

Nilanjan Adhya, Mohit Tawarmalani, and Nikolaos V Sahinidis. A Lagrangian approach to the pooling problem. *Industrial & Engineering Chemistry Research*, 38(5):1956–1972, 1999.

Faiz A Al-Khayyal and James E Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983.

Mohammed Alfaki and Dag Haugland. Comparison of discrete and continuous models for the pooling problem. In *11th workshop on algorithmic approaches for transportation modelling, optimization, and systems*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

Mohammed Alfaki and Dag Haugland. A multi-commodity flow formulation for the generalized pooling problem. *Journal of Global Optimization*, 56(3):917–937, 2013a.

Mohammed Alfaki and Dag Haugland. Strong formulations for the pooling problem. *Journal of Global Optimization*, 56(3):897–916, 2013b.

Charles Audet, Jack Brimberg, Pierre Hansen, Sébastien Le Digabel, and Nenad Mladenović. Pooling problem: Alternate formulations and solution methods. *Management science*, 50(6):761–776, 2004.

Radu Baltean-Lugojan and Ruth Misener. Piecewise parametric structure in the pooling problem: from sparse strongly-polynomial solutions to np-hardness. *Journal of Global Optimization*, 71 (4):655–690, 2018.

A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization.* Society for Industrial and Applied Mathematics, 2001.

Aharon Ben-Tal, Gideon Eiger, and Vladimir Gershovitz. Global minimization by reducing the duality gap. *Mathematical programming*, 63(1):193–212, 1994.

Natashia Boland, Thomas Kalinowski, Fabian Rigterink, and Martin Savelsbergh. A special case of the generalized pooling problem arising in the mining industry. *Optimization Online e-prints*, 2015.

Natashia Boland, Thomas Kalinowski, and Fabian Rigterink. New multi-commodity flow formulations for the pooling problem. *Journal of Global Optimization*, 66(4):669–710, 2016.

Natashia Boland, Thomas Kalinowski, and Fabian Rigterink. A polynomially solvable case of the pooling problem. *Journal of Global Optimization*, 67(3):621–630, 2017.

Samuel Burer and Fatma Kılınç-Karzan. How to convexify the intersection of a second order cone and a nonconvex quadratic. *Mathematical Programming*, 162:393–429, 2017.

Samuel Burer and Adam N Letchford. On nonconvex quadratic programming with box constraints. *SIAM Journal on Optimization*, 20(2):1073–1089, 2009.

Michael Bynum, Anya Castillo, Jean-Paul Watson, and Carl D Laird. Tightening mccormick relaxations toward global solution of the ACOPF problem. *IEEE Transactions on Power Systems*, 34(1):814–817, 2018.

Carleton Coffrin, Hassan L Hijazi, and Pascal Van Hentenryck. Strengthening convex relaxations with bound tightening for power network optimization. In *Principles and Practice of Constraint Programming: 21st International Conference, CP 2015, Cork, Ireland, August 31–September 4, 2015, Proceedings*, pages 39–57. Springer, 2015.

Santanu S Dey and Akshay Gupte. Analysis of milp techniques for the pooling problem. *Operations Research*, 63(2):412–427, 2015.

Santanu S Dey, Burak Kocuk, and Asteroide Santana. Convexifications of rank-one-based substructures in QCQPs and applications to the pooling problem. *Journal of Global Optimization*, 77(2):227–272, 2020.

Leslie Richard Foulds, Dag Haugland, and Kurt Jørnsten. A bilinear approach to the pooling problem. *Optimization*, 24(1-2):165–180, 1992.

Andreas Grothey and Ken McKinnon. On the effectiveness of sequential linear programming for the pooling problem. *arXiv preprint arXiv:2002.10899*, 2020.

Akshay Gupte, Shabbir Ahmed, Santanu S Dey, and Myun Seok Cheon. Relaxations and discretizations for the pooling problem. *Journal of Global Optimization*, 67(3):631–669, 2017.

Akshay Gupte, Arie MCA Koster, and Sascha Kuhnke. A dynamic adaptive discretization algorithm for the pooling problem. In *International Network Optimization Conference 2019 (INOC 2019)*, 2019.

Akshay Gupte, Thomas Kalinowski, Fabian Rigterink, and Hamish Waterer. Extended formulations for convex hulls of some bilinear functions. *Discrete Optimization*, 36:100569, 2020.

Dag Haugland. The computational complexity of the pooling problem. *Journal of Global Optimization*, 64(2):199–215, 2016.

Dag Haugland and Eligius MT Hendrix. Pooling problems with polynomial-time algorithms. *Journal of Optimization Theory and Applications*, 170(2):591–615, 2016.

Co A Haverly. Studies of the behavior of recursion for the pooling problem. *Acm sigmap bulletin*, (25):19–28, 1978.

Jean B Lasserre, Kim-Chuan Toh, and Shouguang Yang. A bounded degree SOS hierarchy for polynomial optimization. *EURO Journal on Computational Optimization*, 5(1-2):87–117, 2017.

Qifeng Li and Vijay Vittal. Convex hull of the quadratic branch AC power flow equations and its application in radial distribution networks. *IEEE Transactions on Power Systems*, 33(1): 839–850, 2017.

Ahmadreza Marandi, Joachim Dahl, and Etienne De Klerk. A numerical evaluation of the bounded degree sum-of-squares hierarchy of Lasserre, Toh, and Yang on the pooling problem. *Annals of Operations Research*, 265(1):67–92, 2018.

Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part I–convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

Viet Pham, Carl Laird, and Mahmoud El-Halwagi. Convex hull discretization approach to the global optimization of pooling problems. *Industrial & Engineering Chemistry Research*, 48(4): 1973–1979, 2009.

Yash Puranik and Nikolaos V Sahinidis. Bounds tightening based on optimality conditions for nonconvex box-constrained optimization. *Journal of Global Optimization*, 67:59–77, 2017.

Hamidur Rahman and Ashutosh Mahajan. Facets of a mixed-integer bilinear covering set with bounds on variables. *Journal of Global Optimization*, 74(3):417–442, 2019.

Asteroide Santana and Santanu S Dey. The convex hull of a quadratic constraint over a polytope. *SIAM Journal on Optimization*, 30(4):2983–2997, 2020.

Mohit Tawarmalani and Nikolaos V Sahinidis. The pooling problem. In *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*, pages 253–283. Springer, 2002.

Asgeir Tomasgard, Frode Rømo, Marte Fodstad, and Kjetil Midthun. Optimization models for the natural gas value chain. In *Geometric modelling, numerical simulation, and optimization*, pages 521–558. Springer, 2007.

A Computations

A.1 Literature Instances

Table 13: Characteristics of the literature instances

Instance	$\lvert S\rvert$	$\lvert I\rvert$	$\lvert T\rvert$	$\lvert A\rvert$	$\lvert K\rvert$
Haverly1	3	2	2	9	1
Haverly2	3	2	2	9	1
Haverly3	3	2	2	9	1
BenTal4	4	2	2	10	1
BenTal5	5	3	5	38	2
Adhya1	5	2	4	15	4
Adhya2	5	2	4	15	6
Adhya3	8	3	4	26	6
Adhya4	8	2	5	20	4
Foulds2	6	2	4	22	1
Foulds3	11	8	16	216	1
Foulds4	11	8	16	216	1
Foulds5	11	4	16	108	1

A.1.1 Exact Formulations

Table 14: Literature Instances (*Source-Based*: Exact).

Instance	Gurobi			Gurobi with OBBT			
	Obj. Value*	Time	% *O*-Gap	Obj. Value	Prep. Time**	Gurobi Time	% *O*-Gap
Haverly1	-400	0.14	0.00%	-400	0.21	0.02	0.00%
Haverly2	-600	0.24	0.00%	-600	0.24	0.05	0.00%
Haverly3	-750	14.76	0.01%	-750	0.29	0.08	0.00%
BenTal4	-450	0.30	0.00%	-450	0.32	0.02	0.00%
BenTal5	-3500	6.59	0.00%	-3500	1.53	1.12	0.00%
Adhya1	-550	3600.00	2.03%	-550	0.52	0.45	0.00%
Adhya2	-550	3600.02	0.07%	-550	0.75	0.23	0.00%
Adhya3	-560	3600.01	2.48%	-561	1.16	3.66	0.00%
Adhya4	-878	3600.16	4.04%	-878	0.87	0.28	0.01%
Foulds2	-1100	0.45	0.00%	-1100	0.92	0.20	0.00%
Foulds3	-8	20.78	0.00%	-8	104.98	1867.48	0.00%
Foulds4	-8	66.73	0.00%	-8	88.67	432.41	0.00%
Foulds5	-8	12.07	0.00%	-8	41.65	104.89	0.00%
Average		**1117.10**	**0.66%**		**18.62**	**185.45**	**0.00%**

* Objective Value. ** Preprocessing time.

Table 15: Literature Instances (*Terminal-Based*: Exact).

Instance	Gurobi			Gurobi+OBBT			
	Obj. Value	Time	% *O*-Gap	Obj. Value	Prep. Time	Gurobi Time	% *O*-Gap
Haverly1	-400	0.03	0.01%	-400	0.21	0.02	0.00%
Haverly2	-600	0.03	0.00%	-600	0.24	0.02	0.00%
Haverly3	-750	0.03	0.00%	-750	0.29	0.00	0.00%
BenTal4	-450	0.03	0.00%	-450	0.32	0.02	0.00%
BenTal5	-3500	0.22	0.00%	-3500	1.53	0.08	0.00%
Adhya1	-550	3600.02	8.67%	-550	0.52	0.09	0.00%
Adhya2	-550	0.25	0.00%	-550	0.75	0.11	0.00%
Adhya3	-561	0.90	0.00%	-561	1.16	0.14	0.00%
Adhya4	-878	0.39	0.00%	-878	0.87	0.12	0.00%
Foulds2	-1100	0.03	0.00%	-1100	0.92	0.02	0.00%
Foulds3	-8	1.68	0.00%	-8	104.98	0.86	0.00%
Foulds4	-8	0.77	0.00%	-8	88.67	0.39	0.00%
Foulds5	-8	2.15	0.00%	-8	41.65	32.53	0.00%
Average		**277.43**	**0.67%**		**18.62**	**2.65**	**0.00%**

A.1.2 Linear Programming Relaxations

Table 16: Literature Instances Without OBBT (*Source-Based*: LP Relaxations)

Instance	$\mathcal{F}_1^{\mathcal{S}}$		$\mathcal{F}_2^{\mathcal{S}}$		$\mathcal{F}_3^{\mathcal{S}}$		$\mathcal{F}_4^{\mathcal{S}}$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Haverly1	0.00	25.00%	0.02	25.00%	0.02	25.00%	0.02	25.00%
Haverly2	0.02	66.67%	0.02	66.67%	0.02	66.67%	0.00	66.67%
Haverly3	0.02	16.67%	0.02	16.67%	0.02	16.67%	0.02	16.67%
BenTal4	0.01	22.22%	0.00	22.22%	0.00	22.18%	0.00	22.18%
BenTal5	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.03	0.00%
Adhya1	0.02	55.18%	0.02	55.68%	0.02	55.18%	0.02	55.18%
Adhya2	0.02	4.51%	0.02	4.51%	0.02	4.51%	0.02	4.51%
Adhya3	0.02	2.46%	0.02	2.46%	0.03	2.46%	0.02	2.46%
Adhya4	0.02	10.76%	0.02	11.21%	0.02	10.76%	0.02	10.76%
Foulds2	0.00	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
Foulds3	0.11	0.00%	0.13	0.00%	0.74	0.00%	0.82	0.00%
Foulds4	0.11	0.00%	0.09	0.00%	0.97	0.00%	0.58	0.00%
Foulds5	0.19	0.00%	0.06	0.00%	0.13	0.00%	0.38	0.00%
Average	**0.04**	**15.65%**	**0.03**	**15.72%**	**0.15**	**15.65%**	**0.15**	**15.65%**

Table 17: Literature Instances Without OBBT (*Terminal-Based*: LP Relaxations)

Instance	$\mathcal{F}_1^{\mathcal{T}}$		$\mathcal{F}_2^{\mathcal{T}}$		$\mathcal{F}_3^{\mathcal{T}}$		$\mathcal{F}_4^{\mathcal{T}}$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Haverly1	0.00	25.00%	0.00	25.00%	0.00	25.00%	0.02	25.00%
Haverly2	0.02	66.67%	0.00	66.67%	0.02	66.67%	0.02	66.67%
Haverly3	0.00	16.67%	0.02	6.67%	0.01	6.67%	0.02	6.67%
BenTal4	0.02	21.30%	0.02	22.22%	0.00	21.30%	0.02	21.30%
BenTal5	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
Adhya1	0.00	55.68%	0.02	52.78%	0.00	52.78%	0.02	52.78%
Adhya2	0.03	4.51%	0.02	4.51%	0.03	4.51%	0.02	4.51%
Adhya3	0.01	2.46%	0.02	2.46%	0.02	2.46%	0.01	2.46%
Adhya4	0.02	11.21%	0.01	9.56%	0.02	9.56%	0.00	9.56%
Foulds2	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.00	0.00%
Foulds3	0.03	0.00%	0.06	0.00%	0.10	0.00%	0.11	0.00%
Foulds4	0.05	0.00%	0.06	0.00%	0.11	0.00%	0.09	0.00%
Foulds5	0.03	0.00%	0.03	0.00%	0.06	0.00%	0.09	0.00%
Average	**0.02**	**15.65%**	**0.02**	**14.60%**	**0.03**	**14.53%**	**0.03**	**14.53%**

Table 18: Literature Instances With OBBT (*Source-Based*: LP Relaxations)

Instance	$\mathcal{F}_1^{\mathcal{S}}$		$\mathcal{F}_2^{\mathcal{S}}$		$\mathcal{F}_3^{\mathcal{S}}$		$\mathcal{F}_4^{\mathcal{S}}$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Haverly1	0.00	0.00%	0.02	0.00%	0.00	0.00%	0.00	0.00%
Haverly2	0.02	37.51%	0.00	0.00%	0.00	0.00%	0.02	0.00%
Haverly3	0.02	4.86%	0.00	0.00%	0.02	0.00%	0.02	0.00%
BenTal4	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
BenTal5	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
Adhya1	0.00	47.79%	0.02	49.01%	0.02	43.90%	0.00	43.90%
Adhya2	0.02	3.95%	0.02	3.73%	0.02	3.23%	0.00	3.23%
Adhya3	0.02	2.11%	0.00	2.39%	0.00	2.11%	0.02	2.06%
Adhya4	0.02	7.62%	0.00	9.34%	0.00	7.41%	0.02	7.41%
Foulds2	0.00	0.00%	0.00	0.00%	0.00	0.00%	0.02	0.00%
Foulds3	0.09	0.00%	0.03	0.00%	0.05	0.00%	0.11	0.00%
Foulds4	0.09	0.00%	0.05	0.00%	0.09	0.00%	0.14	0.00%
Foulds5	0.37	0.00%	0.03	0.00%	0.05	0.00%	0.09	0.00%
Average	**0.05**	**7.99%**	**0.01**	**4.96%**	**0.02**	**4.36%**	**0.03**	**4.35%**

Table 19: Literature Instances With OBBT (*Terminal-Based*: LP Relaxations)

Instance	$\mathcal{F}_1^{\mathcal{T}}$		$\mathcal{F}_2^{\mathcal{T}}$		$\mathcal{F}_3^{\mathcal{T}}$		$\mathcal{F}_4^{\mathcal{T}}$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Haverly1	0.02	0.00%	0.00	0.00%	0.01	0.00%	0.02	0.00%
Haverly2	0.02	0.00%	0.02	37.50%	0.00	0.00%	0.00	0.00%
Haverly3	0.00	0.00%	0.00	4.86%	0.02	0.00%	0.02	0.00%
BenTal4	0.02	0.00%	0.00	0.00%	0.00	0.00%	0.00	0.00%
BenTal5	0.00	0.00%	0.02	0.00%	0.00	0.00%	0.02	0.00%
Adhya1	0.00	49.01%	0.00	47.79%	0.02	43.90%	0.02	43.90%
Adhya2	0.00	3.73%	0.02	3.95%	0.02	3.23%	0.02	3.23%
Adhya3	0.02	2.39%	0.00	2.11%	0.00	2.11%	0.00	2.06%
Adhya4	0.00	9.34%	0.02	7.62%	0.02	7.41%	0.00	7.41%
Foulds2	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
Foulds3	0.06	0.00%	0.08	0.00%	0.17	0.00%	0.06	0.00%
Foulds4	0.06	0.00%	0.08	0.00%	0.14	0.00%	0.09	0.00%
Foulds5	0.03	0.00%	0.05	0.00%	0.06	0.00%	0.14	0.00%
Average	**0.01**	**4.96%**	**0.02**	**7.99%**	**0.03**	**4.36%**	**0.03**	**4.35%**

A.1.3 Discretization Relaxations

Table 20: Literature Instances Without OBBT (MIP Relaxations: $H = 3$).

Instance	$\mathcal{M}_1^{\mathcal{S}}(H)$		$\mathcal{M}_2^{\mathcal{S}}(H)$		$\mathcal{M}_1^{\mathcal{T}}(H)$		$\mathcal{M}_2^{\mathcal{T}}(H)$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Haverly1	0.14	6.45%	0.08	0.00%	0.02	0.00%	0.02	0.00%
Haverly2	0.11	34.44%	0.09	0.00%	0.02	0.00%	0.02	0.00%
Haverly3	0.13	11.90%	0.09	0.00%	0.02	1.90%	0.02	0.00%
BenTal4	0.14	0.00%	0.11	0.00%	0.02	0.00%	0.02	0.00%
BenTal5	0.30	0.00%	2.79	0.00%	0.06	0.00%	0.05	0.00%
Adhya1	0.33	37.60%	0.39	3.05%	0.09	2.66%	0.11	0.80%
Adhya2	0.11	4.51%	0.28	3.05%	0.09	1.93%	0.13	0.80%
Adhya3	0.36	2.46%	4.44	1.96%	0.13	0.47%	0.20	0.17%
Adhya4	0.28	11.14%	0.78	2.89%	0.17	0.92%	0.06	0.00%
Foulds2	0.42	0.00%	0.08	0.00%	0.06	0.00%	0.02	0.00%
Foulds3	119.44	0.00%	777.14	0.00%	0.25	0.00%	0.33	0.00%
Foulds4	83.69	0.00%	937.96	0.00%	0.27	0.00%	0.33	0.00%
Foulds5	20.47	0.00%	48.59	0.00%	0.35	0.00%	0.49	0.00%
Average	**17.38**	**8.35%**	**136.38**	**0.84%**	**0.12**	**0.61%**	**0.14**	**0.14%**

Table 21: Literature Instances With OBBT (MIP Relaxations: $H = 3$).

Instance	$\mathcal{M}_1^{\mathcal{S}}(H)$		$\mathcal{M}_2^{\mathcal{S}}(H)$		$\mathcal{M}_1^{\mathcal{T}}(H)$		$\mathcal{M}_2^{\mathcal{T}}(H)$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
Haverly1	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
Haverly2	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
Haverly3	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
BenTal4	0.02	0.00%	0.03	0.00%	0.02	0.00%	0.02	0.00%
BenTal5	0.08	0.00%	0.30	0.00%	0.12	0.00%	0.06	0.00%
Adhya1	0.08	2.03%	0.12	0.66%	0.08	2.03%	0.05	0.66%
Adhya2	0.05	0.83%	0.11	0.64%	0.05	0.83%	0.06	0.64%
Adhya3	0.09	0.06%	0.55	0.04%	0.09	0.06%	0.08	0.04%
Adhya4	0.11	0.14%	0.09	0.04%	0.08	0.14%	0.05	0.04%
Foulds2	0.05	0.00%	0.06	0.00%	0.03	0.00%	0.02	0.00%
Foulds3	1.14	0.00%	22.80	0.00%	0.41	0.00%	0.41	0.00%
Foulds4	1.14	0.00%	11.14	0.00%	0.45	0.00%	0.55	0.00%
Foulds5	0.98	0.00%	17.52	0.00%	0.41	0.00%	0.48	0.00%
Average	**0.29**	**0.24%**	**4.06**	**0.11%**	**0.14**	**0.24%**	**0.14**	**0.11%**

A.1.4 Discretization Restrictions

Table 22: Literature Instances Without OBBT (MIP Restrictions: $H = 3$).

Instance	$\mathcal{G}_1^{\mathcal{S}}(H)$		$\mathcal{G}_2^{\mathcal{S}}(H)$		$\mathcal{G}_1^{\mathcal{T}}(H)$		$\mathcal{G}_2^{\mathcal{T}}(H)$	
	Time	% P-Gap	Time	% P-Gap	Time	% P-Gap	Time	% P-Gap
Haverly1	0.08	0.00%	0.05	0.00%	0.02	0.00%	0.02	0.00%
Haverly2	0.09	0.00%	0.08	0.00%	0.02	0.00%	0.03	0.00%
Haverly3	0.09	0.00%	0.06	3.45%	0.03	0.00%	0.03	4.17%
BenTal4	0.10	0.00%	0.09	0.00%	0.03	0.00%	0.03	1.61%
BenTal5	3.86	0.00%	0.67	0.00%	1.21	0.00%	0.06	0.00%
Adhya1	0.55	0.08%	0.28	2.42%	0.08	0.08%	0.06	2.42%
Adhya2	0.43	0.08%	0.27	2.42%	0.13	0.08%	0.14	2.42%
Adhya3	6.99	0.12%	7.99	0.04%	0.13	0.12%	0.14	0.04%
Adhya4	1.16	2.49%	0.31	3.15%	0.19	2.49%	0.09	3.15%
Foulds2	4.69	0.00%	0.24	0.00%	0.41	2.33%	0.03	0.00%
Foulds3	3600.19	33.74%	41.77	0.00%	3600.56	12.18%	0.55	0.00%
Foulds4	3600.24	38.86%	34.47	0.00%	3601.05	11.81%	0.41	0.00%
Foulds5	3600.23	84.62%	35.30	0.00%	3600.16	97.61%	1.05	0.00%
Average	**832.20**	**12.31%**	**9.35**	**0.88%**	**831.08**	**9.75%**	**0.20**	**1.06%**

Table 23: Literature Instances With OBBT (MIP Restrictions: $H = 3$).

Instance	$\mathcal{G}_1^{\mathcal{S}}(H)$		$\mathcal{G}_2^{\mathcal{S}}(H)$		$\mathcal{G}_1^{\mathcal{T}}(H)$		$\mathcal{G}_2^{\mathcal{T}}(H)$	
	Time	% P-Gap	Time	% P-Gap	Time	% P-Gap	Time	% P-Gap
Haverly1	0.02	0.00%	0.02	0.00%	0.02	0.00%	0.00	0.00%
Haverly2	0.00	0.00%	0.02	0.00%	0.02	0.00%	0.02	0.00%
Haverly3	0.02	0.00%	0.03	15.36%	0.00	0.00%	0.02	15.36%
BenTal4	0.02	0.00%	0.02	1.61%	0.02	0.00%	0.02	1.61%
BenTal5	0.25	0.00%	0.16	0.00%	0.03	0.00%	0.05	0.00%
Adhya1	0.05	0.05%	0.06	2.38%	0.06	0.05%	0.03	2.38%
Adhya2	0.03	0.05%	0.03	2.38%	0.03	0.05%	0.05	2.38%
Adhya3	0.03	0.13%	0.11	0.05%	0.05	0.13%	0.05	0.05%
Adhya4	0.08	2.43%	0.05	3.11%	0.11	2.43%	0.05	3.11%
Foulds2	0.05	15.74%	0.02	0.00%	0.05	15.74%	0.02	0.00%
Foulds3	3600.26	15.30%	13.55	0.00%	3600.52	14.71%	0.59	0.00%
Foulds4	3600.18	15.94%	9.05	0.00%	3600.22	15.26%	0.41	0.00%
Foulds5	3600.26	162.88%	14.28	0.00%	3600.53	162.88%	1.73	0.00%
Average	**830.86**	**16.35%**	**2.88**	**1.92%**	**830.90**	**16.25%**	**0.23**	**1.92%**

A.2 Mining Instances

Table 24: Characteristics of the mining instances

Instance	$\|S\|$	$\|I\|$	$\|T\|$	$\|A\|$	$\|ASI\|$	$\|AII\|$	$\|AIT\|$	$\|K\|$	$SP1$	$SP2$
2009H2	73	73	49	242	73	71	98	4	37	36
2009Q3	31	31	21	102	31	29	42	4	16	15
2009Q4	38	38	26	126	38	36	52	4	19	19
2010	170	170	122	582	170	168	244	4	82	88
2010H1	86	86	63	296	86	84	126	4	41	45
2010H2	84	84	58	282	84	82	116	4	41	43
2010Q1	39	39	28	132	39	37	56	4	19	20
2010Q2	43	43	30	144	43	41	60	4	20	23
2010Q3	39	39	24	124	39	37	48	4	19	20
2010Q4	43	43	31	146	43	41	62	4	21	22
2011	121	121	94	428	121	119	188	4	61	60
2011H1	67	67	49	230	67	65	98	4	34	33
2011H2	53	53	42	188	53	51	84	4	26	27
2011Q1	35	35	26	120	35	33	52	4	17	18
2011Q2	30	30	21	100	30	28	42	4	16	14
2011Q3	19	19	14	64	19	17	28	4	10	9
2011Q4	28	28	22	98	28	26	44	4	14	14
2012	107	107	78	368	107	105	156	4	51	56
2012H1	65	65	45	218	65	63	90	4	30	35
2012H2	41	41	31	142	41	39	62	4	21	20
2012Q1	26	26	16	82	26	24	32	4	13	13
2012Q2	33	33	22	108	33	31	44	4	16	17
2012Q3	27	27	22	96	27	25	44	4	14	13
2012Q4	16	16	9	48	16	14	18	4	8	8

A.2.1 Exact Formulations

Table 25: Mining Instances without Bound Tightening (Exact)

Instance	Source-Based			Terminal-Based		
	Obj. Value	Time	% O-Gap	Obj. Value	Time	% O-Gap
2009H2	4151473	3600.37	3.57%	4144584	3600.50	1.18%
2009Q3	2281115	99.05	0.00%	2281116	8.95	0.01%
2009Q4	1787783	3600.10	13.70%	1786489	3600.21	5.80%
2010	-	-	-	12520846	3600.62	13.81%
2010H1	8179534	3600.66	24.63%	7268601	3600.52	4.32%
2010H2	4251752	3600.88	2.78%	4242371	3600.53	1.29%
2010Q1	2984583	3600.54	3.54%	2963624	3600.27	0.04%
2010Q2	3067893	3600.41	13.94%	2999787	609.51	0.01%
2010Q3	2456264	57.66	0.00%	2456253	13.02	0.00%
2010Q4	599411	137.38	0.00%	599438	156.28	0.01%
2011	-	-	-	20661572	3601.15	0.72%
2011H1	10584658	233.10	0.00%	10515508	3600.61	0.19%
2011H2	10952784	3600.30	1.13%	10935998	3600.36	0.02%
2011Q1	6346475	316.49	0.00%	6346706	10.85	0.01%
2011Q2	3185908	60.56	0.01%	3185907	66.61	0.01%
2011Q3	2264683	5.22	0.01%	2264640	0.48	0.01%
2011Q4	5028409	102.31	0.01%	5028390	13.41	0.01%
2012	11392712	3601.03	2.33%	11335100	3600.58	1.94%
2012H1	-	-	-	7633206	3600.25	0.68%
2012H2	3385102	600.03	0.01%	3385098	39.65	0.00%
2012Q1	1626709	22.44	0.00%	1626714	22.42	0.00%
2012Q2	2967361	52.31	0.01%	2967365	19.14	0.01%
2012Q3	2395874	103.02	0.01%	2395894	12.11	0.01%
2012Q4	534669	4.89	0.01%	534669	2.83	0.00%
Average		**1457.08**	**3.13%**		**1690.87**	**1.25%**

Table 26: Mining Instances with Bound Tightening (Exact)

Instance	Source-Based			Terminal-Based		
	Obj. Value	Time	% O-Gap	Obj. Value	Time	% O-Gap
2009H2	4162660	3600.57	3.52%	4143629	3600.28	0.69%
2009Q3	2281127	56.51	0.00%	2281165	3.77	0.00%
2009Q4	1779013	3600.28	9.06%	1805388	3600.31	6.92%
2010	-	-	-	12033027	3600.57	7.60%
2010H1	8742718	3600.61	27.82%	7219261	3600.25	3.64%
2010H2	4251731	3600.97	2.65%	4242420	3600.31	1.04%
2010Q1	2984619	3600.70	3.43%	2963606	3069.51	0.01%
2010Q2	3033586	3600.11	9.44%	2999824	1376.30	0.01%
2010Q3	2456269	60.75	0.00%	2456338	16.19	0.01%
2010Q4	599409	89.56	0.01%	599408	64.42	0.00%
2011	21688866	3601.01	6.18%	20660089	3601.18	0.60%
2011H1	10534199	3600.44	1.17%	10515628	3600.58	0.14%
2011H2	10942115	3600.22	0.88%	10935929	2719.32	0.01%
2011Q1	6346475	1864.82	0.01%	6346516	9.61	0.00%
2011Q2	3185907	113.04	0.01%	3185911	56.20	0.01%
2011Q3	2264649	0.94	0.01%	2264581	0.28	0.01%
2011Q4	5028390	74.79	0.00%	5028389	17.38	0.01%
2012	11403482	3600.79	2.37%	11321127	313.46	0.00%
2012H1	7643828	3600.26	1.76%	7631505	3600.23	0.34%
2012H2	3385076	608.52	0.01%	3385065	16.00	0.00%
2012Q1	1626707	16.19	0.01%	1626718	3.66	0.01%
2012Q2	2967367	19.95	0.01%	2967363	11.60	0.01%
2012Q3	2395886	83.22	0.01%	2395869	9.15	0.00%
2012Q4	534669	4.39	0.01%	534680	0.49	0.01%
Average		**1852.12**	**2.97%**		**1520.46**	**0.88%**

A.2.2 Linear Programming Relaxations

Table 27: Mining Instances Without Bound Tightening (*Source-Based*: LP Relaxations)

Instance	D-Gap (Boland et al.)	$\mathcal{F}_1^{\mathcal{S}}$ Time	$\mathcal{F}_1^{\mathcal{S}}$ % D-Gap	$\mathcal{F}_2^{\mathcal{S}}$ Time	$\mathcal{F}_2^{\mathcal{S}}$ % D-Gap	$\mathcal{F}_3^{\mathcal{S}}$ Time	$\mathcal{F}_3^{\mathcal{S}}$ % D-Gap	$\mathcal{F}_4^{\mathcal{S}}$ Time	$\mathcal{F}_4^{\mathcal{S}}$ % D-Gap
2009H2	37%	0.93	8.70%	0.94	8.70%	1.28	8.70%	1.76	8.70%
2009Q3	29%	0.08	4.18%	0.08	4.18%	0.08	4.18%	0.16	4.18%
2009Q4	42%	0.11	29.70%	0.11	29.70%	0.17	29.70%	0.17	29.70%
2010	26%	58.20	11.60%	57.25	11.60%	45.15	11.60%	22.07	11.60%
2010H1	32%	1.34	16.30%	1.45	16.30%	3.73	16.30%	2.47	16.30%
2010H2	15%	1.33	4.52%	1.28	4.52%	1.41	4.52%	2.52	4.52%
2010Q1	20%	0.11	6.15%	0.12	6.15%	0.17	6.15%	0.30	6.15%
2010Q2	35%	0.11	23.91%	0.11	23.91%	0.17	23.91%	0.20	23.91%
2010Q3	20%	0.09	4.49%	0.09	4.49%	0.16	4.49%	0.25	4.49%
2010Q4	29%	0.14	16.29%	0.11	16.29%	0.17	16.29%	0.22	16.29%
2011	19%	15.38	3.17%	15.70	3.17%	10.24	3.17%	6.77	3.17%
2011H1	9%	0.72	2.58%	0.77	2.58%	0.89	2.58%	1.03	2.58%
2011H2	22%	0.22	2.71%	0.30	2.71%	0.42	2.71%	0.50	2.71%
2011Q1	11%	0.08	1.85%	0.08	1.85%	0.12	1.85%	0.11	1.85%
2011Q2	4%	0.05	3.92%	0.05	3.92%	0.06	3.92%	0.08	3.92%
2011Q3	10%	0.02	0.51%	0.03	0.51%	0.02	0.51%	0.05	0.51%
2011Q4	16%	0.06	1.72%	0.05	1.72%	0.06	1.72%	0.06	1.72%
2012	8%	10.58	3.53%	11.03	3.53%	7.94	3.53%	5.58	3.53%
2012H1	5%	0.64	4.22%	0.55	4.22%	0.78	4.22%	0.95	4.22%
2012H2	10%	0.11	1.32%	0.12	1.32%	0.17	1.32%	0.27	1.32%
2012Q1	14%	0.05	11.78%	0.05	11.78%	0.05	11.78%	0.05	11.78%
2012Q2	2%	0.06	0.79%	0.06	0.79%	0.08	0.79%	0.11	0.79%
2012Q3	6%	0.05	1.51%	0.05	1.51%	0.06	1.51%	0.06	1.51%
2012Q4	26%	0.02	6.98%	0.02	6.98%	0.02	6.98%	0.03	6.98%
Average	**19%**	**3.77**	**7.18%**	**3.77**	**7.18%**	**3.06**	**7.18%**	**1.91**	**7.18%**

Table 28: Mining Instances Without Bound Tightening (*Terminal-Based*: LP Relaxations)

Instance	D-Gap (Boland et al.)	$\mathcal{F}_1^{\mathcal{T}}$ Time	$\mathcal{F}_1^{\mathcal{T}}$ % D-Gap	$\mathcal{F}_2^{\mathcal{T}}$ Time	$\mathcal{F}_2^{\mathcal{T}}$ % D-Gap	$\mathcal{F}_3^{\mathcal{T}}$ Time	$\mathcal{F}_3^{\mathcal{T}}$ % D-Gap	$\mathcal{F}_4^{\mathcal{T}}$ Time	$\mathcal{F}_4^{\mathcal{T}}$ % D-Gap
2009H2	37%	0.92	8.70%	0.72	8.70%	1.21	8.70%	1.55	8.70%
2009Q3	29%	0.08	4.18%	0.08	4.18%	0.09	4.18%	0.09	4.18%
2009Q4	42%	0.11	29.70%	0.08	29.70%	0.19	29.70%	0.35	29.70%
2010	26%	31.05	11.60%	33.89	11.60%	31.35	11.60%	22.47	11.60%
2010H1	32%	1.17	16.30%	1.16	16.30%	3.45	16.30%	2.14	16.30%
2010H2	15%	1.05	4.52%	0.95	4.52%	47.99	4.52%	3.14	4.52%
2010Q1	20%	0.11	6.15%	0.09	6.15%	1.78	6.15%	0.30	6.15%
2010Q2	35%	0.14	23.91%	0.11	23.91%	0.25	23.91%	0.31	23.91%
2010Q3	20%	0.09	4.49%	0.08	4.49%	0.22	4.49%	0.27	4.49%
2010Q4	29%	0.30	16.29%	0.14	16.29%	0.62	16.29%	0.33	16.29%
2011	19%	13.88	3.17%	9.03	3.17%	45.68	3.17%	11.64	3.17%
2011H1	9%	0.69	2.58%	0.53	2.58%	0.98	2.58%	1.02	2.58%
2011H2	22%	0.74	2.71%	0.27	2.71%	1.36	2.71%	0.95	2.71%
2011Q1	11%	0.08	1.85%	0.06	1.85%	0.11	1.85%	0.20	1.85%
2011Q2	4%	0.06	3.92%	0.05	3.92%	0.06	3.92%	0.08	3.92%
2011Q3	10%	0.03	0.51%	0.02	0.51%	0.03	0.51%	0.03	0.51%
2011Q4	16%	0.06	1.72%	0.05	1.72%	0.08	1.72%	0.11	1.72%
2012	8%	8.06	3.53%	6.00	3.53%	468.42	3.53%	75.58	3.53%
2012H1	5%	1.86	4.22%	0.50	4.22%	1.53	4.22%	2.72	4.22%
2012H2	10%	0.11	1.32%	0.09	1.32%	0.20	1.32%	0.28	1.32%
2012Q1	14%	0.06	11.78%	0.03	11.78%	0.05	11.78%	0.06	11.78%
2012Q2	2%	0.08	0.79%	0.05	0.79%	0.06	0.79%	0.11	0.79%
2012Q3	6%	0.05	1.51%	0.05	1.51%	0.06	1.51%	0.08	1.51%
2012Q4	26%	0.02	6.98%	0.03	6.98%	0.02	6.98%	0.03	6.98%
Average	**19%**	**2.53**	**7.18%**	**2.25**	**7.18%**	**25.24**	**7.18%**	**5.16**	**7.18%**

Table 29: Mining Instances With Bound Tightening (*Source-Based*: LP Relaxations)

Instance	D-Gap	$\mathcal{F}_1^{\mathcal{S}}$		$\mathcal{F}_2^{\mathcal{S}}$		$\mathcal{F}_3^{\mathcal{S}}$		$\mathcal{F}_4^{\mathcal{S}}$	
	(Boland et al.)	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
2009H2	37%	0.97	5.50%	0.95	4.98%	1.37	4.33%	2.42	4.31%
2009Q3	29%	0.09	2.03%	0.08	1.68%	0.17	1.37%	0.22	1.36%
2009Q4	42%	0.19	19.28%	0.12	22.06%	0.31	14.28%	0.50	14.12%
2010	26%	46.08	9.94%	58.56	9.46%	49.49	8.44%	28.67	8.41%
2010H1	32%	1.81	14.11%	1.75	14.54%	3.45	13.22%	3.91	13.17%
2010H2	15%	1.59	3.72%	1.47	2.13%	5.89	1.80%	3.91	1.80%
2010Q1	20%	0.27	4.11%	0.14	5.05%	0.37	3.67%	0.39	3.59%
2010Q2	35%	0.22	19.41%	0.12	18.93%	0.33	18.05%	0.45	17.63%
2010Q3	20%	0.16	3.47%	0.11	1.68%	0.19	1.48%	0.33	1.48%
2010Q4	29%	0.23	10.57%	0.14	12.67%	0.37	8.90%	0.47	8.82%
2011	19%	15.01	1.95%	16.00	2.14%	16.53	1.56%	10.56	1.53%
2011H1	9%	0.77	1.71%	0.70	1.85%	1.22	1.43%	1.61	1.40%
2011H2	22%	0.55	1.84%	0.52	1.94%	0.64	1.45%	0.89	1.44%
2011Q1	11%	0.11	1.17%	0.09	1.29%	0.16	1.06%	0.19	1.02%
2011Q2	4%	0.08	2.85%	0.08	3.38%	0.16	2.29%	0.23	2.29%
2011Q3	10%	0.03	0.44%	0.03	0.11%	0.03	0.10%	0.04	0.10%
2011Q4	16%	0.08	1.07%	0.07	0.83%	0.16	0.39%	0.20	0.38%
2012	8%	15.74	2.34%	11.98	2.07%	13.32	1.65%	8.47	1.56%
2012H1	5%	0.89	3.02%	0.86	2.59%	1.09	2.15%	1.83	2.01%
2012H2	10%	0.17	0.61%	0.11	0.55%	0.19	0.35%	0.31	0.35%
2012Q1	14%	0.05	7.99%	0.05	4.41%	0.08	4.04%	0.12	3.67%
2012Q2	2%	0.08	0.62%	0.06	0.59%	0.11	0.34%	0.14	0.34%
2012Q3	6%	0.06	0.95%	0.05	0.74%	0.06	0.54%	0.16	0.53%
2012Q4	26%	0.02	4.17%	0.02	3.87%	0.03	3.26%	0.03	3.18%
Average	**19%**	**3.55**	**5.12%**	**3.92**	**4.98%**	**3.99**	**4.01%**	**2.75**	**3.94%**

Table 30: Mining Instances With Bound Tightening (*Terminal-Based*: LP Relaxations)

Instance	D-Gap	$\mathcal{F}_1^{\mathcal{T}}$		$\mathcal{F}_2^{\mathcal{T}}$		$\mathcal{F}_3^{\mathcal{T}}$		$\mathcal{F}_4^{\mathcal{T}}$	
	(Boland et al.)	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
2009H2	37%	0.69	3.01%	0.72	7.69%	1.00	2.74%	1.83	2.68%
2009Q3	29%	0.06	1.23%	0.06	4.02%	0.08	1.14%	0.19	1.12%
2009Q4	42%	0.12	10.61%	0.09	26.31%	0.22	8.17%	0.44	8.01%
2010	26%	30.65	7.20%	39.10	11.13%	31.23	6.83%	23.67	6.68%
2010H1	32%	1.63	9.32%	1.27	15.61%	2.75	8.69%	4.03	8.59%
2010H2	15%	1.44	2.18%	1.14	4.06%	2.53	2.15%	4.55	2.08%
2010Q1	20%	0.14	4.16%	0.14	5.29%	0.25	4.05%	0.45	3.88%
2010Q2	35%	0.17	15.11%	0.16	21.90%	0.27	13.63%	0.50	13.57%
2010Q3	20%	0.09	2.37%	0.09	4.28%	0.16	2.33%	0.27	2.20%
2010Q4	29%	0.22	7.47%	0.19	14.07%	0.36	7.37%	0.75	7.13%
2011	19%	8.28	1.12%	7.77	2.67%	8.88	1.05%	14.13	1.05%
2011H1	9%	0.59	0.87%	0.56	2.47%	1.03	0.84%	1.67	0.83%
2011H2	22%	0.53	1.17%	0.37	2.11%	0.71	1.06%	1.62	1.05%
2011Q1	11%	0.08	0.45%	0.08	1.71%	0.14	0.41%	0.20	0.41%
2011Q2	4%	0.06	2.03%	0.06	3.83%	0.08	2.03%	0.20	1.97%
2011Q3	10%	0.03	0.06%	0.02	0.39%	0.03	0.05%	0.06	0.05%
2011Q4	16%	0.06	0.68%	0.07	1.51%	0.10	0.68%	0.40	0.66%
2012	8%	5.50	1.74%	9.61	3.16%	6.35	1.67%	12.27	1.65%
2012H1	5%	0.69	2.31%	0.66	3.88%	0.98	2.23%	2.05	2.19%
2012H2	10%	0.16	0.39%	0.11	0.89%	0.20	0.34%	0.39	0.33%
2012Q1	14%	0.03	5.89%	0.03	8.14%	0.05	4.07%	0.08	3.72%
2012Q2	2%	0.06	0.43%	0.06	0.74%	0.11	0.43%	0.27	0.43%
2012Q3	6%	0.06	0.56%	0.05	1.05%	0.05	0.49%	0.14	0.48%
2012Q4	26%	0.02	0.80%	0.02	4.87%	0.02	0.80%	0.03	0.80%
Average	**19%**	**2.14**	**3.38%**	**2.60**	**6.32%**	**2.40**	**3.05%**	**2.92**	**2.98%**

A.2.3 Discretization Relaxations

Table 31: Mining Instances Without Bound Tightening (MIP Relaxations: $H = 3$)

Instance	$\mathcal{M}_1^S(H)$		$\mathcal{M}_2^S(H)$		$\mathcal{M}_1^T(H)$		$\mathcal{M}_2^T(H)$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
2009H2	3068.20	1.40%	3601.18	2.94%	3600.41	3.08%	701.86	2.08%
2009Q3	3.55	0.61%	14.35	1.09%	21.46	0.63%	6.88	0.60%
2009Q4	910.54	5.40%	471.86	13.73%	3600.13	11.49%	433.70	5.66%
2010	3600.44	8.64%	3601.90	9.39%	3600.42	8.98%	3600.30	10.29%
2010H1	3600.51	5.20%	3600.48	12.81%	3600.40	10.51%	3600.19	10.80%
2010H2	1526.73	0.84%	1409.23	1.22%	3311.77	2.50%	3461.43	0.95%
2010Q1	50.72	2.53%	227.12	3.28%	1564.49	3.00%	63.70	2.74%
2010Q2	124.53	5.27%	300.13	15.50%	3600.26	10.35%	147.72	4.53%
2010Q3	5.11	1.07%	54.69	1.23%	39.96		31.39	0.88%
2010Q4	60.98	4.30%	102.13	9.46%	3600.34	3.36%	301.17	7.46%
2011	3600.96	1.87%	3600.78	1.84%	3600.28	1.87%	3600.69	1.83%
2011H1	485.36	0.54%	1081.31	0.85%	3600.97	0.94%	277.20	0.52%
2011H2	3531.88	0.53%	3600.34	1.54%	3600.17	1.68%	686.82	0.68%
2011Q1	12.56	0.19%	25.40	0.37%	55.11	0.02%	19.99	0.26%
2011Q2	57.08	1.01%	93.07	2.03%	357.98	1.81%	27.99	1.19%
2011Q3	0.36	0.06%	0.77	0.09%	0.77	0.09%	1.29	0.08%
2011Q4	6.59	0.27%	32.99	0.40%	114.80	0.44%	28.31	0.18%
2012	3600.35	1.84%	3601.05	1.93%	3600.20	2.41%	3600.73	2.91%
2012H1	3600.16	1.81%	3600.82	2.39%	3600.36	2.90%	3600.18	2.27%
2012H2	37.29	2.88%	28.91	0.31%	187.13	2.97%	23.02	0.22%
2012Q1	4.58	3.31%	12.55	3.95%	20.77	3.93%	6.03	2.78%
2012Q2	2.33	0.29%	11.75	0.27%	15.74	0.43%	14.91	0.16%
2012Q3	8.30	0.23%	10.05	0.40%	60.24	0.34%	8.23	0.26%
2012Q4	0.87	1.43%	3.22	2.41%	3.02	2.89%	2.20	1.45%
Average	**1162.50**	**2.15%**	**1211.92**	**3.73%**	**1889.89**	**3.33%**	**1010.25**	**2.53%**

Table 32: Mining Instances With Bound Tightening (MIP Relaxations: $H = 3$)

Instance	$\mathcal{M}_1^S(H)$		$\mathcal{M}_2^S(H)$		$\mathcal{M}_1^T(H)$		$\mathcal{M}_2^T(H)$	
	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap	Time	% D-Gap
2009H2	3600.86	0.84%	3601.17	1.97%	3601.22	0.54%	955.23	1.74%
2009Q3	6.47	0.22%	8.14	0.44%	4.62	0.13%	12.23	0.52%
2009Q4	654.33	4.06%	3600.70	6.71%	211.13	1.59%	1337.54	4.13%
2010	3600.29	7.81%	3601.81	8.41%	3600.36	5.90%	3600.37	10.14%
2010H1	3600.30	4.29%	3600.64	8.56%	3600.58	3.73%	3600.26	11.46%
2010H2	1586.15	0.45%	1935.09	1.02%	1299.25	0.62%	3600.32	1.26%
2010Q1	35.85	1.71%	201.96	2.76%	87.61	1.89%	188.42	2.07%
2010Q2	130.54	2.92%	454.19	11.68%	71.52	2.78%	310.88	3.56%
2010Q3	7.03	0.41%	20.95	0.88%	24.14	0.55%	37.51	0.63%
2010Q4	102.66	3.11%	147.66	4.98%	75.40	2.73%	400.47	4.79%
2011	3600.63	1.54%	3600.32	1.29%	3600.92	0.68%	3600.22	1.69%
2011H1	288.12	0.27%	404.51	0.44%	273.03	0.18%	601.67	0.37%
2011H2	3552.30	0.32%	3600.18	1.24%	3600.51	0.64%	1732.28	0.55%
2011Q1	5.17	0.13%	12.66	0.23%	11.75	0.07%	33.89	0.23%
2011Q2	25.92	0.67%	56.50	0.96%	35.58	0.49%	41.53	0.69%
2011Q3	0.55	0.01%	0.94	0.08%	0.69	0.01%	1.34	0.04%
2011Q4	5.27	0.11%	31.33	0.13%	16.94	0.10%	41.49	0.14%
2012	3600.36	1.18%	3600.29	1.59%	3600.25	1.07%	3600.28	2.14%
2012H1	3600.25	1.45%	3600.58	1.98%	1794.80	0.54%	3600.14	2.33%
2012H2	11.11	0.11%	20.36	0.14%	11.33	0.08%	58.76	0.17%
2012Q1	4.48	1.05%	14.72	2.48%	6.56	0.77%	8.77	2.19%
2012Q2	4.20	0.14%	9.85	0.14%	2.66	0.17%	16.73	0.12%
2012Q3	6.31	0.13%	11.21	0.25%	6.61	0.07%	14.99	0.21%
2012Q4	1.05	0.80%	5.75	1.06%	1.12	0.15%	1.91	1.13%
Average	**1167.92**	**1.41%**	**1339.23**	**2.48%**	**1064.11**	**1.06%**	**1141.55**	**2.18%**

A.2.4 Addition of Valid Inequalities to LP Relaxations

Table 33: Mining Instances With Bound Tightening (*Source-based*: LP Relaxations+$\mathcal{V}_{ab}^{\mathcal{S}}$)

Instance	Gap	$\mathcal{F}_1^{\mathcal{S}}+\mathcal{V}_{ab}^{\mathcal{S}}$		$\mathcal{F}_2^{\mathcal{S}}+\mathcal{V}_{ab}^{\mathcal{S}}$		$\mathcal{F}_3^{\mathcal{S}}+\mathcal{V}_{ab}^{\mathcal{S}}$		$\mathcal{F}_4^{\mathcal{S}}+\mathcal{V}_{ab}^{\mathcal{S}}$	
	(Boland et al.)	Time	% *D*-Gap	Time	% *D*-Gap	Time	% *D*-Gap	Time	% *D*-Gap
2009H2	37%	2.39	5.25%	2.20	4.77%	2.42	4.31%	3.38	4.29%
2009Q3	29%	0.22	1.74%	0.22	1.64%	0.19	1.34%	0.28	1.33%
2009Q4	42%	0.42	17.09%	0.41	15.78%	0.44	14.23%	0.64	14.08%
2010	26%	24.00	9.75%	25.12	9.32%	28.83	8.44%	45.98	8.41%
2010H1	32%	3.58	13.95%	3.70	14.33%	3.95	13.22%	6.23	13.17%
2010H2	15%	3.39	3.13%	3.27	2.11%	3.25	1.80%	5.12	1.80%
2010Q1	20%	0.47	4.07%	0.42	4.53%	0.56	3.67%	0.66	3.59%
2010Q2	35%	0.50	18.93%	0.50	18.30%	0.56	17.89%	0.80	17.55%
2010Q3	20%	0.37	2.53%	0.31	1.65%	0.30	1.48%	0.45	1.48%
2010Q4	29%	0.61	10.06%	0.44	12.67%	0.53	8.90%	0.75	8.82%
2011	19%	11.49	1.89%	10.12	2.11%	11.33	1.55%	17.66	1.53%
2011H1	9%	1.69	1.67%	1.73	1.83%	1.73	1.43%	2.37	1.40%
2011H2	22%	1.11	1.75%	0.97	1.92%	1.03	1.44%	1.62	1.43%
2011Q1	11%	0.22	1.13%	0.25	1.25%	0.28	1.06%	0.34	1.02%
2011Q2	4%	0.23	2.78%	0.20	3.38%	0.20	2.29%	0.30	2.29%
2011Q3	10%	0.05	0.26%	0.03	0.11%	0.03	0.10%	0.06	0.10%
2011Q4	16%	0.27	1.01%	0.18	0.72%	0.16	0.39%	0.28	0.38%
2012	8%	8.53	2.01%	7.72	1.85%	8.34	1.48%	12.53	1.43%
2012H1	5%	1.81	2.58%	1.61	2.28%	1.81	1.89%	2.83	1.83%
2012H2	10%	0.44	0.45%	0.41	0.47%	0.41	0.34%	0.59	0.34%
2012Q1	14%	0.14	5.94%	0.14	2.99%	0.14	2.81%	0.22	2.80%
2012Q2	2%	0.22	0.44%	0.20	0.58%	0.17	0.34%	0.28	0.34%
2012Q3	6%	0.17	0.76%	0.17	0.66%	0.14	0.54%	0.25	0.53%
2012Q4	26%	0.02	3.59%	0.03	3.79%	0.03	3.25%	0.03	3.18%
Average	**19%**	**2.60**	**4.70%**	**2.51**	**4.54%**	**2.79**	**3.92%**	**4.32**	**3.88%**

Table 34: Mining Instances With Bound Tightening (*Terminal-based*: LP Relaxations+$\mathcal{V}_{ab}^{\mathcal{T}}$)

Instance	Gap	$\mathcal{F}_1^{\mathcal{T}}+\mathcal{V}_{ab}^{\mathcal{T}}$		$\mathcal{F}_2^{\mathcal{T}}+\mathcal{V}_{ab}^{\mathcal{T}}$		$\mathcal{F}_3^{\mathcal{T}}+\mathcal{V}_{ab}^{\mathcal{T}}$		$\mathcal{F}_4^{\mathcal{T}}+\mathcal{V}_{ab}^{\mathcal{T}}$	
	(Boland et al.)	Time	% *D*-Gap	Time	% *D*-Gap	Time	% *D*-Gap	Time	% *D*-Gap
2009H2	37%	1.97	2.99%	2.25	5.23%	2.42	2.74%	2.80	2.68%
2009Q3	29%	0.20	1.21%	0.27	2.52%	0.16	1.14%	0.30	1.12%
2009Q4	42%	0.37	8.60%	0.44	11.27%	0.37	8.15%	0.77	7.98%
2010	26%	39.79	7.20%	29.98	10.43%	47.73	6.83%	82.30	6.68%
2010H1	32%	3.87	9.32%	3.33	14.77%	5.60	8.69%	5.06	8.59%
2010H2	15%	3.77	2.18%	4.03	3.10%	4.05	2.15%	5.56	2.08%
2010Q1	20%	0.45	4.09%	0.47	4.98%	0.39	4.05%	0.77	3.88%
2010Q2	35%	0.47	15.08%	0.47	17.77%	0.47	13.60%	0.83	13.56%
2010Q3	20%	0.28	2.37%	0.33	3.01%	0.27	2.33%	0.47	2.20%
2010Q4	29%	0.73	7.37%	0.72	8.39%	0.58	7.37%	1.14	7.13%
2011	19%	12.66	1.12%	10.33	2.26%	17.63	1.05%	25.41	1.05%
2011H1	9%	1.80	0.86%	1.98	2.12%	1.69	0.84%	3.39	0.83%
2011H2	22%	1.44	1.17%	1.77	1.63%	1.62	1.06%	2.66	1.05%
2011Q1	11%	0.31	0.43%	0.28	1.24%	0.28	0.41%	0.45	0.41%
2011Q2	4%	0.20	2.03%	0.20	3.27%	0.17	2.03%	0.33	1.97%
2011Q3	10%	0.05	0.06%	0.06	0.23%	0.05	0.05%	0.06	0.05%
2011Q4	16%	0.19	0.68%	0.20	1.33%	0.20	0.68%	0.34	0.66%
2012	8%	10.62	1.66%	11.25	2.07%	13.19	1.61%	22.71	1.61%
2012H1	5%	2.11	2.20%	2.64	2.61%	2.42	2.14%	3.97	2.14%
2012H2	10%	0.47	0.38%	0.47	0.40%	0.47	0.34%	0.66	0.33%
2012Q1	14%	0.14	5.36%	0.14	3.86%	0.12	3.76%	0.20	3.45%
2012Q2	2%	0.23	0.43%	0.28	0.66%	0.20	0.43%	0.31	0.43%
2012Q3	6%	0.17	0.56%	0.17	0.63%	0.16	0.49%	0.25	0.48%
2012Q4	26%	0.03	0.80%	0.02	1.22%	0.02	0.80%	0.03	0.80%
Average	**19%**	**3.43**	**3.26%**	**3.00**	**4.38%**	**4.18**	**3.03%**	**6.70**	**2.96%**

Table 35: Mining Instances With Bound Tightening (*Source-Based*: LP Relaxations+$\mathcal{V}_{ac}^{\mathcal{S}}$)

Instance	Gap	$\mathcal{F}_1^{\mathcal{S}}+\mathcal{V}_{ac}^{\mathcal{S}}$		$\mathcal{F}_2^{\mathcal{S}}+\mathcal{V}_{ac}^{\mathcal{S}}$		$\mathcal{F}_3^{\mathcal{S}}+\mathcal{V}_{ac}^{\mathcal{S}}$		$\mathcal{F}_4^{\mathcal{S}}+\mathcal{V}_{ac}^{\mathcal{S}}$	
	(Boland et al.)	Time	% *D*-Gap	Time	% *D*-Gap	Time	% *D*-Gap	Time	% *D*-Gap
2009H2	37%	1.00	5.49%	1.25	4.98%	1.87	4.33%	2.22	4.31%
2009Q3	29%	0.08	2.02%	0.08	1.67%	0.11	1.37%	0.16	1.36%
2009Q4	42%	0.19	19.12%	0.12	22.02%	0.30	14.23%	0.47	14.08%
2010	26%	25.44	9.94%	28.22	9.46%	21.62	8.44%	57.93	8.41%
2010H1	32%	2.42	14.11%	3.24	14.54%	3.64	13.22%	4.56	13.17%
2010H2	15%	3.34	3.50%	2.81	2.13%	3.41	1.80%	5.27	1.69%
2010Q1	20%	0.27	4.11%	0.14	5.05%	0.31	3.67%	0.47	3.59%
2010Q2	35%	0.22	19.06%	0.16	18.77%	0.44	17.90%	0.55	17.57%
2010Q3	20%	0.17	3.10%	0.11	1.68%	0.17	1.48%	0.34	1.48%
2010Q4	29%	0.22	10.57%	0.17	12.67%	0.45	8.90%	0.47	8.82%
2011	19%	9.84	1.95%	10.64	2.14%	8.22	1.56%	11.64	1.53%
2011H1	9%	0.84	1.70%	0.92	1.85%	1.16	1.43%	1.92	1.40%
2011H2	22%	0.66	1.84%	0.53	1.94%	0.66	1.45%	1.10	1.44%
2011Q1	11%	0.11	1.17%	0.11	1.28%	0.16	1.06%	0.28	1.02%
2011Q2	4%	0.09	2.85%	0.09	3.38%	0.12	2.29%	0.23	2.29%
2011Q3	10%	0.02	0.44%	0.02	0.11%	0.03	0.10%	0.03	0.10%
2011Q4	16%	0.08	1.06%	0.06	0.83%	0.09	0.39%	0.20	0.38%
2012	8%	6.75	2.31%	7.52	2.07%	7.66	1.65%	7.50	1.56%
2012H1	5%	1.05	2.97%	0.95	2.59%	1.17	2.15%	2.12	2.01%
2012H2	10%	0.20	0.60%	0.14	0.54%	0.31	0.35%	0.34	0.35%
2012Q1	14%	0.06	7.75%	0.05	4.41%	0.08	4.04%	0.14	3.67%
2012Q2	2%	0.08	0.62%	0.08	0.59%	0.09	0.34%	0.17	0.34%
2012Q3	6%	0.06	0.95%	0.06	0.74%	0.06	0.54%	0.14	0.53%
2012Q4	26%	0.03	3.75%	0.02	3.86%	0.02	3.25%	0.03	3.18%
Average	**19%**	**2.22**	**5.04%**	**2.40**	**4.97%**	**2.17**	**4.00%**	**4.10**	**3.93%**

Table 36: Mining Instances With Bound Tightening (*Terminal-Based*: LP Relaxations+$\mathcal{V}_{ac}^{\mathcal{T}}$)

Instance	Gap	$\mathcal{F}_1^{\mathcal{T}}+\mathcal{V}_{ac}^{\mathcal{T}}$		$\mathcal{F}_2^{\mathcal{T}}+\mathcal{V}_{ac}^{\mathcal{T}}$		$\mathcal{F}_3^{\mathcal{T}}+\mathcal{V}_{ac}^{\mathcal{T}}$		$\mathcal{F}_4^{\mathcal{T}}+\mathcal{V}_{ac}^{\mathcal{T}}$	
	(Boland et al.)	Time	% *D*-Gap	Time	% *D*-Gap	Time	% *D*-Gap	Time	% *D*-Gap
2009H2	37%	0.75	3.01%	0.69	7.69%	0.99	2.74%	1.66	2.68%
2009Q3	29%	0.05	1.23%	0.05	4.02%	0.06	1.14%	0.17	1.12%
2009Q4	42%	0.12	10.61%	0.11	26.31%	0.22	8.17%	0.44	8.01%
2010	26%	19.78	7.20%	24.98	11.13%	16.40	6.83%	34.86	6.68%
2010H1	32%	2.58	9.32%	3.28	15.61%	1.98	8.69%	5.55	8.59%
2010H2	15%	2.55	2.18%	2.58	4.06%	2.08	2.15%	4.37	2.08%
2010Q1	20%	0.19	4.16%	0.19	5.29%	0.17	4.05%	0.44	3.88%
2010Q2	35%	0.20	15.11%	0.20	21.90%	0.23	13.63%	0.50	13.57%
2010Q3	20%	0.12	2.37%	0.09	4.28%	0.17	2.33%	0.30	2.20%
2010Q4	29%	0.25	7.47%	0.22	14.07%	0.25	7.37%	0.73	7.13%
2011	19%	7.39	1.12%	10.94	2.67%	5.73	1.05%	27.86	1.05%
2011H1	9%	0.75	0.87%	0.59	2.47%	0.86	0.84%	1.61	0.83%
2011H2	22%	0.56	1.17%	0.37	2.11%	0.64	1.06%	1.66	1.05%
2011Q1	11%	0.09	0.45%	0.09	1.71%	0.12	0.41%	0.23	0.41%
2011Q2	4%	0.06	2.03%	0.06	3.83%	0.06	2.03%	0.22	1.97%
2011Q3	10%	0.02	0.06%	0.02	0.39%	0.02	0.05%	0.05	0.05%
2011Q4	16%	0.06	0.68%	0.06	1.51%	0.11	0.68%	0.36	0.66%
2012	8%	6.72	1.74%	4.84	3.16%	5.58	1.67%	22.75	1.65%
2012H1	5%	0.70	2.31%	0.72	3.88%	0.91	2.23%	2.03	2.19%
2012H2	10%	0.20	0.39%	0.12	0.89%	0.22	0.34%	0.56	0.33%
2012Q1	14%	0.05	5.89%	0.05	8.14%	0.05	4.07%	0.09	3.72%
2012Q2	2%	0.06	0.43%	0.09	0.74%	0.06	0.43%	0.22	0.43%
2012Q3	6%	0.06	0.56%	0.05	1.05%	0.06	0.49%	0.20	0.48%
2012Q4	26%	0.02	0.80%	0.02	4.87%	0.02	0.80%	0.03	0.80%
Average	**19%**	**1.81**	**3.38%**	**2.10**	**6.32%**	**1.54**	**3.05%**	**4.45**	**2.98%**