

Tightening Methods Based on Nontrivial Bounds on Bilinear Terms

Yifu Chen,^a Christos T. Maravelias^{b*}

*a. Department of Chemical and Biological Engineering, University of Wisconsin-Madison
1415 Engineering Dr., Madison, WI 53706, USA*

*b. Department of Chemical & Biological Engineering and Andlinger Center for Energy and the Environment
Princeton University, 86 Olden Street, Princeton, NJ 08544, USA*

*Corresponding author: maravelias@princeton.edu

Abstract

We develop tightening and solution methods for optimization problems containing bilinear terms. We focus on the bilinear term $w = xy$ with nonnegative variables $x \in [x^L, x^U]$ and $y \in [y^L, y^U]$, where w is semi-continuous and upper and lower bounded by w^U and w^L when positive. w^U and w^L are said to be nontrivial upper and lower bounds if w^U is smaller than $x^U y^U$ and w^L is greater than $x^L y^L$, respectively. We derive a family of valid linear constraints and show that, when one of the nontrivial bounds is active, such constraints are tangent to one branch of the hyperbola that represents the bilinear term. We propose different preprocessing methods for generating strong constraints from the family. Computational results demonstrate the effectiveness of the proposed methods in terms of reducing optimality gap and computational time.

Keywords: Preprocessing, Nonlinear Optimization, Nonconvex Optimization, Semi-Continuous Variables, Valid Constraints

List of Symbols

Indices/Sets

$i \in \mathbf{I}$: Inputs (Streams)

$j \in \mathbf{J}$: Pools

$k \in \mathbf{K}$: Products

$l \in \mathbf{L}$: Properties

Parameters

α_i : Unit cost of stream i

α_{jk}^F : Fixed cost for flow between pool j and product k

β_k : Price of product k

β_k^P : Unit penalty for unmet demand for product k

γ_j : Capacity of pool j

μ_{jk} : Lower bound on positive flow between pool j and product k

ν_{jk} : Capacity of the pipeline between pool j and product k

π_{il} : Value of property l for stream i

ψ_{kl} : Upper bounding specification for property l for product k

φ_k : Minimum demand for product k

ω_k : Maximum demand for product k

Nonnegative continuous variables

F_{ij} : Flow of stream i to pool j

$\hat{F}_{ijk}$: Flow of stream i from pool j to product k

U_k : Unmet demand for product k

$\bar{F}_{jk}$: Flow from pool j to product k

R_{jk} : Split fraction for total inlet flows from pool j to product k

Binary variable

Z_{jk} : = 1 if there is positive flow from pool j to product k

Models

M^C : Model for the pooling problem with only continuous variables

M^{SC} : Model for the pooling problem with semi-continuous variables

1. Introduction

Optimization problems containing bilinear terms have a number of applications in different industrial sectors, from refining (Wicaksono and Karimi 2008; Gounaris, Misener, and Floudas 2009; Misener and Floudas 2012; Kolodziej, Castro, and Grossmann 2013; Gupte et al. 2017; Chen and Maravelias 2020) and wastewater treatment (Bagajewicz 2000; Jeżowski 2010) to mining (Blom et al. 2014; Blom, Pearce, and Stuckey 2016; Boland et al. 2016). Such problems are important in terms of the potential economic benefits that can be achieved if solved efficiently (DeWitt et al. 1989; Kelly and Mann 2003).

One optimization problem containing bilinear terms that has been studied extensively is the pooling problem, which is a nonconvex optimization problem. First studied by Harvey (Haverly 1978), the pooling problem continues to be an active research topic (Misener and Floudas 2009; Gupte et al. 2017). Various formulations for the pooling problem have been proposed (Haverly 1978; Ben-Tal, Eiger, and Gershovitz 1994; Tawarmalani and Sahinidis 2002; Audet et al. 2004; Alfaki and Haugland 2013; Boland, Kalinowski, and Rigterink 2016), and a number of variants of the pooling problem have been studied. For example, Meyer and Floudas (Meyer and Floudas 2006) studied the generalized pooling problem where there can be flows between pools. Misener et al. (Misener, Gounaris, and Floudas 2010) studied the pooling problem containing complex emission constraints. D'Ambrosio et al. (D'Ambrosio, Linderoth, and Luedtke 2011) studied valid constraints for the pooling problem with binary variables.

Solution methods for optimization problems containing bilinear terms have been studied extensively. One research focus is to tighten the linear relaxation of such problem. For example, Gounaris et al. studied different piecewise linear relaxation methods for bilinear terms and compared their computational performance (Gounaris, Misener, and Floudas 2009), Castro proposed piecewise linear relaxations with variable bounds tightening (Castro 2015), Dey and Gupte analyzed mixed-integer linear programming (MILP) techniques to address bilinear terms (Dey and Gupte 2015), Chen and Maravelias utilized information from the given problem parameters to derive valid linear constraints that tighten the relaxed problem in which bilinear terms are replaced by linear inequalities (Chen and Maravelias 2020). Nonlinear relaxations of such problem have also been studied. For example, Kimizuka et al. studied the second order cone relaxation of such problem (Kimizuka, Kim, and Yamashita 2019) and Luedtke et al. studied a strong convex nonlinear relaxation derived from extended formulation (Luedtke et al. 2020).

To effectively solve optimization problems containing bilinear terms, one common approach is to construct convex relaxations of bilinear terms at each node in a branch-and-bound (B&B) algorithm. Consider the bilinear term $w = xy$ with nonnegative variables $x \in [x^L, x^U]$ and $y \in [y^L, y^U]$ and the

set $\mathbf{S} = \{(w, x, y) \in \mathbb{R}_+^3: x^L \leq x \leq x^U, y^L \leq y \leq y^U, w = xy\}$. Using the method proposed by McCormick (McCormick 1976) leads to four linear inequalities parameterized by x^L, x^U, y^L , and y^U ; the four linear inequalities describe the convex hull of $\mathbf{S}$.

If w is also upper bounded by a positive parameter $w^U < x^U y^U$, then w^U is said to be a nontrivial upper bound on w . Similarly, if w is lower bounded by a positive parameter $w^L > x^L y^L$, then w^L is said to be a nontrivial lower bound on w . In the presence of nontrivial bounds, we consider the set

$$\mathbf{S}_1^+ = \{(w, x, y) \in \mathbb{R}_+^3: x^L \leq x \leq x^U, y^L \leq y \leq y^U, w^L \leq w \leq w^U, w = xy\}$$

The convex hull of $\mathbf{S}_1^+$ has been studied by Belotti et al. (Belotti, Miller, and Namazifar 2010; 2011). Specifically, they showed that it can be described with infinitely many linear inequalities, some of which belong to a family of inequalities called “lifted tangent inequalities”. More recently, Anstreicher et al. studied the convex hull representation for bilinear terms with bounds on the product, and derived closed-form representations containing second-order cone constraints (Anstreicher, Burer, and Park 2020). We note that both works mentioned above focus on sets that contain continuous variables only.

In this paper we focus on the following set:

$$\mathbf{S}_1 = \{(w, x, y, Z) \in \mathbb{R}_+^3 \times \{0,1\}: x^L Z \leq x \leq x^U, y^L Z \leq y \leq y^U Z, w^L Z \leq w \leq w^U Z, w = xy\}$$

which can be viewed as a generalization of $\mathbf{S}_1^+$ since $\mathbf{S}_1$ becomes $\mathbf{S}_1^+$ when $Z = 1$. We derive a family of valid linear constraints for $\mathbf{S}_1$, and show that, in the presence of nontrivial bounds, such constraints tighten the convex relaxation of the bilinear term obtained using the McCormick inequalities. We note that when $Z = 1$ the constraints proposed in this paper coincide with a subset of the “lifted tangent inequalities” and can be viewed as outer approximation cuts of the second order cone presented by Anstreicher et al. (Anstreicher, Burer, and Park 2020). However, compared to previous work by Belotti et al., the constraints proposed here are given in a different (parameterized) form, which enables straightforward optimization-based generation for such constraints. We apply our methods to the pooling problem that (1) contains only continuous variables, and (2) contains binary and semi-continuous variables.

Note that if variable x in $\mathbf{S}_1$ is upper bounded by $x^U Z$ instead of x^U , the resulting set will be the union of a point and a nonconvex set. By relaxing the nonconvex set with the results obtained by Anstreicher et al. (Anstreicher, Burer, and Park 2020), one can obtain a relaxed set that is the union of a point and a convex set. The perspective formulation for this relaxed set has been studied by Günlük and Linderoth (Günlük and Linderoth 2010) who proposed linear constraints known as perspective cuts.

We note that semi-continuous variables are common in models for network flow problems. Papageorgiou et al. (Papageorgiou et al. 2012) studied the transportation problem with product blending containing fixed costs. Such problem leads to a mixed-integer program (MILP), and facet-defining constraints have been proposed. Pooling problems with binary variables have also been studied; for example, D’Ambrosio et al. studied the pooling problem with binary variables that model the on/off of the flow from stream to pool and proposed valid constraints (D’Ambrosio, Linderoth, and Luedtke 2011). Previous works focus on utilizing stream properties and product specifications to derive valid constraints. Here, we propose constraints that are based on nontrivial bounds on the bilinear terms.

This paper is structured as follows. In section 2, we present background material, including problem statement and models for the pooling problem, and the implication of nontrivial bounds for such problem. In section 3 we derive a family of valid linear constraints that utilizes bounds on bilinear terms. In section 4, we propose methods to generate strong constraints from the family for the pooling problem. In section 5, we show computational results including models with constraints generated from different methods and a branch-and-cut algorithm that incorporates the proposed constraints. Throughout the paper, unless otherwise specified, we use Roman lowercase italic letters for indices, Roman uppercase bold letters for sets, Greek lowercase letters for parameters, and Roman uppercase italics for variables.

2. Background

We present the problem statement and nonlinear models for the pooling problem. We introduce nontrivial bounds on bilinear terms and the convex relaxation of bilinear terms in the presence of such bounds.

2.1. Problem statement

In the standard setting, the pooling problem is defined in terms of the following sets:

$i \in \mathbf{I}$: Inputs (Streams)

$j \in \mathbf{J}$: Pools

$k \in \mathbf{K}$: Products

$l \in \mathbf{L}$: Properties

Given are:

α_i : Unit cost of stream i

β_k : Price of product k

γ_j : Capacity of pool j

ν_{jk} : Capacity of the pipeline between pool j and product k

π_{il} : Value of property l for stream i

ψ_{kl} : Upper bounding specification for property l for product k

ω_k : Maximum demand for product k

For any product, the combined flows from all pools to that product must satisfy the corresponding specification. We aim to find flows (from streams to pools and from pools to products) that maximize profit. We assume that there are no flows between pools, no stream flow accumulation in pools, and all product properties are the average of the properties of the streams blended weighted by volume fraction. Without loss of generality, we assume we have only upper bounding specifications.

2.2. Nonlinear models for the pooling problem

2.2.1 Model containing only continuous variables

Various models have been proposed for the pooling problem (Haverly 1978; Ben-Tal, Eiger, and Gershovitz 1994; Tawarmalani and Sahinidis 2002; Audet et al. 2004; Alfaki and Haugland 2013; Boland, Kalinowski, and Rigterink 2016). In this paper, we study models similar to the one proposed by Alfaki and Haugland (Alfaki and Haugland 2013). We define the following nonnegative continuous variables:

F_{ij} : Flow of stream i to pool j

R_{jk} : Split fraction for total inlet flows for pool j to product k ($R_{jk} \in [0,1]$)

$\hat{F}_{ijk}$: Flow of stream i from pool j to product k

We have the following constraints:

Pool capacity:

$$\sum_i F_{ij} \leq \gamma_j, \quad j \quad (1)$$

Product demand:

$$\sum_i \sum_j \hat{F}_{ijk} \leq \omega_k, \quad k \quad (2)$$

Product specifications:

$$\sum_i \sum_j \pi_{il} \hat{F}_{ijk} \leq \psi_{kl} \sum_i \sum_j \hat{F}_{ijk}, \quad k, l \quad (3)$$

Upper bound on the flows from pools to products:

$$\sum_i \hat{F}_{ijk} \leq v_{jk}, \quad j, k \quad (4)$$

Stream splitting:

$$\hat{F}_{ijk} = F_{ij} R_{jk}, \quad i, j, k \quad (5)$$

Note that Eqn. (5) is an equality constraint with a bilinear term.

For split fraction R_{jk} we have:

$$\sum_k R_{jk} = 1, \quad j \quad (6)$$

Eqns. (5) and (6) enforce that there is no flow accumulation in pools.

Reformulation-Linearization Technique (RLT) (Sherali and Adams 1999) constraints can be added to strengthen the formulation. Summing over index k on both sides of Eqn. (5), we have:

$$\sum_k \hat{F}_{ijk} = F_{ij} \sum_k R_{jk}, \quad i, j$$

which, combined with Eqn. (6), leads to:

$$\sum_k \hat{F}_{ijk} = F_{ij}, \quad i, j \quad (7)$$

Another RLT constraint can be obtained by multiplying both sides of Eqn. (1) with R_{jk} (a nonnegative variable):

$$\sum_i F_{ij} R_{jk} \leq \gamma_j R_{jk}, \quad j, k$$

which, combined with Eqn. (5), leads to:

$$\sum_i \hat{F}_{ijk} \leq \gamma_j R_{jk}, \quad j, k \quad (8)$$

The objective function is profit maximization:

$$\max \sum_i \sum_j \left(\sum_k \beta_k \hat{F}_{ijk} - \alpha_i F_{ij} \right) \quad (9)$$

Eqns. (1) - (9) comprise a nonlinear model for the pooling problem which contains only continuous variables and is henceforth referred to as M^C .

2.2.2 Model containing semi-continuous variables

In practice, in addition to the pipeline capacity modeled in Eqn. (4), there may exist a lower bound on $\sum_i \hat{F}_{ijk}$ for each (j, k) pair when $\sum_i \hat{F}_{ijk}$ is nonzero. In other words, when the flow from pool j to product k is nonzero, it must be greater or equal to a positive parameter. Let μ_{jk} denote such a parameter ($\mu_{jk} < v_{jk}$ and $\mu_{jk} < \gamma_j$). We define the following semi-continuous variable:

$\bar{F}_{jk}$: Flow from pool j to product k

and the following binary variable:

Z_{jk} : = 1 if there is positive flow from pool j to product k

We have the following constraints:

$$\bar{F}_{jk} = \sum_i \hat{F}_{ijk}, \quad j, k \quad (10)$$

$$\mu_{jk}Z_{jk} \leq \bar{F}_{jk} \leq \nu_{jk}Z_{jk}, \quad j, k \quad (11)$$

Eqn. (10) is introduced for illustration. Eqn. (11) ensures that when $Z_{jk} = 0$, $\bar{F}_{jk} = 0$; when $Z_{jk} = 1$, $\bar{F}_{jk} \in [\mu_{jk}, \nu_{jk}]$.

Note that for split fraction R_{jk} we now have:

$$\frac{\mu_{jk}}{\gamma_j}Z_{jk} \leq R_{jk} \leq Z_{jk}, \quad j, k \quad (12)$$

When $Z_{jk} = 0$, we have $\bar{F}_{jk} = 0$, and thus $R_{jk} = 0$ for the corresponding split fraction. When $Z_{jk} = 1$, then $\bar{F}_{jk} \geq \mu_{jk}$ so the lower bound on R_{jk} in this case should be μ_{jk}/γ_j and by definition $R_{jk} \leq 1$. Thus, R_{jk} is now also a semi-continuous variable. We note that Eqn. (6) is no longer valid in the model containing semi-continuous variables since it is possible that a pool j has no outlet flows with $Z_{jk} = 0, \forall k$, and thus from Eqn. (12) we have $R_{jk} = 0, \forall k$. However, a relaxation of Eqn. (6), $\sum_k R_{jk} \leq 1, \forall j$, remains valid.

We note that one can derive additional valid constraints from Eqn. (12). Since for every (j, k) pair, we have $(\mu_{jk}/\gamma_j)Z_{jk} \leq R_{jk}$, by summing over index k we have $\sum_k (\mu_{jk}/\gamma_j)Z_{jk} \leq \sum_k R_{jk}, \forall j$, and since $\sum_k R_{jk} \leq 1, \forall j$ we have a knapsack constraint $\sum_k (\mu_{jk}/\gamma_j)Z_{jk} \leq 1, \forall j$, which is equivalent to $\sum_k \mu_{jk}Z_{jk} \leq \gamma_j, \forall j$. For the pooling problem, this means that cannot turn on too many pipelines from a pool if the sum of the lower bounds on flows of those pipelines exceeds pool capacity. In practice, this is not the case, since $\sum_k \mu_{jk}$ will still be smaller than γ_j given that (μ_{jk}/γ_j) is small, and thus such a constraint is unlikely to be active.

We again consider profit maximization with additional fixed cost terms:

$$\max \sum_i \sum_j \left(\sum_k \beta_k \hat{F}_{ijk} - \alpha_i F_{ij} \right) - \sum_j \sum_k \alpha_{jk}^F Z_{jk} \quad (13)$$

Eqns. (1) - (3), (5) and (7) - (13) comprise a nonlinear model for the pooling problem with semi-continuous variables, henceforth referred to as M^{SC} .

For M^{SC} we also consider the objective of minimizing cost considering penalty for unmet demand. Let φ_k denote the minimum demand for product k and define a nonnegative continuous variable U_k for unmet demand for product k , we have:

$$U_k \geq \varphi_k - \sum_j \bar{F}_{jk}, \quad k \quad (14)$$

and the objective function is:

$$\min \sum_i \sum_j \alpha_i F_{ij} + \sum_j \sum_k \alpha_{jk}^F Z_{jk} + \sum_k \beta_k^P U_k \quad (15)$$

where β_k^P is the unit penalty for unmet demand for product k .

2.3. Nontrivial bounds on bilinear terms

2.3.1 Bounds on flow variables

Summing over index i for the constraints in Eqn. (5), we obtain:

$$\sum_i \hat{F}_{ijk} = R_{jk} \sum_i F_{ij}, \quad j, k \quad (16)$$

If we define $\tilde{F}_j$ as follows:

$$\tilde{F}_j = \sum_i F_{ij}, \quad j \quad (17)$$

From Eqn. (7) and Eqn. (17), we can re-write Eqn. (16) as:

$$\bar{F}_{jk} = \tilde{F}_j R_{jk}, \quad j, k \quad (18)$$

Eqn.(18) is an equality constraint with a bilinear term; it is implied from constraints in both M^C and M^{SC} . Note that $\tilde{F}_j$ is upper bounded by γ_j since $\sum_i F_{ij} \leq \gamma_j$ (see Eqn. (1)) and R_{jk} is upper bounded by 1. Thus, from the right-hand-side (RHS) of Eqn. (18) we know that $\bar{F}_{jk}$ is upper bounded by γ_j . However, $\bar{F}_{jk}$ is also upper bounded by v_{jk} since $\sum_i \hat{F}_{ijk} \leq v_{jk}$ (see Eqn. (4)), which is typically smaller than γ_j since, in general, the pipeline capacity from pool to product is significantly smaller than the pool capacity. We note that above analysis holds for both M^C and M^{SC} .

We next examine the lower bounds on both sides of Eqn. (18) for model M^{SC} when $Z_{jk} = 1$. We note that in M^{SC} , $\tilde{F}_j$ is also semi-continuous since we have:

$$\tilde{F}_j \geq \mu_{jk} Z_{jk}, \quad j, k \quad (19)$$

which is implied by Eqn. (11) and Eqn. (18). In this case, from Eqn. (11) we have $\bar{F}_{jk} \geq \mu_{jk}$; we also have $\tilde{F}_j \geq \mu_{jk}$, and from Eqn. (12) we have $R_{jk} \geq \mu_{jk}/\gamma_j$. We note that from the RHS of Eqn. (18) with bounds on $\tilde{F}_j$ and R_{jk} mentioned above, one can only derive the lower bound on $\bar{F}_{jk}$ as μ_{jk}^2/γ_j , which is smaller (thus less tight) than μ_{jk} since we have $\mu_{jk} < \gamma_j$.

2.3.2 Nontrivial bounds

Definition 1 Consider a bilinear term $w = xy$ with $x \leq x^U, y \leq y^U$, and $w \leq w^U$. w^U is said to be a nontrivial upper bound on w if $w^U < x^U y^U$.

Definition 2 Consider a bilinear term $w = xy$ with $x \geq x^L, y \geq y^L$, and $w \geq w^L$. w^L is said to be a nontrivial lower bound on w if $w^L > x^L y^L$.

From Definition 1 and Definition 2, v_{jk} can be nontrivial upper bounds on $\bar{F}_{jk}$ and when $Z_{jk} = 1$, μ_{jk} can be nontrivial lower bounds on $\bar{F}_{jk}$ in M^{SC} .

In this paper we are interested in the set defined as follows:

$$\mathbf{S}_1 = \{(w, x, y, Z) \in \mathbb{R}_+^3 \times \{0,1\}: x^L Z \leq x \leq x^U, y^L Z \leq y \leq y^U Z, w^L Z \leq w \leq w^U Z, w = xy\}$$

with $w^L > x^L y^L$, $w^U < x^U y^U$, and $w^L < w^U$. Set $\mathbf{S}_1$ contains variables and constraints similar to those in $\mathbf{M}^{\text{SC}}$; for a (j, k) pair one can consider $\bar{F}_{jk}$ as w , $\bar{F}_j$ as x , R_{jk} as y , Z_{jk} as Z , and Eqns. (5), (11), (12), and (18) are similar to constraints that define $\mathbf{S}_1$.

When $Z = 1$, $\mathbf{S}_1$ becomes:

$$\mathbf{S}_1^+ = \{(w, x, y) \in \mathbb{R}_+^3: x^L \leq x \leq x^U, y^L \leq y \leq y^U, w^L \leq w \leq w^U, w = xy\} \quad (20)$$

with w^L and w^U being nontrivial lower and upper bounds on w . When $w^L = 0$, $\mathbf{S}_1^+$ represents the feasible space of a bilinear term with nontrivial upper bound, which arises in $\mathbf{M}^C$. We next discuss the implication of nontrivial bounds on the convex relaxation of the bilinear terms.

2.4. Convex relaxation of bilinear terms

Global optimization for nonconvex problems involves solving convex relaxations of the original problem. Using McCormick inequalities (McCormick 1976) to relax $w = xy$ with bounds on x and y defined in $\mathbf{S}_1^+$ we have:

$$w \geq y^L x + x^L y - x^L y^L \quad (21)$$

$$w \geq y^U x + x^U y - x^U y^U \quad (22)$$

$$w \leq y^L x + x^U y - x^U y^L \quad (23)$$

$$w \leq y^U x + x^L y - x^L y^U \quad (24)$$

We define set $\mathbf{S}_2^+$, which is a relaxation of $\mathbf{S}_1^+$, as follows:

$$\mathbf{S}_2^+ = \{(w, x, y) \in \mathbb{R}_+^3: x^L \leq x \leq x^U, y^L \leq y \leq y^U, w^L \leq w \leq w^U, \text{Eqns. (21) - (24)}\}$$

and set $\mathbf{S}_3 = \{w \in \mathbb{R}_+: w = w^U\}$. The feasible spaces defined by $\mathbf{S}_1^+ \cap \mathbf{S}_3$ and $\mathbf{S}_2^+ \cap \mathbf{S}_3$ are shown in Figure 1. In the next section, we derive a family of valid linear constraints for $\mathbf{S}_1$ (thus valid for $\mathbf{S}_1^+$ as well) that tightens $\mathbf{S}_2^+$.

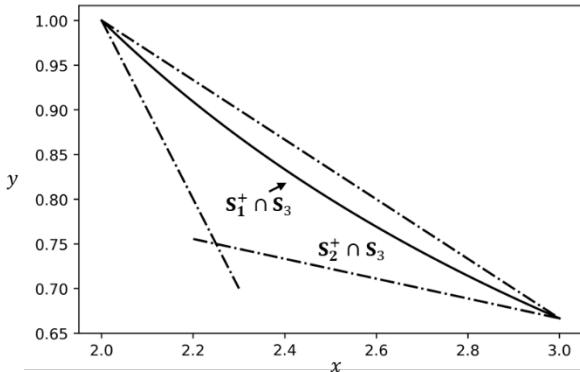


Figure 1. Illustrative graph for bilinear terms xy with $x \in [1, 3]$, $y \in [1/3, 1]$ and its relaxation when the nontrivial upper bound $w^U = 2$ is active. The intersection of $\mathbf{S}_1^+$ and $\mathbf{S}_3$ is the solid curve $xy = w^U$, and the intersection of $\mathbf{S}_2^+$ and $\mathbf{S}_3$ is the triangular region defined by the three dashed lines. Note that while we have $x \in [2, 3]$, $y \in [2/3, 1]$ when $xy = w^U$, Eqns. (21) - (24) that define $\mathbf{S}_2^+$ are generated with $x^L = 1, x^U = 3, y^L = 1/3, y^U = 1$.

3. Valid linear constraints

We first present a family of valid linear constraints for a bilinear term with nontrivial upper and lower bounds, show that such constraints are tangent to the hyperbolas that represent the bilinear term when one of such bounds is active, and discuss the connections with previous works. We then propose methods to generate strong tightening constraints from the family.

3.1. A family of valid linear constraints

We present a family of valid linear constraints for $\mathbf{S}_1$ in Proposition 1.

Proposition 1 $\rho^2 x + w^U y + 2\rho(\sigma_1 w + \sigma_2 Z) \geq 0$ with $\sigma_1 = (\sqrt{w^L w^U} - w^U)/(w^U - w^L)$, $\sigma_2 = w^U(w^L - \sqrt{w^L w^U})/(w^U - w^L)$, and parameter $\rho \geq 0$ is valid for $\mathbf{S}_1$.

Proof.

Since Z is binary, we first consider the case where $Z = 0$. In this case, $\mathbf{S}_1$ becomes:

$$\mathbf{S}_1^- = \{(w, x, y) \in \mathbb{R}_+^3: x^L \cdot 0 \leq x \leq x^U, y^L \cdot 0 \leq y \leq y^U \cdot 0, w^L \cdot 0 \leq w \leq w^U \cdot 0, w = xy\}$$

which is equivalent to:

$$\mathbf{S}_1^- = \{(w, x, y) \in \mathbb{R}_+^3: 0 \leq x \leq x^U, y = 0, w = 0\}$$

One can verify $\rho^2 x + w^U y + 2\rho(\sigma_1 w + \sigma_2 Z) \geq 0$ is valid for $\mathbf{S}_1^-$ by inspection since $\rho^2 x$ is nonnegative and all other terms are zero.

We then consider the case where $Z = 1$. In this case, $\mathbf{S}_1$ becomes $\mathbf{S}_1^+$ in Eqn. (20):

$$\mathbf{S}_1^+ = \{(w, x, y) \in \mathbb{R}_+^3: x^L \leq x \leq x^U, y^L \leq y \leq y^U, w^L \leq w \leq w^U, w = xy\}$$

and the proposed constraint becomes:

$$\rho^2 x + w^U y + 2\rho(\sigma_1 w + \sigma_2) \geq 0 \quad (25)$$

Assuming $(w, x, y) \in \mathbf{S}_1^+$, we first examine $\rho^2 x + w^U y$. Since $a^2 \geq 0$ for any $a \in \mathbb{R}$, we have the valid inequality $(\rho\sqrt{x} - \sqrt{w^U y})^2 \geq 0$, which, after expanding the left-hand-side (LHS), we obtain

$$\rho^2 x - 2\rho\sqrt{w^U xy} + w^U y \geq 0$$

and thus

$$\rho^2 x + w^U y \geq 2\rho\sqrt{w^U xy} \quad (26)$$

Since $(w, x, y) \in \mathbf{S}_1^+$, we have $w = xy$. Thus, Eqn. (26) can be re-written as

$$\rho^2 x + w^U y \geq 2\rho\sqrt{w^U w} \quad (27)$$

With Eqn. (27), we know that the LHS of Eqn. (25) is lower bounded by the following:

$$\rho^2 x + w^U y + 2\rho(\sigma_1 w + \sigma_2) \geq 2\rho\sqrt{w^U w} + 2\rho(\sigma_1 w + \sigma_2)$$

Re-writing the RHS of the above equation in a compact form we have:

$$\rho^2 x + w^U y + 2\rho(\sigma_1 w + \sigma_2) \geq 2\rho(\sqrt{w^U w} + \sigma_1 w + \sigma_2) \quad (28)$$

We next show that the RHS of Eqn. (28) is nonnegative by showing $\sqrt{w^U w} + \sigma_1 w + \sigma_2 \geq 0$ (recall that we have $\rho > 0$). We first examine the zeros of the following quadratic function w.r.t $\sqrt{w}$:

$$\sqrt{w^U w} + \sigma_1 w + \sigma_2 = \sigma_1(\sqrt{w})^2 + \sqrt{w^U w} + \sigma_2 = 0 \quad (29)$$

We note that $\sqrt{w^L}$ is one root for such function, since

$$\sqrt{w^U w^L} + \sigma_1 w^L + \sigma_2 = \sqrt{w^U w^L} + \frac{w^L(\sqrt{w^U w^L} - w^U)}{w^U - w^L} + \frac{w^U(w^L - \sqrt{w^U w^L})}{w^U - w^L} = \frac{\sqrt{w^U w^L}(w^U - w^L) + \sqrt{w^U w^L}(w^L - w^U) + w^U w^L - w^U w^L}{w^U - w^L} = 0$$

And $\sqrt{w^U}$ is the other root for such function, since

$$\sqrt{w^U w^U} + \sigma_1 w^U + \sigma_2 = w^U + \frac{w^U(\sqrt{w^U w^L} - w^U)}{w^U - w^L} + \frac{w^U(w^L - \sqrt{w^U w^L})}{w^U - w^L} = \frac{w^U(w^U - w^L) + w^U(w^L - w^U)}{w^U - w^L} = 0$$

We further note that the coefficient of the quadratic term, σ_1 , in Eq (29), is negative since $w^L < w^U$ (see the definition of σ_1 in Proposition 1). Thus, we have

$$\sqrt{w^U w} + \sigma_1 w + \sigma_2 \geq 0 \quad (30)$$

for $\sqrt{w^L} \leq \sqrt{w} \leq \sqrt{w^U}$, which is equivalent to $w^L \leq w \leq w^U$ since $0 \leq \sqrt{w^L} \leq \sqrt{w} \leq \sqrt{w^U}$. Combining Eqn. (28) and (30), we have:

$$\rho^2 x + w^U y + 2\rho(\sigma_1 w + \sigma_2) \geq 2\rho(\sqrt{w^U w} + \sigma_1 w + \sigma_2) \geq 0 \quad (31)$$

is valid for $(w, x, y) \in \mathbf{S}_1^+$.

Combining both cases for $Z = 0$ and $Z = 1$, we have

$$\rho^2 x + w^U y + 2\rho(\sigma_1 w + \sigma_2 Z) \geq 0 \quad (32)$$

is valid for $\mathbf{S}_1$. ■

When $\rho > 0$, the family of constraints in Eqn. (32) can lead to some strong inequalities, particularly when $w = w^L$ or $w = w^U$ (i.e., when one of the nontrivial bounds is active).

Remark 1 When $w = w^U$, Eqn. (32) becomes $\rho^2 x + w^U y - 2\rho w^U \geq 0$. When $\rho > 0$, one can easily verify that for the branch of the $xy = w^U$ hyperbola with both x and y positive, line $\rho^2 x + w^U y - 2\rho w^U = 0$ is tangent to this hyperbola at point $(w^U/\rho, \rho)$ (the slope for the tangent line at this point is $(-\rho^2/w^U)$).

Remark 2 When $w = w^L$, Eqn. (32) becomes $\rho^2 x + w^U y - 2\rho\sqrt{w^U w^L} \geq 0$. When $\rho > 0$, one can easily verify that for the branch of the $xy = w^L$ hyperbola with both x and y positive, line $\rho^2 x + w^U y - 2\rho\sqrt{w^U w^L} = 0$ is tangent to this hyperbola at point $(\sqrt{w^U w^L}/\rho, \rho\sqrt{w^L/w^U})$ (the slope for the tangent line at this point is $(-\rho^2/w^U)$).

We show an illustrative graph for the proposed constraint and the above two remarks in Figure 2, with numerical examples.

Remark 3 By setting $w^L = 0$ and $Z = 1$, from Proposition 1 we have

$$\rho^2 x + w^U y - 2\rho w \geq 0 \quad (33)$$

which is valid for

$$\mathbf{S}_1^* = \{(w, x, y) \in \mathbb{R}_+^3: x \leq x^U, y \leq y^U, w \leq w^U, w = xy\} \quad (34)$$

Set $\mathbf{S}_1^*$ contains continuous variables only. We note that the variables and constraints involved in $\mathbf{S}_1^*$ are similar to those in $\mathbf{M}^C$. Specifically, for a (j, k) pair one can consider $\bar{F}_{jk}$ as w , $\tilde{F}_j$ as x , and R_{jk} as y .

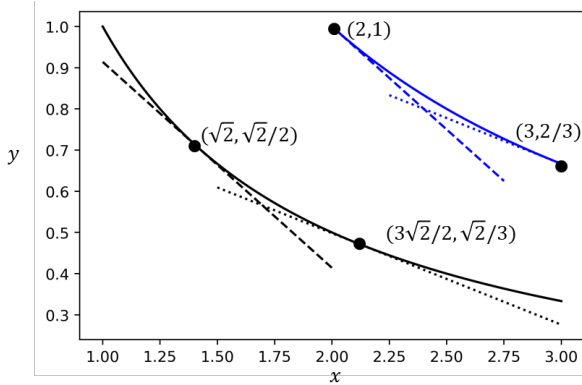


Figure 2. Illustrative graph for bilinear terms $w = xy$ with $x \in [1, 3]$, $y \in [1/3, 1]$, and $w \in [1, 2]$ when one of its nontrivial bounds $w^U = 2$ or $w^L = 1$ is active. The blue curve represents $xy = 2$ and the black curve represents $xy = 1$. Dashed blue and black lines represent the intersection of Eqn. (25) and $w = 2$ and $w = 1$, respectively, with $\rho = 1$. Dotted blue and black lines represent the intersection of Eqn. (25) and $w = 2$ and $w = 1$, respectively, with $\rho = 2/3$. Coordinates for points of tangency are shown in parentheses.

3.2. Generation of strong valid linear constraints

Eqn. (32) contains infinitely many constraints. We propose methods to generate strong tightening constraints. Specifically, given a point $(w^*, x^*, y^*, Z^*) \notin \mathbf{S}_1$ obtained from solving an optimization problem over a relaxation of $\mathbf{S}_1$, we determine the value of ρ to obtain a constraint that cuts off such a point.

3.2.1 Generation based on constraint violation maximization

We consider the following quadratic optimization problem:

$$\min_{\rho} f(\rho) = \rho^2 x^* + w^U y^* + 2\rho(\sigma_1 w^* + \sigma_2 Z^*) \quad (35)$$

which has a closed form solution $\rho = -(\sigma_1 w^* + \sigma_2 Z^*)/x^*$ if $x^* > 0$. Such ρ may lead to a constraint in Eqn. (32) that is violated by (w^*, x^*, y^*, Z^*) , and the violation, measured by the value of $\rho^2 x^* +$

$w^U y^* + 2\rho(\sigma_1 w^* + \sigma_2 Z^*)$, is the greatest. We note that the optimal objective function value to the optimization problem (35) can be nonnegative. If that is the case, Eqn. (32) will not be able to cut off (w^*, x^*, y^*, Z^*) . To address this issue, we first check the sign of the discriminant of $f(\rho)$: $[4(\sigma_1 w^* + \sigma_2 Z^*)^2 - 4x^* w^U y^*]$; if positive, the optimal objective function value to the above optimization problem is guaranteed to be negative, and we proceed to generate a constraint (otherwise, no constraint will be generated).

3.2.2 Generation based on solving the minimum distance problem

The minimum distance problem for constraint generation has been studied (Stubbs and Mehrotra 1999; Sawaya and Grossmann 2005). Here, we focus on the case where $(w^*, x^*, y^*, Z^*) \in \mathbf{S}_3 = \{w \in \mathbb{R}_+ : w = w^U\}$, that is, the nontrivial upper bound is active ($w^* = w^U$ if $Z^* = 1$). Of particular interest is the point $(w^*, x^*, y^*, Z^*) \in \mathbf{S}_3$ with $x^* y^* < w^U$. Note that such a point is not in $\mathbf{S}_1 \cap \mathbf{S}_3$. To find a constraint that cuts off (w^*, x^*, y^*, Z^*) , we first find a point (x, y) on the curve $xy = w^U$ that has the minimum distance to (w^*, x^*, y^*, Z^*) by considering the following optimization problem:

$$\min_{x,y} \left\{ \frac{1}{x^U - x^L} |x - x^*| + \frac{1}{y^U - y^L} |y - y^*| : xy = w^U, x \in [x^L, x^U], y \in [y^L, y^U] \right\}$$

which can be viewed as minimizing the weighted 1-norm distance between (x^*, y^*) and (x, y) . Note that points (x, y) on the curve $xy = w^U$ can be represented using $(w^U/\rho, \rho)$ with ρ being a variable having the same bounds with y . When $xy = w^U$ with $x \leq x^U$, we have $y \in [w^U/x^U, y^U]$, thus $\rho \in [w^U/x^U, y^U]$. We rewrite the above optimization problem as:

$$\min_{\rho} \left\{ \frac{1}{x^U - x^L} \left| \frac{w^U}{\rho} - x^* \right| + \frac{1}{y^U - y^L} |\rho - y^*| : \rho \in [w^U/x^U, y^U] \right\} \quad (36)$$

We claim that the solution to the above problem is the following (see proof in Appendix A):

- (1) If $y^* \leq \sqrt{w^U(y^U - y^L)/(x^U - x^L)} \leq w^U/x^*$, then $\rho = \sqrt{w^U(y^U - y^L)/(x^U - x^L)}$.
- (2) If $\sqrt{w^U(y^U - y^L)/(x^U - x^L)} < y^*$, then $\rho = y^*$.
- (3) If $\sqrt{w^U(y^U - y^L)/(x^U - x^L)} > w^U/x^*$, then $\rho = w^U/x^*$.

After obtaining ρ , we have the point $(w^U/\rho, \rho)$ on the curve $xy = w^U$ that has the minimum distance to (w^*, x^*, y^*, Z^*) . We then generate Eqn. (32) with such ρ . Recall that when the nontrivial upper bound w^U is active, Eqn. (32) is tangent to the curve $xy = w^U$ at point $(w^U/\rho, \rho)$.

4. Solution methods

In this section we present different methods for generating the proposed constraints for model M^C and M^{SC} .

4.1. Methods for model M^C

For model M^C , we consider the following constraint obtained from summing over index $i \in I^* \subseteq I$ for the constraints in Eqn. (5):

$$\sum_{i \in I^*} \hat{F}_{ijk} = R_{jk} \sum_{i \in I^*} F_{ij}, \quad j, k \quad (37)$$

We note that $\sum_{i \in I^*} \hat{F}_{ijk} \leq v_{jk}$ and $\sum_{i \in I^*} F_{ij} \leq \gamma_j$, thus from Remark 3, by considering $\sum_{i \in I^*} \hat{F}_{ijk}$ as w and $\sum_{i \in I^*} F_{ij}$ as x , we have the following valid constraint for M^C :

$$\rho^2 \sum_{i \in I^*} F_{ij} + v_{jk} R_{jk} - 2\rho \sum_{i \in I^*} \hat{F}_{ijk} \geq 0 \quad (38)$$

We next present two constraint generation methods for M^C that determine the value of ρ and the selection of set I^* for Eqn (38).

4.1.1. Generation at the root node

Eqn. (38) can be generated at the root node in multiple rounds. At each round, we solve a linear relaxation of M^C , and generate constraints based on the solution to the relaxed problem. We then resolve the relaxed problem with the generated constraints and perform another round of constraint generation.

Let m denote the rounds of constraint generation. Model M_m^{C-L} contains all constraints in M^C , except that the nonlinear constraint Eqn. (5) is replaced by:

$$\hat{F}_{ijk} \geq \gamma_j R_{jk} + F_{ij} - \gamma_j, \quad i, j, k \quad (39)$$

$$\hat{F}_{ijk} \leq \gamma_j R_{jk}, \quad i, j, k \quad (40)$$

$$\hat{F}_{ijk} \leq F_{ij}, \quad i, j, k \quad (41)$$

M_m^{C-L} also contains the following constraint:

$$\rho_{jkm'}^2 \sum_{i \in I_{jkm'}^*} F_{ij} + v_{jk} R_{jk} - 2\rho_{jkm'} \sum_{i \in I_{jkm'}^*} \hat{F}_{ijk} \geq 0, \quad (j, k, m') \in \mathbf{C}_m \quad (42)$$

where $\mathbf{C}_m$ contains (j, k, m') combinations that lead to Eqn. (42) in all previous rounds, and the set I_{jkm}^* is defined as follows: at each round, we solve M_m^{C-L} and, for each (j, k) pair, define set $I_{jkm}^* = \{i: \hat{F}_{ijk}^* > 0\}$. Given the solution to M_m^{C-L} , we calculate parameter ρ_{jkm} by solving the following optimization problem which is similar to the one in (35):

$$\min_{\rho_{jkm}} g(\rho_{jkm}) = \left(\sum_{i \in I_{jkm}^*} F_{ij}^* \right) \rho_{jkm}^2 - 2 \left(\sum_{i \in I_{jkm}^*} \hat{F}_{ijk}^* \right) \rho_{jkm} + v_{jk} R_{jk}$$

We first check the sign of the discriminant of $g(\rho_{jkm})$: $[4 \left(\sum_{i \in I_{jkm}^*} \hat{F}_{ijk}^* \right)^2 - 4v_{jk} R_{jk}^* \sum_{i \in I_{jkm}^*} F_{ij}^*]$; if positive, then there exists a parameter ρ_{jkm} that leads to a constraint in Eqn. (42) violated by the current solution to M_m^{C-L} . We then calculate $\rho_{jkm} = \sum_{i \in I_{jkm}^*} \hat{F}_{ijk}^* / \sum_{i \in I_{jkm}^*} F_{ij}^*$ (such parameter will lead to a constraint that is violated by the current solution to M_m^{C-L} by the greatest margin). We also

update set $\mathbf{C}_{m+1}$, which contains the index for Eqn. (42). We then solve M_{m+1}^{C-L} , which contains Eqn. (42) that are generated in previous rounds. We repeat until no new constraints are generated or we reach the maximum number of constraint generation rounds (θ). The pseudocode of the aforementioned method is given in Algorithm 1.

Algorithm 1. Constraint generation at root node

Inputs: $c = \text{True}$, $m = 0$, θ , $\mathbf{I}_{jkm}^* = \emptyset$, and $\mathbf{C}_m = \emptyset$
While $c = \text{True}$ **AND** $m < \theta$ **do**
 $c = \text{False}$
 Solve M_m^{C-L} .
 Read solution $\hat{F}_{ijk}^*$, R_{jk}^* , and F_{ij}^*
 $\mathbf{I}_{jkm}^* = \{i: \hat{F}_{ijk}^* > 0\}$
 $\mathbf{C}_{m+1} = \mathbf{C}_m$
 For $j \in \mathbf{J}$ **do**
 For $k \in \mathbf{K}$ **do**
 If $4 \left(\sum_{i \in \mathbf{I}_{jkm}^*} \hat{F}_{ijk}^* \right)^2 - 4v_{jk}R_{jk}^* \sum_{i \in \mathbf{I}_{jkm}^*} F_{ij}^* > 0$ **then**
 $\rho_{jkm} = \sum_{i \in \mathbf{I}_{jkm}^*} \hat{F}_{ijk}^* / \sum_{i \in \mathbf{I}_{jkm}^*} F_{ij}^*$
 $\mathbf{C}_{m+1} = \mathbf{C}_{m+1} \cup \{(j, k, m)\}$
 $c = \text{True}$
 End
 End
 End
 $m = m + 1$
End
Outputs: ρ_{jkm} , $\mathbf{I}_{jkm}^*$, and $\mathbf{C}_{m+1}$

We discuss an example to illustrate the procedure of generating the aforementioned constraint and its effectiveness. We have $\mathbf{I} = \{I1, I2\}$, $\mathbf{J} = \{J1\}$, $\mathbf{K} = \{K1, K2\}$, $\mathbf{L} = \{L1\}$, $\gamma_{J1} = 3$, and the parameters given in Table 1.

Table 1. Parameters for the illustrative example

	α_i	β_k	$v_{J1,k}$	$\pi_{i,L1}$	$\psi_{i,L1}$	ω_k
I1	2	—	—	0.5	—	—
I2	1	—	—	1	—	—
K1	—	10	2	—	0.75	3
K2	—	5	1	—	1	3

Solving the illustrative example with nonlinear model M^C leads to a solution with optimal objective function value of 20.5. Solving the illustrative example using M_0^{C-L} leads to a solution with an objective function value of 21. The optimal solution to M_0^{C-L} is shown in Figure 3.

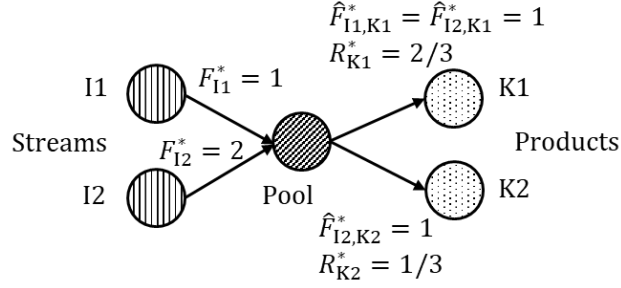


Figure 3. The optimal solution to the illustrative example from solving M_0^{C-L} (index j dropped for simplicity).

We examine the optimal solution for flow to product K2. We have $I_{K2,0}^* = \{I2\}$. Since $4(\hat{F}_{I2,K2}^*)^2 - 4\nu_{K2}R_{K2}^*(F_{I2}^*) = 4 - 8/3 > 0$, we calculate $\rho_{K2,0} = \hat{F}_{I2,K2}^*/F_{I2}^* = 1/2$, and generate the following constraint:

$$(1/4)F_{I2} + R_{K2} - \hat{F}_{I2,K2} \geq 0 \quad (43)$$

In the next round, we solve M_1^{C-L} again after adding Eqn. (43). The optimal objective function value now becomes 20.78, which is closer to the objective function value obtained from solving the nonlinear model M^C (which is 20.5). Figure 4 shows the intersection of Eqn. (43) with $\hat{F}_{I2,K2} = 1$.

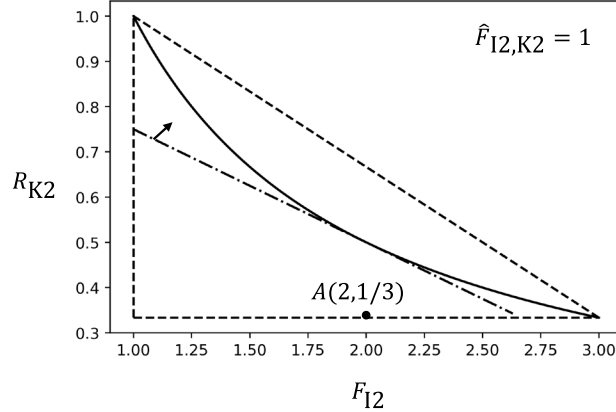


Figure 4. Illustrative graph showing the intersection of Eqn. (43) with $\hat{F}_{I2,K2} = 1$. Solid curve represents $F_{I2}R_{K2} = \hat{F}_{I2,K2}$ in M^C ; dashed lines represent Eqn. (39) - (41) in M_m^{C-L} ; point A represents (F_{I2}^*, R_{K2}^*) obtained from solving M_0^{C-L} ; dot-dashed line represents Eqn. (43).

4.1.2. Generation using a branch-and-cut framework

Let $n \in \mathbf{N} = \{0, 1, \dots\}$ denote nodes in the B&B tree with $n = 0$ being the root node. At each node, we solve M_n^{C-L} which contains all constraints in M^C , except that the nonlinear constraint Eqn. (5) is replaced by:

$$\hat{F}_{ijk} \geq F_{ijn}^L R_{jk} + R_{jkn}^L F_{ij} - F_{ijn}^L R_{jkn}^L \quad (44)$$

$$\hat{F}_{ijk} \geq F_{ijn}^U R_{jk} + R_{jkn}^U F_{ij} - F_{ijn}^U R_{jkn}^U \quad (45)$$

$$\hat{F}_{ijk} \leq F_{ijn}^U R_{jk} + R_{jkn}^L F_{ij} - F_{ijn}^U R_{jkn}^L \quad (46)$$

$$\hat{F}_{ijk} \leq F_{ijn}^L R_{jk} + R_{jkn}^U F_{ij} - F_{ijn}^L R_{jkn}^U \quad (47)$$

where F_{ijn}^L/F_{ijn}^U and R_{jkn}^L/R_{jkn}^U are lower/upper bounds on F_{ij} and R_{jk} at node n , respectively. For the root node, we have $F_{ij,0}^L = 0, \forall i, j, F_{ij,0}^U = \gamma_j, \forall i, j, R_{jk,0}^L = 0 \forall j, k$, and $R_{jk,0}^U = 1, \forall j, k$. The values of F_{ijn}^L/F_{ijn}^U and R_{jkn}^L/R_{jkn}^U will be updated when new nodes are generated.

In addition, M_n^{C-L} also contains the following constraint:

$$\rho_{jkn'}^2 \sum_{i \in \mathbf{I}_{jkn'}^*} F_{ij} + v_{jk} R_{jk} - 2\rho_{jkn'} \sum_{i \in \mathbf{I}_{jkn'}^*} \hat{F}_{ijk} \geq 0, \quad (j, k, n') \in \hat{\mathbf{C}}_n \quad (48)$$

where $\hat{\mathbf{C}}_n$ contains (j, k, n') combinations that lead to Eqn. (48) in all previous nodes. For subsets $\mathbf{I}_{jkn}^*$, at each node we solve M_n^{C-L} and, for each (j, k) pair, define $\mathbf{I}_{jkn}^* = \{i: \hat{F}_{ijk}^* > 0\}$.

We generate Eqn. (48) using Algorithm 2, with the optimal solution to M_n^{C-L} used as inputs. In Algorithm 2 we check if the nonlinear constraints in M^C are satisfied; if not, we aim to generate constraints that cut off the current optimal solution to M_n^{C-L} . For constraint generation, for each (j, k) pair we again define subsets $\mathbf{I}_{jkn}^*$ and check the sign of $[4 \left(\sum_{i \in \mathbf{I}_{jkn}^*} \hat{F}_{ijk}^* \right)^2 - 4v_{jk} R_{jk}^* \sum_{i \in \mathbf{I}_{jkn}^*} F_{ij}^*]$; if positive, we calculate parameter $\rho_{jkn} = \sum_{i \in \mathbf{I}_{jkn}^*} \hat{F}_{ijk}^* / \sum_{i \in \mathbf{I}_{jkn}^*} F_{ij}^*$.

We note that Eqn. (48) is globally valid, since different variable bounds at different nodes will only affect the possible value of ρ_{jkn} , and from Proposition 1, constraints in the form of Eqn. (48) are valid for bilinear term $\sum_{i \in \mathbf{I}_{jkn}^*} \hat{F}_{ijk} = R_{jk} \sum_{i \in \mathbf{I}_{jkn}^*} F_{ij}$ with nontrivial upper bound v_{jk} regardless of the value of ρ_{jkn} .

Algorithm 2. Generating constraints at nodes

Inputs: $n, v_{jk}, F_{ij}^*, R_{jk}^*, \hat{F}_{ijk}^*$, and $\hat{\mathbf{C}}_n$ $c_1 = \text{True}, c_2 = \text{False}$ **For** $j \in \mathbf{J}$ **do****For** $k \in \mathbf{K}$ **do** $c_3 = \text{False}$ **For** $i \in \mathbf{I}$ **do****If** $\hat{F}_{ijk}^* \neq F_{ij}^* R_{jk}^*$ **then** $c_1 = \text{False}$ $c_3 = \text{True}$ **Break****End****End** $\mathbf{I}_{jkn}^* = \{i: \hat{F}_{ijk}^* > 0\}$ **If** $c_3 = \text{True}$ **AND** $4 \left(\sum_{i \in \mathbf{I}_{jkn}^*} \hat{F}_{ijk}^* \right)^2 - 4v_{jk}R_{jk}^* \sum_{i \in \mathbf{I}_{jkn}^*} F_{ij}^* > 0$ **then** $\rho_{jkn} = \sum_{i \in \mathbf{I}_{jkn}^*} \hat{F}_{ijk}^* / \sum_{i \in \mathbf{I}_{jkn}^*} F_{ij}^*$ $\hat{\mathbf{C}}_n = \hat{\mathbf{C}}_n \cup \{(j, k, n)\}$ $c_2 = \text{True}$ **End****End****End****Outputs:** $\rho_{jkn}, c_1, c_2, \mathbf{I}_{jkn}^*$, and $\hat{\mathbf{C}}_n$

We present a customized branch-and-cut (B&C) algorithm that integrates Algorithm 2 within a B&C framework in Figure 5. We start with the solution to the relaxation at the root node (M_0^{C-L}) with objective function value Z_0^* . The list of open nodes (node list) contains only the root node, and set $\hat{\mathbf{C}}_0$ is empty. Z_0^* is the initial upper bound on the objective function value (UB), and the initial lower bound on the objective function value (LB) is set to zero since a trivial feasible solution exists with all variables being zero. We select a node n in the node list, read its solution and run Algorithm 2. After running Algorithm 2, if all nonlinear constraints in M^C are satisfied (i.e., $c_1 = \text{True}$), then this solution is a feasible solution to M^C ; if new constraints are generated in Algorithm 2 (i.e., $c_2 = \text{True}$), then we add them to M_n^{C-L} and solve it again. Note that set $\hat{\mathbf{C}}_n$ for Eqn. (48) is updated in Algorithm 2, and constraints in Eqn. (48), once generated, will be included in all later nodes. After updating the solution to M_n^{C-L} , we perform local search to find a feasible solution to M^C , using a local nonlinear solver, with the solution to M_n^{C-L} as the initial point. After the local search, we update LB (if applicable) and then perform branching. Two child nodes are generated through branching, and the relaxations associated with them are solved right after branching. Such relaxations contain all constraints in the parent node (including Eqn. (48)). The details for the implemented node selection rule, local search, prune rule, and branching strategy can be found in Appendix B. The algorithm terminates when (1) the optimality gap, defined as $(1 - LB/UB)$, is within a chosen tolerance (ϵ), or (2) the maximum number of processed nodes ($\hat{\theta}$) has been reached.

Finally, while the B&C algorithm introduced above can be used to solve the pooling problem, we note that, since the constraints in Eqn. (48) are globally valid, it can also be utilized as a preprocessing algorithm to generate valid constraints whose indices are stored in set $\hat{\mathbf{C}}_n$ with n being, essentially, the last node that has been processed.

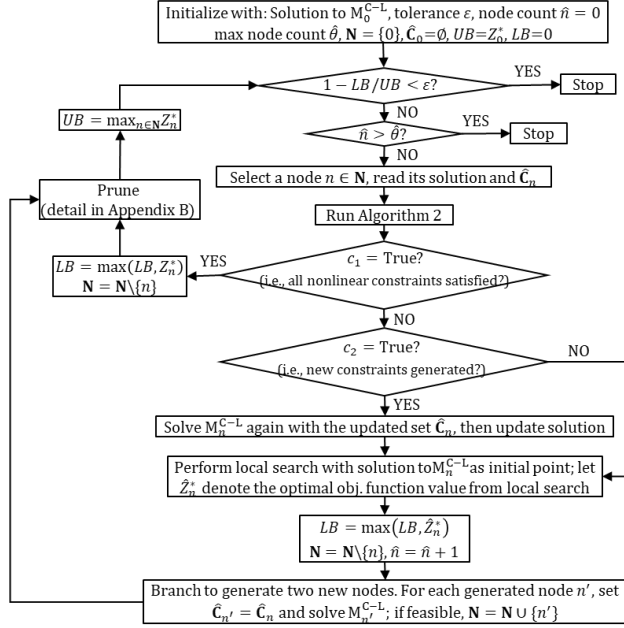


Figure 5. Flowchart of the customized B&C algorithm.

4.2. Methods for model M^{SC}

4.2.1. Generation at the root node

For model M^{SC} , we generate the following constraint at the root node:

$$\rho_{jkm'}^2 \tilde{F}_j + v_{jk} R_{jk} + 2\rho_{jkm'} (\sigma_{jk,1} \bar{F}_{jk} + \sigma_{jk,2} Z_{jk}) \geq 0, \quad (j, k, m') \in \mathbf{C}_m \quad (49)$$

where $\sigma_{jk,1} = (\sqrt{\mu_{jk} v_{jk}} - v_{jk}) / (v_{jk} - \mu_{jk})$, $\sigma_{jk,2} = v_{jk} (\mu_{jk} - \sqrt{\mu_{jk} v_{jk}}) / (v_{jk} - \mu_{jk})$.

We consider model M_m^{SC-L} which contains all constraints in model M^{SC} , except that the nonlinear constraint Eqn. (5) is replaced by Eqn. (39) - (41). M_m^{SC-L} also contains Eqn. (49).

We first present Algorithm 3 that generates constraints based on maximizing constraint violation. At each round of constraint generation, we solve the continuous relaxation of M_m^{SC-L} (in which $Z_{jk} \in [0,1]$), and, similar to Algorithm 1, for each (j, k) pair, we check the sign of $[4(\sigma_{jk,1} \bar{F}_{jk}^* + \sigma_{jk,2} Z_{jk}^*)^2 - 4v_{jk} \tilde{F}_j^* R_{jk}^*]$; if positive, we calculate parameter $\rho_{jkm} = -(\sigma_{jk,1} \bar{F}_{jk}^* + \sigma_{jk,2} Z_{jk}^*) / \tilde{F}_j^*$ and generate Eqn. (49). We repeat until no new constraints are generated or we reach the maximum number of constraint generation rounds.

Algorithm 3. Constraint generation from maximizing violation

Inputs: $c = \text{True}$, $m = 0$, θ , and $\mathbf{C}_m = \emptyset$

While $c = \text{True}$ **AND** $m < \theta$ **do**

$c = \text{False}$

 Solve the continuous relaxation of $M_m^{\text{SC-L}}$.

 Read solution $\tilde{F}_j^*$, R_{jk}^* , $\bar{F}_{jk}^*$, Z_{jk}^*

$\mathbf{C}_{m+1} = \mathbf{C}_m$

For $j \in \mathbf{J}$ **do**

For $k \in \mathbf{K}$ **do**

If $4(\sigma_{jk,1}\bar{F}_{jk}^* + \sigma_{jk,2}Z_{jk}^*)^2 - 4v_{jk}\tilde{F}_j^*R_{jk}^* > 0$ **then**

$\mathbf{C}_{m+1} = \mathbf{C}_{m+1} \cup \{(j, k, m)\}$

$\rho_{jkm} = -(\sigma_{jk,1}\bar{F}_{jk}^* + \sigma_{jk,2}Z_{jk}^*)/\tilde{F}_j^*$

$c = \text{True}$

End

End

End

$m = m + 1$

End

Outputs: ρ_{jkm} and $\mathbf{C}_{m+1}$

Similarly, Algorithm 4 generates constraints by solving the continuous relaxation of $M_m^{\text{SC-L}}$ iteratively, but based on solving the minimum distance problem. After solving the continuous relaxation of $M_m^{\text{SC-L}}$, for each (j, k) pair we check the following two conditions: (1) $\bar{F}_{jk}^* = v_{jk}$, and (2) $\tilde{F}_j^*R_{jk}^* < \bar{F}_{jk}^*$. If both conditions hold, that means the nontrivial upper bound is active and the nonlinear constraint is violated. We calculate ρ_{jkm} by solving the minimum distance problem discussed in section 3.2.2. Note that when $\tilde{F}_jR_{jk} = v_{jk}$ we have $\tilde{F}_j \in [v_{jk}, \gamma_j]$, $R_{jk} \in [v_{jk}/\gamma_j, 1]$. Thus, we calculate ρ_{jkm} as follows:

- (1) If $R_{jk}^* \leq \sqrt{v_{jk}(1 - v_{jk}/\gamma_j)/(\gamma_j - v_{jk})} = \sqrt{v_{jk}/\gamma_j} \leq v_{jk}/\tilde{F}_j^*$, then $\rho_{jkm} = \sqrt{v_{jk}/\gamma_j}$.
- (2) If $R_{jk}^* > \sqrt{v_{jk}/\gamma_j}$, then $\rho_{jkm} = R_{jk}^*$.
- (3) If $\sqrt{v_{jk}/\gamma_j} > v_{jk}/\tilde{F}_j^*$, then $\rho_{jkm} = v_{jk}/\tilde{F}_j^*$.

Algorithm 4. Constraint generation from minimizing distance

Inputs: $c = \text{True}$, $m = 0$, θ , and $\mathbf{C}_m = \emptyset$

While $c = \text{True}$ **AND** $m < \theta$ **do**

$c = \text{False}$

 Solve the continuous relaxation of $M_m^{\text{SC-L}}$.

 Read solution $\tilde{F}_j^*$, R_{jk}^* , $\bar{F}_{jk}^*$, Z_{jk}^*

$\mathbf{C}_{m+1} = \mathbf{C}_m$

For $j \in \mathbf{J}$ **do**

For $k \in \mathbf{K}$ **do**

If $\bar{F}_{jk}^* = v_{jk}$ **AND** $\tilde{F}_j^* R_{jk}^* < \bar{F}_{jk}^*$ **then**

$\mathbf{C}_{m+1} = \mathbf{C}_{m+1} \cup \{(j, k, m)\}$

If $R_{jk}^* \leq \sqrt{v_{jk}/\gamma_j} \leq v_{jk}/\tilde{F}_j^*$ **then**

$\rho_{jkm} = \sqrt{v_{jk}/\gamma_j}$

Else If $R_{jk}^* > \sqrt{v_{jk}/\gamma_j}$ **then**

$\rho_{jkm} = R_{jk}^*$

Else

$\rho_{jkm} = v_{jk}/\tilde{F}_j^*$

End

End

End

End

$m = m + 1$

End

Outputs: ρ_{jkm} and $\mathbf{C}_{m+1}$

4.2.2. Generation using predefined parameters

Since for a given parameter ρ we have one valid constraint, we can generate constraints with a predefined set of values of ρ . Specifically, we have the following:

$$\rho_{jko}^2 \tilde{F}_j + v_{jk} R_{jk} + 2\rho_{jko}(\sigma_{jk,1} \bar{F}_{jk} + \sigma_{jk,2} Z_{jk}) \geq 0, \quad j, k, o \in \mathbf{O} \quad (50)$$

where $\mathbf{O} = \{0, 1, \dots\}$ is the index of constraints for a given (j, k) pair, and ρ_{jko} is a predefined parameter. Eqn. (50) is generated without solving any optimization problem. Recall that for a given (j, k) pair, when $\bar{F}_{jk} = v_{jk}$, Eqn. (50) is tangent to $\tilde{F}_j R_{jk} = v_{jk}$ at the point corresponding to $\tilde{F}_j = v_{jk}/\rho$, $R_{jk} = \rho$. Note that when $\tilde{F}_j R_{jk} = v_{jk}$, we have $R_{jk} \in [v_{jk}/\gamma_j, 1]$, thus one straightforward way to define ρ_{jko} is the following:

$$\rho_{jko} = v_{jk}/\gamma_j + o(1 - v_{jk}/\gamma_j)/|\mathbf{O}|, \quad j, k \quad (51)$$

Eqn. (51) generates values of ρ_{jko} evenly distributed in $[v_{jk}/\gamma_j, 1]$. We show an illustrative graph for the points of tangency on bilinear curve generated from such ρ_{jko} in Figure 6.

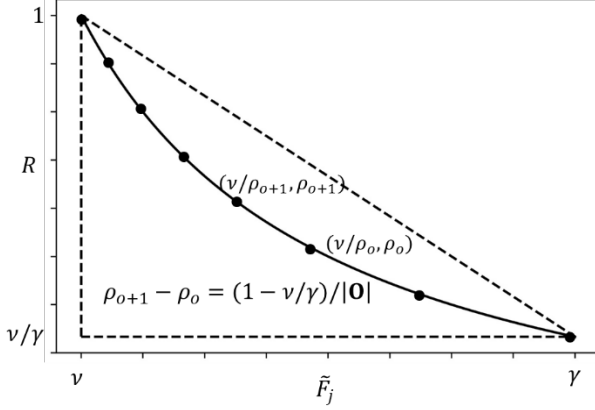


Figure 6. Illustrative graph for points $(v_{jk}/\rho_{jko}, \rho_{jko})$ on curve $\tilde{F}_j R_{jk} = v_{jk}$ with ρ_{jko} generated from Eqn. (51) with $|\mathbf{O}| = 8$ (indices j and k are dropped in the graph for simplicity).

5. Computational results

In this section, we present computational results for models employing the proposed constraint generation methods. Computational experiments are conducted on a Windows 10 machine with Intel Core i5 at 2.70 GHz and 8 GB of RAM. Models are coded in GAMS 30.3 and solved using the global optimization solver BARON 19.12.7. We also provide computational results obtained using Gurobi 9.0 in Appendix C. For all runs, CPU time limit is set at 300 seconds and the tolerance for relative optimality gap is set at 0.01%. Instances are modified from the 90 randomly generated instances in D’Ambrosio et al. (D’Ambrosio, Linderoth, and Luedtke 2011), which are included in QPLIB, a library of quadratic programming instances (Furini et al. 2019). The 90 instances contain 15 streams, 5 – 10 pools, 10 products, and 1 – 4 properties. All parameters defined in M^C are taken from those 90 instances. There are several parameters defined in those 90 instances that are not defined in this paper, we do not consider those parameters. For M^{SC} , most parameters are defined in those 90 instances, except for the lower bound on the positive flow μ_{jk} . We set μ_{jk} to a fraction of the pipeline capacity v_{jk} . More details can be found in the supporting material.

5.1. Model M^C

We test the proposed constraints generated using different methods, as discussed in section 4. Specifically, we consider the following variants of M^C :

- 1) M_R^C : model M^C with Eqn. (42) generated iteratively at the root node; $\mathbf{C}_m$ in Eqn. (42) is obtained by running Algorithm 1 with $\theta = 10$.
- 2) $M_{B\&C}^C$: model M^C with Eqn. (48) generated using the B&C algorithm; $\hat{\mathbf{C}}_n$ in Eqn. (48) is obtained by running the algorithm shown in Figure 5 with $\hat{\theta} = 10$.

- 3) M_H^C : model M^C with Eqn. (42) and Eqn. (48) generated from a hybrid approach; we first obtain C_m for Eqn. (42) by running Algorithm 1 with $\theta = 5$ and then obtain $\hat{C}_n$ for Eqn. (48) by running the B&C algorithm with $\hat{\theta} = 5$ (all M_n^{C-L} include previously generated Eqn. (42)).

We show the performance profiles for the instances that satisfy the following two criteria: (1) instances are solved to global optimality by at least one of the models within 300 seconds; and (2) instances are not solved by the slowest model for that instance within ten seconds. The performance profile shown in Figure 7 contains 49 instances that satisfy such criteria, where the horizontal axis is the factor for performance ratio (which is defined as the solution time for a model to solve an instance over the shortest solution time among all models for the same instance), and the vertical axis is the fraction of instances (Dolan and Moré 2002). We note that certain proposed methods, notably M_R^C , bring substantial computational improvements. Constraint generation for M_R^C , $M_{B\&C}^C$, and M_H^C typically takes 10 - 20 seconds and generates around 100 constraints (time for constraint generation is also counted towards the 300 seconds time limit).

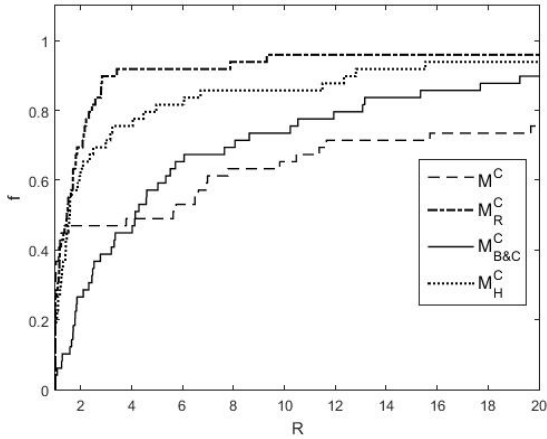


Figure 7. Performance profile for M^C and its variants solved with BARON.

There are several tested instances that are not solved by BARON in 300 seconds with the original model M^C . We solve seven instances among them with the B&C algorithm shown in Figure 5 with $\hat{\theta} = 100$. To demonstrate the effectiveness of the proposed constraints, we also solve the same seven instances with a B&B algorithm which is similar to the B&C algorithm but does not include the constraint generation part (flowchart can be found in Appendix D). Table 2 shows CPU time and optimality gap $(1 - LB/UB)$ after 100 nodes have been processed for both B&B and B&C algorithms. We also show the optimality gap calculated from the upper and lower bounds on the objective function value reported by BARON after 300 seconds.

Table 2. Solution statistics for B&B and B&C algorithms over select instances

Instance	B&B		B&C		BARON
	Time(s)	Gap	Time(s)	Gap	Gap
1*	75.5	0	30.2	0	1.98%
2	136.6	0.53%	167.5	0.33%	2.34%
3	128.6	1.08%	149.6	1.03%	2.13%
4	132.5	1.63%	142.6	1.12%	1.82%
5	205.5	0.94%	210.2	0.66%	1.60%
6	205.9	2.37%	206.5	1.94%	2.80%
7	200.8	3.16%	200.3	0.52%	0.69%

* Instance 1 is solved by both B&B and B&C algorithm within 100 nodes, and we show its solution time.

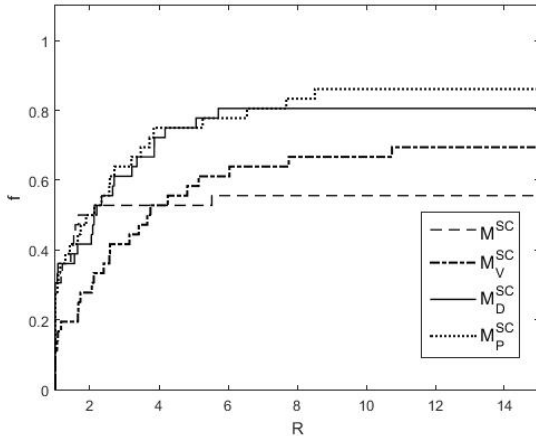
We note that for all seven instances after 100 nodes, we obtain smaller optimality gap from B&C algorithm compared to B&B algorithm.

5.2. Model M^{SC}

We consider the following variants of M^{SC} :

- 1) M_V^{SC} : model M^{SC} with Eqn. (49) generated using Algorithm 3 with $\theta = 10$.
- 2) M_D^{SC} : model M^{SC} with Eqn. (49) generated using Algorithm 4 with $\theta = 10$.
- 3) M_P^{SC} : M^{SC} with Eqn. (50) expressed for predefined parameter ρ_{jko} calculated from Eqn. (51) with $|\mathbf{O}| = 10$.

We again tested the above three models over 90 modified instances and show, in Figure 8, the performance profile generated from the 36 instances satisfying the two aforementioned criteria.

**Figure 8.** Performance profile for M^{SC} and its variants solved with BARON: profit maximization.

We also test our methods for M^{SC} using an objective function where we minimize cost and the penalty for unmet demand, defined in (15). Starting from over 90 modified instances, Figure 9 shows

the performance profile based on the 35 instances satisfying the aforementioned criteria. It is shown that the proposed methods reduce computational requirements.

For M^{SC} , we observe that M_D^{SC} performs well on both tested objectives, which indicates that solving the minimizing distance problem may be a promising approach to generate the proposed constraints. For both objectives, M_V^{SC} and M_D^{SC} typically contain around 100 additional constraints generated in around 10 seconds, and M_P^{SC} contains around 500 - 1000 additional constraints.

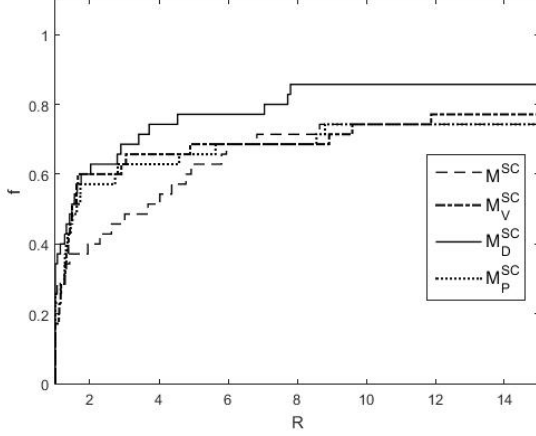


Figure 9. Performance profile for M^{SC} and its variants solved with BARON: cost minimization.

5.3. Other formulation

We also test our methods on models based on another pooling formulation, known as the pq-formulation (Tawarmalani and Sahinidis 2002), in which we have a nonnegative continuous variable $q_{ij} \in [0,1]$ for the proportion of stream i within the total outlet flow from pool j , and the following nonlinear constraint:

$$\hat{F}_{ijk} = q_{ij} \bar{F}_{jk}, \quad i, j, k \quad (52)$$

Summing over index k for Eqn. (52), we obtain:

$$\sum_k \hat{F}_{ijk} = q_{ij} \sum_k \bar{F}_{jk}, \quad i, j, k \quad (53)$$

Note that the LHS of Eqn. (53) is upper bounded by the pipeline capacity between stream i and pool j , which can be a nontrivial upper bound since the RHS of Eqn. (53) is bounded by the capacity of pool j . The model based on pq-formulation contains only continuous variables, and it is referred to as M^{C-PQ} . We consider the following variants of M^{C-PQ} :

- 1) M_R^{C-PQ} : model M^{C-PQ} with constraints similar to those in Eqn. (42), generated at the root node iteratively through a procedure similar to Algorithm 1 with $\theta = 10$.
- 2) M_P^{C-PQ} : model M^{C-PQ} with constraints similar to those in Eqn. (50) generated using pre-determined parameter ρ values calculated from an equation similar to Eqn. (51) with $|\mathbf{O}| = 10$.

We show a performance profile containing 33 instances in Figure 10. For model M^{C-PQ} we observe improvement with adding the proposed constraints. The number of constraints generated and the time needed to generate them are similar to previous models.

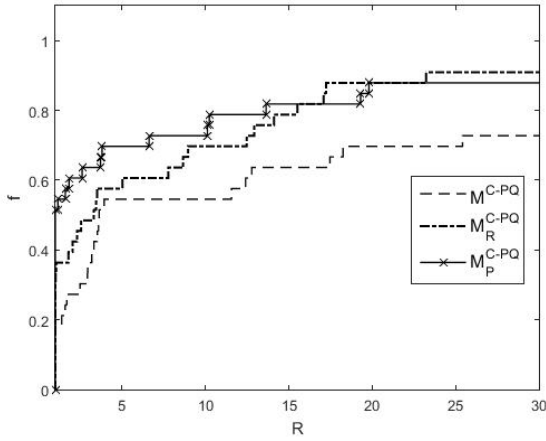


Figure 10. Performance profile for M^{C-PQ} and its variants solved with BARON: profit maximization.

6. Conclusion

We derived a family of strong valid linear constraints for bilinear terms with nontrivial bounds. The proposed constraints are valid for the case where binary and semi-continuous variables are involved. We proposed different methods for generating strong constraints from the family, including generation based on constraint violation maximization and solving the minimum distance problem. We tested the generated constraints on the pooling problem. Computational results demonstrate the effectiveness of the proposed methods in terms of reducing the optimality gap and computational time.

Acknowledgement

The authors acknowledge financial support from the National Science Foundation under grant CBET-2028960.

Appendix A. Solving the minimum distance problem

Consider the following optimization problem:

$$\min_{\rho} \left\{ \frac{1}{x^U - x^L} \left| \frac{w^U}{\rho} - x^* \right| + \frac{1}{y^U - y^L} |\rho - y^*| : \rho \in [w^U/x^U, y^U] \right\} \quad (36)$$

where $x^* y^* < v$. Note that the above optimization problem is solved when the nontrivial upper bound is active, in such case we have $x^* \in [w^U/y^U, x^U]$ and $y^* \in [w^U/x^U, y^U]$. We claim that the solution to the above problem is the following:

- (1) If $y^* \leq \sqrt{w^U(y^U - y^L)/(x^U - x^L)} \leq w^U/x^*$, then $\rho = \sqrt{w^U(y^U - y^L)/(x^U - x^L)}$.
- (2) If $\sqrt{w^U(y^U - y^L)/(x^U - x^L)} < y^*$, then $\rho = y^*$.
- (3) If $\sqrt{w^U(y^U - y^L)/(x^U - x^L)} > w^U/x^*$, then $\rho = w^U/x^*$.

Proof We discuss the above three cases separately.

(1) When $y^* \leq \rho = \sqrt{v(y^U - y^L)/(x^U - x^L)} \leq v/x^*$, we check the sign of $(\frac{w^U}{\rho} - x^*)$ and $(\rho - y^*)$; we have $w^U/\rho - x^* \geq w^U/(w^U/x^*) - x^* \geq x^* - x^* \geq 0$ and $\rho - y^* \geq 0$, thus the above optimization problem is equivalent to:

$$\min_{\rho} \left\{ \frac{1}{x^U - x^L} \left(\frac{w^U}{\rho} - x^* \right) + \frac{1}{y^U - y^L} (\rho - y^*) : \rho \in [w^U/x^U, y^U] \right\}$$

which, after dropping constant terms, becomes:

$$\min_{\rho} \left\{ \frac{1}{x^U - x^L} \cdot \frac{w^U}{\rho} + \frac{1}{y^U - y^L} \cdot \rho : \rho \in [w^U/x^U, y^U] \right\} \quad (54)$$

Furthermore, since $\rho > 0$ we have the following valid inequality:

$$\left(\sqrt{\frac{1}{x^U - x^L} \cdot \frac{w^U}{\rho}} - \sqrt{\frac{1}{y^U - y^L} \cdot w^U} \right)^2 \geq 0$$

which is equivalent to:

$$\frac{1}{x^U - x^L} \cdot \frac{w^U}{\rho} - 2 \sqrt{\frac{w^U}{(x^U - x^L)(y^U - y^L)}} + \frac{1}{y^U - y^L} \cdot w^U \geq 0$$

and thus we have:

$$\frac{1}{x^U - x^L} \cdot \frac{w^U}{\rho} + \frac{1}{y^U - y^L} \cdot w^U \geq 2 \sqrt{\frac{w^U}{(x^U - x^L)(y^U - y^L)}} \quad (55)$$

Eqn. (55) holds as equality when $\rho = \sqrt{w^U(y^U - y^L)/(x^U - x^L)}$ and by construction such ρ is in the range of $[w^U/x^U, y^U]$ (since $w^U/x^U \leq y^* \leq \rho$, and $\rho \leq w^U/x^* \leq w^U/(w^U/y^U) \leq y^U$). Thus, $\rho = \sqrt{w^U(y^U - y^L)/(x^U - x^L)}$ is the solution to (36) when $y^* \leq \sqrt{w^U(y^U - y^L)/(x^U - x^L)} \leq w^U/x^*$.

(2) We first assume that the optimal solution to (36) is $\rho^* < y^*$. If that is the case, we also have $w^U/\rho^* > w^U/y^* > x^*$ since $x^*y^* < w^U$. It follows that such ρ^* is not an optimal solution to (36) since there exists $\varepsilon > 0$ such that $(\rho^* + \varepsilon)$ leads to smaller value for both $|w^U/\rho^* - x^*|$ and $|\rho^* - y^*|$.

We next assume that the optimal solution to (36) is $\rho^* > y^*$. If that is the case, we first note that $w^U/\rho^* > x^*$ should hold since otherwise there exists $\varepsilon > 0$ such that $(\rho^* - \varepsilon)$ leads to the objective function value $(x^* - w^U/(\rho^* - \varepsilon))/(x^U - x^L) + (\rho^* - y^* - \varepsilon)/(y^U - y^L)$, which is smaller than the previous objective function value $(x^* - w^U/(\rho^* - \varepsilon))/(x^U - x^L) + (\rho^* - y^*)/(y^U - y^L)$. Now, since $\rho^* > y^*$ and $w^U/\rho^* > x^*$, we again have the optimization problem defined in (36) with $\rho^* = \sqrt{w^U(y^U - y^L)/(x^U - x^L)} < y^*$, which contradicts with $\rho^* > y^*$. Thus, the optimal solution can only be $\rho^* = y^*$.

(3) We first assume the optimal solution to (36) is $\rho^* > w^U/x^*$. If that is the case, we also have $\rho^* > y^*$ since $x^*y^* < w^U$. It follows that such ρ^* is not an optimal solution to (36) since there exists $\varepsilon > 0$ such that $(\rho^* + \varepsilon)$ leads to smaller value for both $|w^U/\rho^* - x^*|$ and $|\rho^* - y^*|$.

We next assume the optimal solution to (36) is $\rho^* < w^U/x^*$. If that is the case, we first note that $\rho^* > y^*$ should hold since otherwise there exists $\varepsilon > 0$ such that $(\rho^* + \varepsilon)$ leads to the objective function value $(x^* - w^U/(\rho^* - \varepsilon))/(x^U - x^L) + (\rho^* - y^* - \varepsilon)/(y^U - y^L)$, which is smaller than the previous objective function value $(x^* - w^U/(\rho^* - \varepsilon))/(x^U - x^L) + (\rho^* - y^*)/(y^U - y^L)$. Now, since $\rho^* < w^U/x^*$ and $\rho^* > y^*$, we again have the optimization problem defined in (36) with $\rho^* = \sqrt{w^U(y^U - y^L)/(x^U - x^L)} > w^U/x^*$, which contradicts with $\rho^* < w^U/x^*$. Thus, the optimal solution can only be $\rho^* = w^U/x^*$. ■

Appendix B. Details of B&C algorithm

Node selection: we select the node n with the maximum objective function value: $n = \arg \max_n Z_n^*$. If there are multiple nodes with the same maximum objective function value, we select the node with the smallest index.

Local search at node n : we solve M^C using CONOPT, with the initial point being the solution to M_n^{C-L} .

Prune rule: we remove all nodes with $Z_n^* < LB$ from the node list.

Branching strategy at node n : we branch on variable R_{jk} only (note that for M^C , branching only on either R_{jk} or F_{ij} can guarantee ε -optimality, see Epperly and Pistikopoulos (Epperly and Pistikopoulos 1997) for details). We first identify the (i, j, k) combination that corresponds to the most violated nonlinear constraint: $(i, j, k) = \arg \max_{i', j', k'} |\hat{F}_{i'j'k'}^* - F_{i'j'k'}^* R_{j'k'}^*|$, where

$\hat{F}_{i'j'k'}^*$, $F_{i'j'}^*$, and $R_{j'k'}^*$ are obtained from solving M_n^{C-L} . Once the specific (i, j, k) is identified, we evaluate the following equation: $\delta_{jk} = |R_{jk}^* - (R_{jkn}^U - R_{jkn}^L)/2|$. Parameter δ_{jk} aims to quantify the distance between R_{jk}^* and the midpoint for its range. We branch on R_{jk} corresponds to the smallest δ_{jk} . For branching, the break point is at the variable value in the solution to M_n^{C-L} ; in other words, at node n , the range for R_{jk} in the two resulting nodes are $[R_{jkn}^L, R_{jk}^*]$ and $[R_{jk}^*, R_{jkn}^U]$, respectively.

Appendix C. Computational results for Gurobi

We provide results obtained using Gurobi 9.0 for select models and their variants below. Models are again coded in GAMS 30.3 and Gurobi is called from GAMS. The Gurobi input parameter NonConvex is set to 2 so that models “are solved by ... applying spatial branching”, according to Gurobi’s online documentation.

Tested instances are again modified from D’Ambrosio et al. (D’Ambrosio, Linderoth, and Luedtke 2011), with detailed files included in supporting material. Instances that satisfy the two criteria described in section 5.1 are included in the performance profiles below. All models are defined in section 5, except for model M_P^C , which is a variant of model M^C with constraints similar to those in Eqn. (50) generated using pre-determined parameter ρ values calculated from an equation similar to Eqn. (51) with $|\mathbf{O}| = 10$.

Performance profiles in three figures below contain 27, 10, and 22 instances, respectively. Overall, we see that the constraints proposed in this paper can improve the performance of Gurobi as well.

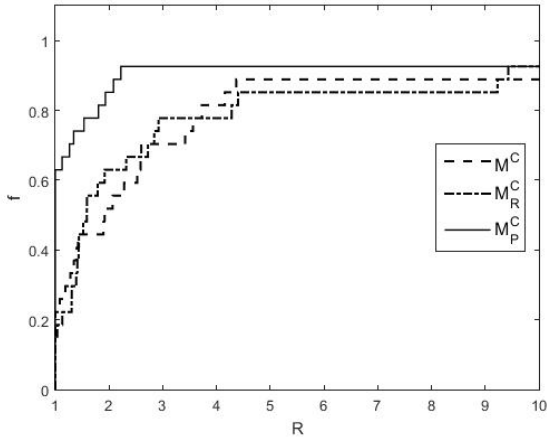


Figure 11. Performance profile for M^C and its variants solved with Gurobi.

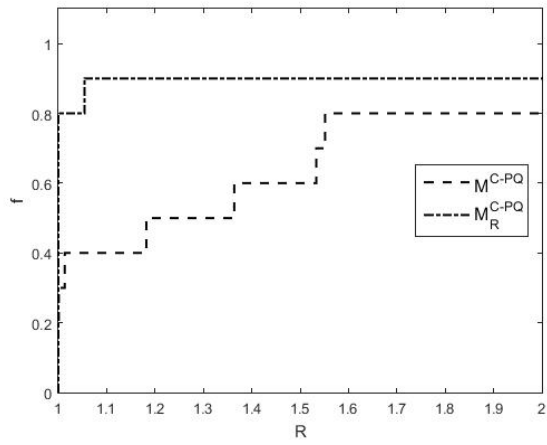


Figure 12. Performance profile for M^{C-PQ} and its variant solved with Gurobi: profit maximization.

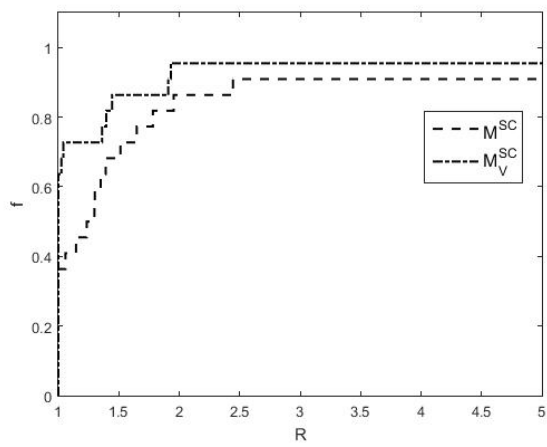


Figure 13. Performance profile for M^{SC} and its variant solved with Gurobi: profit maximization.

Appendix D. B&B algorithm

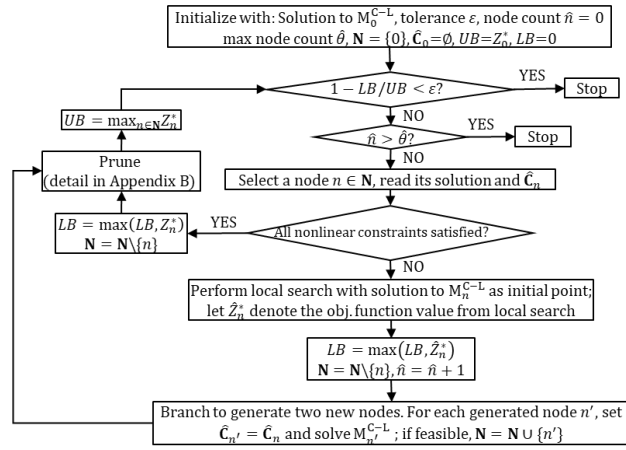


Figure 14. Flowchart for the customized B&B algorithm.

References

- Alfaki, Mohammed, and Dag Haugland. 2013. "Strong Formulations for the Pooling Problem." In *Journal of Global Optimization*, 56:897–916. Springer. <https://doi.org/10.1007/s10898-012-9875-6>.
- Anstreicher, Kurt M., Samuel Burer, and Kyungchan Park. 2020. "Convex Hull Representations for Bounded Products of Variables." *ArXiv*, April. <http://arxiv.org/abs/2004.07233>.
- Audet, Charles, Jack Brimberg, Pierre Hansen, Sébastien Le Digabel, and Nenad Mladenović. 2004. "Pooling Problem: Alternate Formulations and Solution Methods." *Management Science* 50 (6): 761–76. <https://doi.org/10.1287/mnsc.1030.0207>.
- Bagajewicz, Miguel. 2000. "A Review of Recent Design Procedures for Water Networks in Refineries and Process Plants." *Computers & Chemical Engineering* 24 (9–10): 2093–2113. [https://doi.org/10.1016/S0098-1354\(00\)00579-2](https://doi.org/10.1016/S0098-1354(00)00579-2).
- Belotti, Pietro, Andrew J. Miller, and Mahdi Namazifar. 2010. "Valid Inequalities and Convex Hulls for Multilinear Functions." *Electronic Notes in Discrete Mathematics* 36 (C): 805–12. <https://doi.org/10.1016/j.endm.2010.05.102>.
- Belotti, Pietro, Andrew J. Miller, and Mahdi Namazifar. 2011. "Linear Inequalities for Bounded Products of Variables." *SIAG/OPT Views-and-News* 22 (1): 1–8.
- Ben-Tal, Aharon, Gideon Eiger, and Vladimir Gershovitz. 1994. "Global Minimization by Reducing the Duality Gap." *Mathematical Programming* 63 (1–3): 193–212. <https://doi.org/10.1007/BF01582066>.
- Blom, Michelle L., Christina N. Burt, Adrian R. Pearce, and Peter J. Stuckey. 2014. "A Decomposition-Based Heuristic for Collaborative Scheduling in a Network of Open-Pit Mines." *INFORMS Journal on Computing* 26 (4): 658–76. <https://doi.org/10.1287/ijoc.2013.0590>.
- Blom, Michelle L., Adrian R. Pearce, and Peter J. Stuckey. 2016. "A Decomposition-Based Algorithm for the Scheduling of Open-Pit Networks Over Multiple Time Periods." *Management Science* 62 (10): 3059–84. <https://doi.org/10.1287/mnsc.2015.2284>.
- Boland, Natashaia, Thomas Kalinowski, and Fabian Rigterink. 2016. "New Multi-Commodity Flow Formulations for the Pooling Problem." *Journal of Global Optimization* 66 (4): 669–710. <https://doi.org/10.1007/s10898-016-0404-x>.
- Boland, Natashaia, Thomas Kalinowski, Fabian Rigterink, and Martin Savelsbergh. 2016. "A Special Case of the Generalized Pooling Problem Arising in the Mining Industry." http://www.optimization-online.org/DB_FILE/2015/07/5025.pdf.
- Castro, Pedro M. 2015. "Tightening Piecewise McCormick Relaxations for Bilinear Problems." *Computers and Chemical Engineering* 72 (January): 300–311. <https://doi.org/10.1016/j.compchemeng.2014.03.025>.
- Chen, Yifu, and Christos T. Maravelias. 2020. "Preprocessing Algorithm and Tightening Constraints for Multiperiod Blend Scheduling: Cost Minimization." *Journal of Global Optimization* 77 (3): 603–25. <https://doi.org/10.1007/s10898-020-00882-3>.
- D'Ambrosio, Claudia, Jeff Linderoth, and James Luedtke. 2011. "Valid Inequalities for the Pooling Problem with Binary Variables." In *Integer Programming and Combinatorial Optimization*, edited by Oktay Günlük and Gerhard J. Woeginger, 117–29. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Dey, Santanu S., and Akshay Gupte. 2015. "Analysis of MILP Techniques for the Pooling Problem." *Operations Research* 63 (2): 412–27. <https://doi.org/10.1287/opre.2015.1357>.
- Dolan, Elizabeth D., and Jorge J. Moré. 2002. "Benchmarking Optimization Software with Performance Profiles." *Mathematical Programming, Series B* 91 (2): 201–13.

- <https://doi.org/10.1007/s101070100263>.
- Epperly, Thomas G.W., and Efstratios N. Pistikopoulos. 1997. "A Reduced Space Branch and Bound Algorithm for Global Optimization." *Journal of Global Optimization* 11 (3): 287–311. <https://doi.org/10.1023/A:1008212418949>.
- Furini, Fabio, Emiliano Traversi, Pietro Belotti, Antonio Frangioni, Ambros Gleixner, Nick Gould, Leo Liberti, et al. 2019. "QPLIB: A Library of Quadratic Programming Instances." *Mathematical Programming Computation* 11 (2): 237–65. <https://doi.org/10.1007/s12532-018-0147-4>.
- Gounaris, Chrysanthos E., Ruth Misener, and Christodoulos A. Floudas. 2009. "Computational Comparison of Piecewise–Linear Relaxations for Pooling Problems." *Industrial & Engineering Chemistry Research* 48 (12): 5742–66. <https://doi.org/10.1021/ie8016048>.
- Günlük, Oktay, and Jeff Linderoth. 2010. "Perspective Reformulations of Mixed Integer Nonlinear Programs with Indicator Variables." *Mathematical Programming* 124 (1–2): 183–205. <https://doi.org/10.1007/s10107-010-0360-z>.
- Gupte, Akshay, Shabbir Ahmed, Santanu S. Dey, and Myun Seok Cheon. 2017. "Relaxations and Discretizations for the Pooling Problem." *Journal of Global Optimization* 67 (3): 631–69. <https://doi.org/10.1007/s10898-016-0434-4>.
- Haverly, C. A. 1978. "Studies of the Behavior of Recursion for the Pooling Problem." *ACM SIGMAP Bulletin*, no. 25 (December): 19–28. <https://doi.org/10.1145/1111237.1111238>.
- Jeżowski, Jacek. 2010. "Review of Water Network Design Methods with Literature Annotations." *Industrial & Engineering Chemistry Research* 49 (10): 4475–4516. <https://doi.org/10.1021/ie901632w>.
- Kimizuka, Masaki, Sunyoung Kim, and Makoto Yamashita. 2019. "Solving Pooling Problems with Time Discretization by LP and SOCP Relaxations and Rescheduling Methods." *Journal of Global Optimization* 75 (3): 631–54. <https://doi.org/10.1007/s10898-019-00795-w>.
- Kolodziej, Scott P., Pedro M. Castro, and Ignacio E. Grossmann. 2013. "Global Optimization of Bilinear Programs with a Multiparametric Disaggregation Technique." *Journal of Global Optimization* 57 (4): 1039–63. <https://doi.org/10.1007/s10898-012-0022-1>.
- Luedtke, James, Claudia D'Ambrosio, Jeff Linderoth, and Jonas Schweiger. 2020. "Strong Convex Nonlinear Relaxations of the Pooling Problem." *SIAM Journal on Optimization* 30 (2): 1582–1609. <https://doi.org/10.1137/18M1174374>.
- McCormick, Garth P. 1976. "Computability of Global Solutions to Factorable Nonconvex Programs: Part I — Convex Underestimating Problems." *Mathematical Programming* 10 (1): 147–75. <https://doi.org/10.1007/BF01580665>.
- Meyer, Clifford A., and Christodoulos A. Floudas. 2006. "Global Optimization of a Combinatorially Complex Generalized Pooling Problem." *AIChE Journal* 52 (3): 1027–37. <https://doi.org/10.1002/aic.10717>.
- Misener, Ruth, and Christodoulos A. Floudas. 2012. "Global Optimization of Mixed-Integer Quadratically-Constrained Quadratic Programs (MIQCQP) through Piecewise-Linear and Edge-Concave Relaxations." *Mathematical Programming* 136 (1): 155–82. <https://doi.org/10.1007/s10107-012-0555-6>.
- Misener, Ruth, and Christodoulos A. Floudas. 2009. "Advances for the Pooling Problem: Modeling, Global Optimization, and Computational Studies." *Appl. Comput. Math* 8 (1): 3–22. https://www.researchgate.net/profile/Ruth_Misener/publication/242290955_Advances_for_the_pooling_problem_Modeling_global_optimization_and_computational_studies_Survey/links/0046352e7d1dfb40f000000/Advances-for-the-pooling-problem-Modeling-global-optimiza.
- Misener, Ruth, Chrysanthos E. Gounaris, and Christodoulos A. Floudas. 2010. "Mathematical

- Modeling and Global Optimization of Large-Scale Extended Pooling Problems with the (EPA) Complex Emissions Constraints." *Computers & Chemical Engineering* 34 (9): 1432–56. <https://doi.org/10.1016/J.COMPCHEMENG.2010.02.014>.
- Papageorgiou, Dimitri J., Alejandro Toriello, George L. Nemhauser, and Martin W. P. Savelsbergh. 2012. "Fixed-Charge Transportation with Product Blending." *Transportation Science* 46 (2): 281–95. <https://doi.org/10.1287/trsc.1110.0381>.
- Sawaya, Nicolas W., and Ignacio E. Grossmann. 2005. "A Cutting Plane Method for Solving Linear Generalized Disjunctive Programming Problems." *Computers and Chemical Engineering* 29 (9): 1891–1913. <https://doi.org/10.1016/j.compchemeng.2005.04.004>.
- Sherali, Hanif D., and Warren P. Adams. 1999. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Vol. 31. Nonconvex Optimization and Its Applications. Boston, MA: Springer US. <https://doi.org/10.1007/978-1-4757-4388-3>.
- Stubbs, Robert A., and Sanjay Mehrotra. 1999. "A Branch-and-Cut Method for 0-1 Mixed Convex Programming." *Mathematical Programming, Series B* 86 (3): 515–32. <https://doi.org/10.1007/s101070050103>.
- Tawarmalani, Mohit., and Nikolaos V. Sahinidis. 2002. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming : Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers.
- Wicaksono, Danan Suryo, and I. A. Karimi. 2008. "Piecewise MILP Under- and Overestimators for Global Optimization of Bilinear Programs." *AIChE Journal* 54 (4): 991–1008. <https://doi.org/10.1002/aic.11425>.