LP-based preprocessing algorithm and tightening constraints for multiperiod blend scheduling problems

Yingkai Song† and Christos T. Maravelias*,†,‡

†*Andlinger Center for Energy and the Environment, Princeton University, Princeton, NJ 08544, USA*

‡*Department of Chemical and Biological Engineering, Princeton University, Princeton, NJ 08544, USA*

E-mail: maravelias@princeton.edu

Abstract

The multiperiod blend scheduling problem (MBSP) has a wide variety of engineering applications, and is typically formulated as a nonconvex mixed-integer nonlinear program (MINLP). Such an MINLP is challenging to solve due to a large number of bilinear terms and binary variables. One prevalent solution method is branch-and-bound, whose efficiency heavily relies on the tightness of the convex relaxation of the MINLP. In this article, we propose new constraints that can be used for tightening such convex relaxation. These constraints are derived from the physical information lost due to relaxation, and require solving linear programs (LPs) during preprocessing. Extensive numerical tests are executed to examine the effectiveness of the proposed methods. The results show that even though hundreds of LPs may be solved during preprocessing, our new methods can significantly reduce the overall computational time, including both the preprocessing and MINLP solver solution time. Further implications are discussed.

1 Introduction

The multiperiod blend scheduling problem (MBSP)[1,2] is a scheduling extension of the (generalized) pooling problem,[3] and considers blending raw materials with different qualities to produce products, meeting property specifications, over a given scheduling horizon. The objective of the MBSP is to find the least cost blending plan (or maximize profit), subject to various constraints such as raw material availability, operational rules, and product demand requirements. The MBSP has a wide variety of engineering applications such as crude or refined oil scheduling,[4–6] mine planning,[7,8] wastewater treatment,[9,10] copper concentrate blending,[11,12] and specialty chemicals manufacturing.[13] While an optimal solution of the MBSP can bring significant economic benefits,[14–16] state-of-the-art methods can only solve instances of modest size. Thus, improved techniques are sought to solve the MBSP of practical interest.

The MBSP is typically formulated as a nonconvex mixed-integer nonlinear program (MINLP), or more specifically, a mixed-integer quadratically constrained program (MIQCP). In such an MINLP, binary variables are employed to model decisions on material transfers, and bilinear terms are employed to model the composition consistency between blenders' inventories and outflows. In terms of time representation, MINLPs for the MBSP are categorized into discrete- and continuous-time formulations; see Fragkogios and Saharidis[4] for a review. In discrete-time formulations, the scheduling horizon is divided into predefined time periods, and all material transfer operations can only start or end at the boundaries of these time periods. On the other hand, continuous-time formulations can yield solutions where an operation can start at any time. As discussed in Floudas and Lin[17], Discrete-time formulations are straightforward to implement and the solutions are easier to be executed in practice, but the problem size increases drastically as the number of time periods increases. On the contrary, continuous-time formulations typically have smaller size, but have weaker LP relaxations (as will be discussed later) and the solutions may not be fully executable in practice. Hybrid models[6,18] have been proposed to combine the strength of both formulations for the MBSP. This article will focus on discrete-time formulations, and our new methods may be extended to address continuous-time formulations.

Branch-and-bound-based[19] deterministic global optimization methods for MINLP require computing upper and lower bounds of the globally optimal objective value. These bounds are progressively refined, until convergence. For a minimization MBSP, the upper bounds can be computed from any feasible solution of the original MINLP, and the lower bounds are typically computed by constructing and solving auxiliary linear programming (LP) relaxation problems. In such LP relaxation problems, the binary variables are relaxed to be continuous variables, and the bilinear terms are replaced by various (piecewise-)linear underestimators and overestimators as will be summarized below. Tightness of the feasible region of such LP can greatly affect the computational performance of branch-and-bound algorithms. A tighter formulation typically yields tighter bounds for the globally optimal objective value, and thus reduces the number of nodes required to be explored to prove global optimality. Even if a tighter feasible region does not lead to tighter objective function, such smaller feasible region may reduce the computational time for solving a single LP relaxation problem. This is beneficial since the relaxation problems are solved multiple times in a branch-and-bound procedure. There have been abundant studies focusing on developing tight relaxation formulations, which will be summarized next.

Since the McCormick envelopes[20] were applied to relax bilinear terms in global optimization, researches have been done to further tighten bilinear term relaxations. Since McCormick envelopes are tighter when tighter variable bounds are available, many studies focused on computing tight variable bounds. In particular, Chen and Maravelias[21] proposed an effective closed-form bound tightening method that is specialized for the MBSP. In addition to variable bound tightening, one class of approaches[22–25] constructs tight piecewise mixed-integer linear programming (MILP) relaxations using piecewise McCormick envelopes. These approaches divide the domain of one variable in each bilinear term into multiple smaller subintervals in various different ways, and then McCormick envelopes are constructed on each subinterval and auxiliary binary variables are introduced to model the disjunction among all subintervals. Another class of approaches[2,26–28] for constructing piecewise MILP relaxations is based on multiparametric disaggregation. These approaches firstly discretize one variable in each bilinear term following a predefined accuracy level, and then rigorously bound the truncation error to obtain valid relaxations. Castro et al.[27,29] compared the McCormick-based and multiparametric disaggregation approaches. When bilinear terms have nontrivial upper or (and) lower bounds, several relaxations[30–32] were proposed and shown to be tighter than trivially combining McCormick envelopes and the bounds. For pooling problems, nontrivial upper bounds for bilinear terms can be derived from pipleline capacity between blenders and products.[32] Second order cone programming relaxations and polyhedral relaxations[33,34] for general quadratically constrained programs were adapted to solve pooling problems and MBSPs. Luedtke et al.[35] proposed strong relaxations for a nonconvex set involving bilinear terms in a modified formulation of pooling problems. Lotero et al.[36] formulated the MBSP via generalized disjunctive programming, which has fewer bilinear terms than traditional MINLP formulations when binary variables are fixed.

Another direction for tightening relaxations of nonconvex MINLP is to generate convex tightening constraints that are redundant in the original MINLP, but can cut off feasible region of the relaxed problems. One prevalent approach is the reformulation-linearization technique (RLT).[37,38] RLT firstly generates valid redundant nonlinear contraints involving nonconvex terms, and then these nonlinear constraints are linearized by introducing new variables, one for each nonconvex term. Several RLT constraints[21,36,39] have been proposed to handle bilinear terms and are used for enhancing the MBSP formulations. Besides RLT, tightening constraints may also be generated from specific problem structure and physical intuition. For blending problems, Papageorgiou et al.[40] studied an MILP formulation of fixed-charge transportation problem with product blending, and proposed tightening constraints by exploiting products' property specification. D'Ambrosio et al.[41] proposed tightening constraints for the generalized pooling problem based on the problem's MILP relaxation. It was shown that these constraints dominate some constraints proposed for MILP of a certain form.[42] Chen and Maravelias[1] proposed valid bounds and tightening constraints that are specialized for the MBSP. These constraints involve the so-called dedicated flow variables, which represent the amounts of raw materials dedicated to each product over the scheduling horizon.

In this article, we propose several classes of new constraints for tightening the relaxations of MINLP formulations of the MBSP. These constraints are derived from the physical information lost due to relaxing integer variables and bilinear terms. For example, due to the relaxation of integer variables, certain semicontinuous flow variables become continuous, and a widely-used operational rule for blenders, that a blender cannot be charged and discharged simultaneously, no longer holds. Due to the relaxation of bilinear terms, a blender's outflows may have inconsistent composition with the blender's inventory. Constructing these constraints typically requires solving LP problems, leading to an LP-based preprocessing algorithm. Notably, the new constraints are guaranteed to dominate some previously proposed constraints.[1] While our new tightening constraints are intended to facilitate solving MINLP via branch-and-bound algorithms, they are also useful in tightening bounding in certain decomposition-based global optimization approaches.[28,36,43,44]

We apply our new tightening constraints to two previously-proposed and widely-used discrete-time MINLP formulations for the MBSP, namely, the proportion-based formulation[21,38] and the split-fraction-based formulation.[21,36] We test the effectiveness of our new proposal by solving numerical instances using state-of-the-art general global optimization solver BARON[45] and MIQCP solver GUROBI.[46] The results show that our LP-based preprocessing algorithm typically requires a few seconds to be executed, but the derived constraints significantly reduce the overall computational time (including preprocessing and solver solution time) for solving both the proportion- and split-fraction-based formulations using both solvers.

The remainder of this article is organized as follows. Section 2.1 presents the structure and feature of the MBSP considered in this article. Section 2.2 presents the proportion- and split-fraction-based formulations for the considered MBSP, and our new tightening constraints will be used to enhance these formulations. Section 3 discusses the intuition behind the new constraints through a motivating example. Section 4 formalizes our new constraints, and Section 5 presents numerical tests to examine the new constraints' effectiveness. Section 6 discusses how to extend the new constraints to other types of MBSPs studied in literature. Throughout this article, we use uppercase letters for nonnegative variables, lowercase letters for indices, boldface uppercase letters for sets, and lowercase Greek letters for parameters.

2 Background

2.1 Problem statement

We consider different raw materials, termed *streams*, with certain properties and availability over a given scheduling horizon. These streams are mixed in blenders (with possible initial inventory) assuming a linear mixing rule to produce products, meeting properties specifications and demand requirements. The products have upper and lower bounds for property values, have minimum accumulated demands that gradually increase along the scheduling horizon, and have maximum demands over the whole scheduling horizon. The blenders cannot be charged by stream sources and delivering products simultaneously. Upper bounds of charging rate, and both lower and upper bounds of withdrawing rate for blenders are given. We assume full connections from stream sources to blenders and from blenders to product sinks. Material transfer between blenders is not allowed. The entire scheduling horizon is divided into a number of time periods with equal length, and all material transfer operations can only start or end at the boundaries of each time period. The objective is to find the optimal schedule for transferring and blending materials, to maximize profit, which is defined as the difference between revenue for selling products and various costs such as expense for buying streams and cost for material transfer operations between units. A topology network of the considered MBSP is given in Figure 1. The MBSP employs the following sets, parameters, and variables:

Sets

$s \in \mathbf{S}$ Streams or stream sources

$q \in \mathbf{Q}$ Properties

$b \in \mathbf{B}$ Blenders

$p \in \mathbf{P}$ Products or product sinks

$t \in \mathbf{T}$ Time periods (ordered set): $\{1, 2, ..., |\mathbf{T}|\}$. The end of a time period t will be referred as the time point t. With slight abuse of notation, define time point 0 as the beginning of the first time period.

Parameters

$\delta_p^{\max}$ Maximum demand for product p throughout the scheduling horizon

$\delta_{p,t}^{\min}$ Minimum accumulated demand for product p by time point t

ξ_s Available amount of stream s throughout the scheduling horizon

$\pi_{s,q}$ Quantity of property q of stream s

$\pi_q^{\max}$ $:= \max\{\pi_{s,q} : \forall s \in \mathbf{S}\}$

$\pi_{p,q}^{\mathrm{U}}$ Quantity upper bound of property q of product p

$\pi_{p,q}^{\mathrm{L}}$ Quantity lower bound of property q of product p

α_s Unit cost of stream s

β_p Unit revenue of product p

ω_b Capacity of blender b

$\zeta_{b,p}^{\mathrm{U}}$ Flowrate upper bound from blender b to product sink p

$\zeta_{b,p}^{\mathrm{L}}$ Flowrate lower bound from blender b to product sink p

$\tilde{\zeta}_{s,b}^{\mathrm{U}}$ Flowrate upper bound from stream source s to blender b

$\gamma_{b,p}$ Fixed cost for transferring material from blender b to product sink p

$\tilde{\gamma}_{s,b}$ Fixed cost for transferring material from stream source s to blender b

$\eta_{s,b}$ Initial inventory of stream s in blender b

Nonnegative continuous variables

$I_{s,b,t}^{\mathrm{S}}$ Inventory of stream s in blender b at time point t

$I_{b,t}$ Total inventory of blender b at time point t

$\tilde{F}_{s,b,t}$ Flowrate from stream source s to blender b during time period t

$F_{s,b,p,t}^{\mathrm{S}}$ Flowrate of stream s from blender b to product sink p during time period t

$F_{b,p,t}$ Total flowrate from blender b to product sink p during time period t

Binary variables

$\tilde{X}_{s,b,t}$ 1 if stream source s is charging blender b during time period t

$X_{b,p,t}$ 1 if blender b is delivering product p during time period t

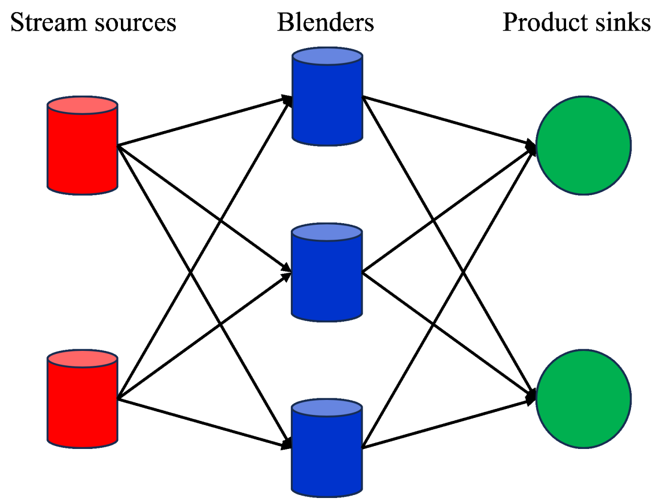


Figure 1: Network of the MBSP

2.2 Base formulations

This subsection presents two widely-used discrete-time MINLP formulations for the MBSP, namely, the proportion-based (P-)formulation[21,38] and the split-fraction-based (S-)formulation.[21,36] These two formulations mainly differ in how the composition consistency between blenders' inventories and outflows is enforced. The P-formulation employs auxiliary variables to represent the proportion of each stream in blenders' inventories, and the outflows are enforced to have the same proportion. Instead, the S-formulation defines the fraction of mixture that is withdrawn from blenders, and all streams are enforced to be withdrawn with the same fraction.

Note that both P- and S-formulations have counterparts which track each property flow instead of each stream. Compared to their property-tracking counterparts, the P- and S-formulations with certain established tightening constraints have at least as tight relaxations, and are in general superior in practice.[36,38,47] However, there is no known theoretical result comparing the relaxations of P- and S-formulations, and no formulation is clearly faster than another. For pooling problems, Cheng and Li[48] established conditions under which the S-formulation's relaxation is not tighter than the P-formulation's relaxation. Whether their conclusions can be extended to the MBSP requires further investigation. The goal of this article is to propose new constraints that can tighten relaxations of both the P- and S-forumulations for the MBSP.

2.2.1 Proportion-based formulation

Besides all variables presented in Section 2.1, the P-formulation introduces variables $C_{s,b,t}$ to denote the proportion of stream s in blender b at time point t. The P-formulation has the following constraints and objective function.

Stream availability constraints:

$$\sum_{b\in\mathbf{B}}\sum_{t\in\mathbf{T}}\tilde{F}_{s,b,t} \le \xi_s, \quad \forall s\in\mathbf{S}. \tag{1}$$

Relating $F_{b,p,t}$ to $X_{b,p,t}$:

$$F_{b,p,t} \le \zeta^{\mathrm{U}}_{b,p}X_{b,p,t}, \quad \forall b\in\mathbf{B}, \forall p\in\mathbf{P}, \forall t\in\mathbf{T}, \tag{2}$$

$$F_{b,p,t} \ge \zeta^{\mathrm{L}}_{b,p}X_{b,p,t}, \quad \forall b\in\mathbf{B}, \forall p\in\mathbf{P}, \forall t\in\mathbf{T}. \tag{3}$$

Relating $\tilde{F}_{s,b,t}$ to $\tilde{X}_{s,b,t}$:

$$\tilde{F}_{s,b,t} \le \tilde{\zeta}^{\mathrm{U}}_{s,b}\tilde{X}_{s,b,t}, \quad \forall s\in\mathbf{S}, \forall b\in\mathbf{B}, \forall t\in\mathbf{T}. \tag{4}$$

Product demand constraints:

$$\sum_{b\in\mathbf{B}}\sum_{t\in\mathbf{T}}F_{b,p,t} \le \delta^{\max}_{p}, \quad \forall p\in\mathbf{P}, \tag{5}$$

$$\sum_{b\in\mathbf{B}}\sum_{t'\le t}F_{b,p,t'} \ge \delta^{\min}_{p,t}, \quad \forall p\in\mathbf{P}, \forall t\in\mathbf{T}. \tag{6}$$

Note that $\delta^{\min}_{p,t}$ is non-decreasing with respect to t.

The operational rule for blenders is enforced by the following constraints:

$$\tilde{X}_{s,b,t} \leq 1 - X_{b,p,t}, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{7}$$

Constraints (7) ensure that during any time period, a blender may have muliple inlet flows or multiple outlet flows, but not both. We note that such operational rule is widely used for MBSPs.[21,28,36,49] However, for specific applications such as crude-oil scheduling,[4] more restrictive operational rule may be applied, e.g., in one time period, a blender can have at most one inlet flow or one outlet flow, but not both.

Mass balance of each stream in blenders:

$$I^{\mathrm{S}}_{s,b,t} = I^{\mathrm{S}}_{s,b,t-1} + \tilde{F}_{s,b,t} - \sum_{p \in \mathbf{P}} F^{\mathrm{S}}_{s,b,p,t}, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall t \in \mathbf{T}. \tag{8}$$

Upper bounds for $I_{b,t}$:

$$I_{b,t} \leq \omega_b, \quad \forall b \in \mathbf{B}, \forall t \in \mathbf{T}. \tag{9}$$

Relating $I^{\mathrm{S}}_{s,b,t}$ to $I_{b,t}$:

$$\sum_{s \in \mathbf{S}} I^{\mathrm{S}}_{s,b,t} = I_{b,t}, \quad \forall b \in \mathbf{B}, \forall t \in \mathbf{T}. \tag{10}$$

Relating $F^{\mathrm{S}}_{s,b,p,t}$ to $F_{b,p,t}$:

$$\sum_{s \in \mathbf{S}} F^{\mathrm{S}}_{s,b,p,t} = F_{b,p,t}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{11}$$

Modeling the proportions of streams in blenders:

$$I^{\mathrm{S}}_{s,b,t} = I_{b,t} C_{s,b,t}, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall t \in \mathbf{T}. \tag{12}$$

Modeling stream proportions in blenders' outflows:

$$F^{\mathrm{S}}_{s,b,p,t} = F_{b,p,t} C_{s,b,t-1}, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{13}$$

Proportion variables sum up to one, modeled as:

$$\sum_{s\in\mathbf{S}} C_{s,b,t} = 1, \quad \forall b \in \mathbf{B}, \forall t \in \mathbf{T}. \tag{14}$$

Note that given (10), (11), (12), and (13), the constraints above are redundant. However, (14) can tighten the relaxation of the P-formulation, where (12) and (13) are relaxed.

Auxiliary constraints that can tighten the relaxation of the P-formulation:

$$I^{\mathrm{S}}_{s,b,t} \leq C_{s,b,t}\omega_b, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall t \in \mathbf{T}, \tag{15}$$

$$F^{\mathrm{S}}_{s,b,p,t} \leq \zeta^{\mathrm{U}}_{b,p} C_{s,b,t-1}, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{16}$$

Property requirements of flows from blenders to products:

$$\sum_{s\in\mathbf{S}} \pi_{s,q} F^{\mathrm{S}}_{s,b,p,t} \leq \pi^{\mathrm{U}}_{p,q} F_{b,p,t}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall q \in \mathbf{Q}, \forall t \in \mathbf{T}, \tag{17}$$

$$\sum_{s\in\mathbf{S}} \pi_{s,q} F^{\mathrm{S}}_{s,b,p,t} \geq \pi^{\mathrm{L}}_{p,q} F_{b,p,t}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall q \in \mathbf{Q}, \forall t \in \mathbf{T}. \tag{18}$$

The following Big-M constraints are firstly used in Chen and Maravelias [21], and are redundant in the P-formulation but can tighten its relaxation. Big-M constraints for properties involving proportion variables:

$$\sum_{s\in\mathbf{S}} C_{s,b,t-1}\pi_{s,q} \leq \pi^{\mathrm{U}}_{p,q} + (\pi^{\max}_{q} - \pi^{\mathrm{U}}_{p,q})(1 - X_{b,p,t}), \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall q \in \mathbf{Q}, \forall t \in \mathbf{T}, \tag{19}$$

$$\sum_{s\in\mathbf{S}} C_{s,b,t-1}\pi_{s,q} \geq \pi^{\mathrm{L}}_{p,q} X_{b,p,t}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall q \in \mathbf{Q}, \forall t \in \mathbf{T}, \tag{20}$$

where $\pi^{\max}_{q} := \max\{\pi_{s,q} : \forall s \in \mathbf{S}\}$. Big-M constraints for properties involving inventories of blenders:

$$\sum_{s\in\mathbf{S}} \pi_{s,q} I^{\mathrm{S}}_{s,b,t-1} \leq \pi^{\mathrm{U}}_{p,q} I_{b,t-1} + \omega_b \pi^{\max}_q (1 - X_{b,p,t}), \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall q \in \mathbf{Q}, \forall t \in \mathbf{T}, \tag{21}$$

$$\sum_{s\in\mathbf{S}} \pi_{s,q} I^{\mathrm{S}}_{s,b,t-1} \geq \pi^{\mathrm{L}}_{p,q} I_{b,t-1} - \omega_b \pi^{\mathrm{L}}_{p,q} (1 - X_{b,p,t}), \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall q \in \mathbf{Q}, \forall t \in \mathbf{T}. \tag{22}$$

The object is to maximize profit, modeled as:

$$\max \sum_{b\in\mathbf{B}} \sum_{p\in\mathbf{P}} \sum_{t\in\mathbf{T}} \left(\beta_p F_{b,p,t} - \gamma_{b,p} X_{b,p,t}\right) - \sum_{s\in\mathbf{S}} \sum_{b\in\mathbf{B}} \sum_{t\in\mathbf{T}} \left(\alpha_s \tilde{F}_{s,b,t} + \tilde{\gamma}_{s,b} \tilde{X}_{s,b,t}\right). \tag{23}$$

Other costs such as storage cost of blenders can also be easily added into the objective function above.

The P-formulation, denoted as M^{P}, consists of Equations (1)–(23).

2.2.2 Split-fraction-based formulation

Instead of using the proportion variables $C_{s,b,t}$, the S-formulation introduces variables $R_{b,p,t}$ to represent the split fraction for inventories in blender b to product sink p during time period t.

All outlet streams of blenders should follow the same split fraction, modeled as:

$$F^{\mathrm{S}}_{s,b,p,t} = I^{\mathrm{S}}_{s,b,t-1} R_{b,p,t}, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{24}$$

Split fraction should be less or equal than one, modeled as:

$$\sum_{p\in\mathbf{P}} R_{b,p,t} \leq 1, \quad \forall b \in \mathbf{B}, \forall t \in \mathbf{T}. \tag{25}$$

Relating $R_{b,p,t}$ to $X_{b,p,t}$:

$$R_{b,p,t} \leq X_{b,p,t}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}, \tag{26}$$

$$R_{b,p,t} \geq \frac{\zeta^{\mathrm{L}}_{b,p}}{\omega_b} X_{b,p,t}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{27}$$

Tightening constraints involving split fractions:

$$F_{b,p,t} \leq \omega_b R_{b,p,t}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{28}$$

The S-formulation, denoted as M^{S}, consists of Equations (1)–(11), (17), (18), (21)–(28).

Note that in both M^{P} and M^{S}, defining the inventory variables $I_{b,t}$ and flow variables $F_{b,p,t}$ is unnecessary, i.e., one may substitute all $I_{b,t}$ and $F_{b,p,t}$ respectively by $\sum_{s\in\mathbf{S}} I^{\mathrm{S}}_{s,b,t}$ and $\sum_{s\in\mathbf{S}} F^{\mathrm{S}}_{s,b,p,t}$, and eliminate Constraints (10) and (11). However, from the authors' experience, this change does not have a significant impact on the solution time of these formulations.

3 Intuition for tightening

We use $\mathrm{RM}^{\mathrm{P(S)}}$ to denote an LP relaxation of the P-(S-)formulation. In $\mathrm{RM}^{\mathrm{P(S)}}$, the binary variables $\tilde{X}_{s,b,t}$ and $X_{b,p,t}$ are relaxed to be continuous variables and the blinear terms in (12), (13), and (24) are relaxed using (piecewise-)linear under- and over-estimators. Since we consider a profit maximization MBSP, the optimal objective function value of $\mathrm{RM}^{\mathrm{P(S)}}$ is an upper bound of the globally optimal objective function value of $\mathrm{M}^{\mathrm{P(S)}}$. We propose new auxiliary constraints to tighten $\mathrm{RM}^{\mathrm{P(S)}}$. These constraints are derived by partially recovering the physical information lost due to relaxation, as discussed next.

Firstly, since all binary variables are relaxed to be continuous, the operational rule that the blenders cannot be charged and discharged simultaneously is not enforced, i.e., for some $(s,b,p,t) \in \mathbf{S}\times\mathbf{B}\times\mathbf{P}\times\mathbf{T}$, $\tilde{F}_{s,b,t}$ and $F_{b,p,t}$ can both be nonzero (see Constraints (2), (4), and (7)). Secondly, observe from Constraints (2) and (3) that, due to the binary variable $X_{b,p,t}$

and the flow bounds $[\zeta^{\mathrm{L}}_{b,p}, \zeta^{\mathrm{U}}_{b,p}]$, the range of the flowrate $\sum_{t\in\mathbf{T}} F_{b,p,t}$ is a union of finitely many intervals, and can be disjoint. However, when $X_{b,p,t}$ can take fractional values, the range for $\sum_{t\in\mathbf{T}} F_{b,p,t}$ is no longer disjoint. Thirdly, consider two products $p, p' \in \mathbf{P}$ that have very different property specifications so that no mixture can be qualified as p and p' simultaneously. As a result, no blenders can deliver p and p' during one time period. However, in $\mathrm{RM}^{\mathrm{P(S)}}$, since the nonlinear constraints are relaxed, there is no composition consistency between the inventory and outflows of each blender. Therefore, it is possible for a blender b and some time period t that, $F_{b,p',t}$ and $F_{b,p,t}$ are both non-zero, despite the fact that the blender's inventory cannot satisfy both products' property specifications. Moreover, if a blender b is about to deliver products during time period t, then the inventory variables $I^{\mathrm{S}}_{s,b,t-1}$ and proportion variables $C_{s,b,t-1}$ may have nontrivial lower and upper bounds, because the mixture in b at time point $t-1$ must satisfy the products' specifications. On the other hand, in $\mathrm{RM}^{\mathrm{P(S)}}$, such bounds for $I^{\mathrm{S}}_{s,b,t-1}$ and $C_{s,b,t-1}$ are not implicitly accounted for. Note that the established tightening constraints (19)–(22) already utilize property specifications to bound $I^{\mathrm{S}}_{s,b,t}$ and $C_{s,b,t}$. The following motivating example shows that constraints developed from the above mentioned physical understanding can indeed help tighten $\mathrm{RM}^{\mathrm{P(S)}}$.

Example 1 Consider an MBSP with two streams, one blender, two products, one property, and three time periods. All parameter values are shown in Table 1.

Table 1: Parameters for the MBSP considered in Example 1

Parameters	Values	Parameters	Values	Parameters	Values
(ξ_{s_1}, ξ_{s_2})	$(10, 10)$	$(\pi^{\mathrm{L}}_{p_1,q_1}, \pi^{\mathrm{L}}_{p_2,q_1})$	$(0.3, 0.5)$	$(\delta^{\min}_{p_1,1}, \delta^{\min}_{p_1,2}, \delta^{\min}_{p_1,3})$	$(0, 0, 3)$
$(\tilde{\xi}^{\mathrm{U}}_{s_1,b_1}, \tilde{\xi}^{\mathrm{U}}_{s_2,b_1})$	$(10, 10)$	$(\pi^{\mathrm{U}}_{p_1,q_1}, \pi^{\mathrm{U}}_{p_2,q_1})$	$(0.4, 0.6)$	$(\delta^{\min}_{p_2,1}, \delta^{\min}_{p_2,2}, \delta^{\min}_{p_2,3})$	$(0, 0, 0)$
$(\pi_{s_1,q_1}, , \pi_{s_2,q_1})$	$(0.2, 0.7)$	$(\zeta^{\mathrm{L}}_{b_1,p_1}, \zeta^{\mathrm{L}}_{b_1,p_2})$	$(4, 0)$		
$(\alpha_{s_1}, \alpha_{s_2})$	$(300, 200)$	$(\zeta^{\mathrm{U}}_{b_1,p_1}, \zeta^{\mathrm{U}}_{b_1,p_2})$	$(6, 3)$		
$(\beta_{p_1}, \beta_{p_2})$	$(800, 2000)$	$(\gamma_{b_1,p_1}, \gamma_{b_1,p_2})$	$(50, 50)$		
$(\eta_{s_1,b_1}, \eta_{s_2,b_1})$	$(0, 0)$	$(\tilde{\gamma}_{s_1,b_1}, \tilde{\gamma}_{s_2,b_1})$	$(50, 50)$		
ω_{b_1}	20	$(\delta^{\max}_{p_1}, \delta^{\max}_{p_2})$	$(10, 10)$		

We construct and solve M^{P} for this problem, and the globally optimal objective function value is 5200.00. Then, we construct RM^{P} by relaxing all binary variables to be continuous, and replacing the nonlinear constraints (12) and (13) by the well-known McCormick envelopes[20] with the following variable bounds for each $(s,p,t) \in \mathbf{S} \times \mathbf{P} \times \mathbf{T}$:

$$
\begin{aligned}
&0 \leq C_{s,b_1,t} \leq 1, \\
&0 \leq I_{b_1,t} \leq \min\{\sum_{s \in \mathbf{S}} \xi_s, \omega_{b_1}\}, \\
&0 \leq F_{b_1,p,t} \leq \min\{\sum_{s \in \mathbf{S}} \xi_s, \omega_{b_1}, \delta_p^{\max}, \zeta_{b_1,p}^{\mathrm{U}}\}.
\end{aligned}
$$

We obtain that the optimal objective value of RM^{P} is 14457.30. Thus, the gap between the globally optimal objective function values of M^{P} and RM^{P} is $14457.30 - 5200.00 = 9257.30$. Now, we propose several tightening constraints for use in RM^{P} to reduce this gap. We use subscripts to indicate additional tightening constraints that are added to RM^{P}. For example, the notation $\mathrm{RM}^{\mathrm{P}}_{(30),(32)}$ below represents a model combining RM^{P} and Constraints (30) and (32).

We observe that in the optimal solution (provided by the solver) of RM^{P}, we have:

$$
\begin{aligned}
&X_{b_1,p_1,t_2} + X_{b_1,p_2,t_2} = 0.7249 + 0.9417 > 1, \\
&X_{b_1,p_1,t_3} + X_{b_1,p_2,t_3} = 0.6989 + 0.9677 > 1.
\end{aligned} \tag{29}
$$

However, it is evident from the parameters $[\pi_{p,q}^{\mathrm{L}}, \pi_{p,q}^{\mathrm{U}}]$ in Table 1 that a mixture in the blender cannot satisfy the property specifications for both p_1 and p_2 simultaneously. Thus, the blender can at most deliver one product during one time period. This leads to the following tightening constraints:

$$
X_{b_1,p_1,t} + X_{b_1,p2,t} \leq 1, \quad \forall t \in \mathbf{T}, \tag{30}
$$

which cut off the optimal solution of RM^{P} as in (29). The optimal objective value of ${\mathrm{RM}^{\mathrm{P}}}_{(30)}$ is found to be 9492.50, which is a significant improvement compared to the previous upper bound 14457.30.

Next, in the optimal solution of ${\rm RM^P}_{(30)}$, we have:

$$
\begin{aligned}
&X_{b_1,p_1,t_2} = X_{b_1,p_2,t_2} = 0.5, \\
&\tilde{X}_{s_1,b_1,t_1} = 0, \quad \text{and} \quad \tilde{X}_{s_2,b_1,t_1} = 0.48.
\end{aligned}
\tag{31}
$$

We observe the following from the streams' and products' property specifications: for product p_1, π_{s_1,q_1} satisfies the upper property bound $\pi^{\rm U}_{p_1,q_1}$ but violates the lower bound $\pi^{\rm L}_{p_1,q_1}$, while π_{s_2,q_1} satisfies $\pi^{\rm L}_{p_1,q_1}$ but violates $\pi^{\rm U}_{p_1,q_1}$, and the same holds for p_2. Thus, both s_1 and s_2 are needed to produce either p_1 or p_2. Moreover, since the blender has no initial inventory, there must be both s_1 and s_2 transferred into the blender before it can deliver any product. This leads to the following tightening constraints:

$$
X_{b_1,p,t} \leq \sum_{t'<t} \tilde{X}_{s,b_1,t'}, \quad \forall s \in \mathbf{S}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{32}
$$

which cut off the optimal solution of ${\rm RM^P}_{(30)}$ as in (31). The strict inequality under the summation sign above is due to the fact that the blender cannot be charged and discharged simultaneously. Solving ${\rm RM^P}_{(30),(32)}$ yields a further tightened upper bound 9455.00.

We can also develop tightening constraints based on bounding the proportion variables $C_{s,b,t}$. For this example, it is easily verified that if the blender is delivering p_1 during time period t, then it must hold that:

$$
0.6 \leq C_{s_1,b_1,t-1} \leq 0.8 \quad \text{and} \quad 0.2 \leq C_{s_2,b_1,t-1} \leq 0.4,
$$

and similarly, if the blender is delivering p_2, we have:

$$
0.2 \leq C_{s_1,b_1,t-1} \leq 0.4 \quad \text{and} \quad 0.6 \leq C_{s_2,b_1,t-1} \leq 0.8.
$$

Utilizing these bounds, we have the following valid constraints:

$$\upsilon^{\mathrm{L}}_{s,p} X_{b_1,p,t} \leq C_{s,b_1,t-1} \leq \upsilon^{\mathrm{U}}_{s,p} X_{b_1,p,t} + (1 - X_{b_1,p,t}), \quad \forall s \in \mathbf{S}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}, \tag{33}$$

where

$$(\upsilon^{\mathrm{L}}_{s_1,p_1}, \upsilon^{\mathrm{L}}_{s_2,p_1}, \upsilon^{\mathrm{U}}_{s_1,p_1}, \upsilon^{\mathrm{U}}_{s_2,p_1}) = (0.6, 0.2, 0.8, 0.4),$$
$$(\upsilon^{\mathrm{L}}_{s_1,p_2}, \upsilon^{\mathrm{L}}_{s_2,p_2}, \upsilon^{\mathrm{U}}_{s_1,p_2}, \upsilon^{\mathrm{U}}_{s_2,p_2}) = (0.2, 0.6, 0.4, 0.8).$$

These constraints state that, during any time period t, if the blender b_1 is delivering a product p (i.e. $X_{b_1,p,t} = 1$), then the proportion $C_{s,b_1,t-1}$ for any stream s must be within the bounds $[\nu^{\mathrm{L}}_{s,p}, \nu^{\mathrm{U}}_{s,p}]$. Otherwise, such bounds are relaxed. Solving $\mathrm{RM^P}_{(30),(32),(33)}$ yields the same optimal objective value and solution as $\mathrm{RM^P}_{(30),(32)}$. Thus, compared to $\mathrm{RM^P}_{(30),(32)}$, Constraints (33) do not directly tighten the upper bound. However, these constraints indeed reduce the feasible region of $\mathrm{RM^P}_{(30),(32)}$, which is also beneficial as discussed in Section 1. To illustrate the reduced feasible region, we fix $C_{s_1,b_1,t_1} = 1$ and $X_{b_1,p_2,t_2} = 0.1$ and resolve $\mathrm{RM}^{\mathrm{P}}_{(30),(32)}$, and then we obtain a solution. However, such setting would be obviously infeasible for $\mathrm{RM^P}_{(30),(32),(33)}$, since (33) is violated.

New tightening constraints can also be developed based on the semicontinuity of the flow variables. In the optimal solution of $\mathrm{RM^P}_{(30),(32),(33)}$, we have:

$$\sum_{t \in \mathbf{T}} F_{b_1,p_1,t} = 3.0000,$$

which agrees with the lower product demand $\delta^{\min}_{p_1,|\mathbf{T}|}$. Intuitively, since the revenue from producing p_2 is significantly greater than the revenue from producing p_1, a profit maximization problem would tend to produce p_2 as much as possible and "push" p_1 to the demand lower bound. However, since $\zeta^{\mathrm{L}}_{b_1,p_1} = 4$, we must have $F_{b_1,p_1,t} \geq 4$ if it is not zero. Thus, we have the following tightening constraint. If $\delta^{\min}_{p_1,|\mathbf{T}|} > 0$, then:

$$\sum_{t \in \mathbf{T}} F_{b_1,p_1,t} \geq \zeta^{\mathrm{L}}_{b_1,p_1}. \tag{34}$$

Solving $\mathrm{RM^P}_{(30),(32),(33),(34)}$ yields a further tightened upper bound 9113.33.

The optimal objective values of all proposed upper bounding models are summarized in Table 2. By employing the new tightening Constraints (30) and (32)–(34), we have reduced the original gap between the globally optimal objective function values of $\mathrm{M^P}$ and $\mathrm{RM^P}$ by 58%.

Table 2: Globally optimal objective values of all models considered in Section 3

Models	Glob. optim. obj.	Notes
$\mathrm{M^P}$	5200.00	
$\mathrm{RM^P}$	14457.30	gap = 9257.30
$\mathrm{RM^P}_{(30)}$	9492.50	
$\mathrm{RM^P}_{(30),(32)}$	9455.00	
$\mathrm{RM^P}_{(30),(32),(33)}$	9455.00	smaller feasible region than $\mathrm{RM^P}_{(30),(32)}$
$\mathrm{RM^P}_{(30),(32),(33),(34)}$	9113.33	smaller gap = 3913.33

The new Constraints (30) and (32)–(34) are formalized and generalized in the next section. Besides these constraints, we also propose other constraints that can be effective in instances that are more complex than Example 1. For instance, we can bound the inventory variables $I^{\mathrm{S}}_{s,b,t}$, similarly to bounding $C_{s,b,t}$ in (33). Based on $[\zeta^{\mathrm{L}}_{b,p}, \zeta^{\mathrm{U}}_{b,p}]$ and $[\delta^{\min}_{p,t}, \delta^{\max}_{p}]$, new constraints are derived to bound the number of material transfers from blenders to product sinks. A similar method can also be applied to bound the number of transfers from stream sources to blenders. The new constraints typically require solving LPs to obtain necessary parameters for tightening, which leads to an LP-based preprocessing algorithm. Numerical tests show that even hundreds of LPs may be solved during preprocessing, but the benefits for having tighter relaxations overweight the computational time for solving these LPs.

4 New tightening constraints

4.1 Group 1

This subsection presents new tightening Constraints (36) based on identifying products that cannot be delivered simultaneously by a blender. We use boldface calligraphic uppercase letters to denote collection of index subsets.

For each $p, p' \in \mathbf{P}$, define a set $\mathbf{Y}_{p,p'}$ such that

$$\mathbf{Y}_{p,p'} := \left\{ \begin{array}{ll} (\hat{I}_{s_1}, \hat{I}_{s_2}, ..., \hat{I}_{s_{|\mathbf{S}|}}) : & \hat{I}_s \leq \min\{\xi_s + \max\{\eta_{s,b} : b \in \mathbf{B}\}, \max\{\omega_b : b \in \mathbf{B}\}\}, \quad \forall s \in \mathbf{S}, \\ & \min\{\zeta^{\mathrm{L}}_{b,p} : b \in \mathbf{B}\} + \min\{\zeta^{\mathrm{L}}_{b,p'} : b \in \mathbf{B}\} \leq \sum_{s \in \mathbf{S}} \hat{I}_s, \\ & \max\{\pi^{\mathrm{L}}_{p,q}, \pi^{\mathrm{L}}_{p',q}\} \sum_{s \in \mathbf{S}} \hat{I}_s \leq \sum_{s \in \mathbf{S}} \pi_{s,q} \hat{I}_s \leq \min\{\pi^{\mathrm{U}}_{p,q}, \pi^{\mathrm{U}}_{p',q}\} \sum_{s \in \mathbf{S}} \hat{I}_s, \quad \forall q \in \mathbf{Q} \end{array} \right\} \tag{35}$$

The constraints for defining $\mathbf{Y}_{p,p'}$ above describe necessary conditions under which a mixture can be delivered to meet demand for both products p and p' simultaneously. The first constraint enforces upper bounds on the amount of each stream in any blender. The second constraint enforces a lower bound on the amount of the mixture that can be delivered as p and p' simultaneously. The third constraint ensures that the mixture satisfies property specifications of both products p and p'. Thus, if $\mathbf{Y}_{p,p'}$ is empty, then no blender can deliver p and p' simultaneously.

For each $p, p' \in \mathbf{P}$, we examine the emptiness of $\mathbf{Y}_{p,p'}$ using the following method. We loop through every property $q \in \mathbf{Q}$. If there exists a property $\hat{q}$ such that $\pi^{\mathrm{U}}_{p,\hat{q}} < \pi^{\mathrm{L}}_{p',\hat{q}}$, then $\mathbf{Y}_{p,p'}$ must be empty. If this does not happen, we solve an LP with a feasible space defined by (35) and with an arbitrary objective function. If this LP is infeasible, then $\mathbf{Y}_{p,p'}$ is empty. Thus, in the worst case, $0.5|\mathbf{P}|(|\mathbf{P}|-1)$ LPs need to be solved for establishing if $\mathbf{Y}_{p,p'}, \forall p, p' \in \mathbf{P}$ is empty.

Now, define $\boldsymbol{\mathcal{P}}$ as a collection of certain subsets of products. Each of these subsets satisfies that, for arbitrary two products p and p' within the subset, the set $\mathbf{Y}_{p,p'}$ is empty. Then, we define another collection $\boldsymbol{\mathcal{P}}^{\max}$, which contains all *maximal subsets* in $\boldsymbol{\mathcal{P}}$. A subset $\mathbf{P}^{\max}$ is maximal if $\mathbf{P}^{\max}$ is not a subset of any other subsets in $\boldsymbol{\mathcal{P}}$. Consider an example of four products (p_1, p_2, p_3, p_4), and suppose that $\mathbf{Y}_{p_1,p_2}$, $\mathbf{Y}_{p_2,p_3}$, $\mathbf{Y}_{p_1,p_3}$, $\mathbf{Y}_{p_1,p_4}$, and $\mathbf{Y}_{p_2,p_4}$ are all empty, and $\mathbf{Y}_{p_3,p_4}$ is nonempty. Then, we have:

$$\begin{aligned}\boldsymbol{\mathcal{P}} &:= \{\{p_1,p_2\},\{p_1,p_3\},\{p_1,p_4\},\{p_2,p_3\},\{p_2,p_4\},\{p_1,p_2,p_3\},\{p_1,p_2,p_4\}\}\,, \\ \boldsymbol{\mathcal{P}}^{\max} &:= \{\{p_1,p_2,p_3\},\{p_1,p_2,p_4\}\}\,.\end{aligned}$$

We construct $\boldsymbol{\mathcal{P}}^{\max}$ using a systematic method. To give an example, consider we have $\{p_1, p_2\}$, $\{p_2, p_3\}$, and $\{p_1, p_3\}$ in $\boldsymbol{\mathcal{P}}$. Then, we can conclude that the union $\{p_1, p_2, p_3\}$ is also in $\boldsymbol{\mathcal{P}}$; that is the subsets $\{p_1, p_2\}$, $\{p_2, p_3\}$, and $\{p_1, p_3\}$ are not maximal.

Given collection $\boldsymbol{\mathcal{P}}^{\max}$, we propose the following new tightening constraints:

$$\sum_{p\in\mathbf{P}^{\max}} X_{b,p,t} \leq 1, \quad \forall b \in \mathbf{B}, \forall \mathbf{P}^{\max} \in \boldsymbol{\mathcal{P}}^{\max}, \forall t \in \mathbf{T}, \tag{36}$$

which states that during one time period, a blender can deliver at most one product in each maximal subset $\mathbf{P}^{\max}$.

4.2 Group 2

This subsection proposes new tightening Constraints (38) and (39) based on bounding the inventory variables $I^{\mathrm{S}}_{s,b,t}$ if a blender is about to deliver products.

Due to the composition consistency between blenders' inventories and outflows, if a blender b is delivering a product p during some time period t, then for each stream $s \in \mathbf{S}$, valid (s,p)-dependent lower bounds $\theta^{\mathrm{L}}_{s,p}$ and upper bounds $\theta^{\mathrm{U}}_{s,p}$ for $I^{\mathrm{S}}_{s,b,t-1}$ can be computed by solving the following LPs. For each $(s,p) \in \mathbf{S} \times \mathbf{P}$:

$$\theta^{\mathrm{L(U)}}_{s,p} := \min(\max) \quad \bar{I}_{s,p}$$

$$\text{s.t.} \quad \bar{I}_{s',p} \le \xi_{s'} + \max\{\eta_{s,b} : b \in \mathbf{B}\}, \quad \forall s' \in \mathbf{S}, \tag{37a}$$

$$\min\{\zeta^{\mathrm{L}}_{b,p} : b \in \mathbf{B}\} \le \sum_{s' \in \mathbf{S}} \bar{I}_{s',p} \le \max\{\omega_b : b \in \mathbf{B}\}, \tag{37b}$$

$$\pi^{\mathrm{L}}_{p,q} \sum_{s' \in \mathbf{S}} \bar{I}_{s',p} \le \sum_{s' \in \mathbf{S}} \pi_{s',q} \bar{I}_{s',p} \le \pi^{\mathrm{U}}_{p,q} \sum_{s' \in \mathbf{S}} \bar{I}_{s',p}, \quad \forall q \in \mathbf{Q}, \tag{37c}$$

where the auxiliary variable $\bar{I}_{s',p}$ represents the possible amount of stream s' used for producing product p within the range $[\min\{\zeta^{\mathrm{L}}_{b,p} : b \in \mathbf{B}\}, \max\{\omega_b : b \in \mathbf{B}\}]$. Constraints (37a) include an upper bound of the amount of each stream in any blender. Constraints (37b) are based on lower and upper bounds of the mixture's amount in any blender if the mixture can be delivered as product p. Constraints (37c) ensure that the mixture satisfies the property specification of product p. Once $\theta^{\mathrm{L}}_{s,p}$ and $\theta^{\mathrm{U}}_{s,p}$ have been calculated, we use them in the following new tightening constraints:

$$X_{b,p,t}\theta^{\mathrm{L}}_{s,p} \le I^{\mathrm{S}}_{s,b,t-1} \le X_{b,p,t}\theta^{\mathrm{U}}_{s,p} + \min\{\xi_s + \eta_{s,b}, \omega_b\}(1 - X_{b,p,t}), \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{38}$$

If $X_{b,p,t} = 1$, then Constraints (38) above bound $I^{\mathrm{S}}_{s,b,t-1}$ by $\theta^{\mathrm{L}}_{s,p}$ and $\theta^{\mathrm{U}}_{s,p}$; otherwise, such bounds are relaxed. A total of $2|\mathbf{S}||\mathbf{P}|$ LPs (37) need to be solved to construct the new constraints.

Now, consider the lower bounds $\theta^{\mathrm{L}}_{s,p}$ obtained by solving the LPs (37). If $X_{b,p,t} = 1$ and $\theta^{\mathrm{L}}_{s,p} - \eta_{s,b} > 0$, then there must be material transfer from stream source s to blender b prior to time period t. This leads to the following new tightening constraints:

$$X_{b,p,t} \le \sum_{t' < t} \tilde{X}_{s,b,t'}, \quad \forall t \in \mathbf{T}, \forall (s,b,p) \in \mathbf{S} \times \mathbf{B} \times \mathbf{P} : \ \theta^{\mathrm{L}}_{s,p} - \eta_{s,b} > 0. \tag{39}$$

Similar tightening constraints were proposed by Chen and Maravelias.[1] For each $(p,q) \in$

$\mathbf{P} \times \mathbf{Q}$, they define sets of *good* streams in terms of $\pi^{\mathrm{L}}_{p,q}$ and $\pi^{\mathrm{U}}_{p,q}$ as follows:

$$\mathbf{S}^{\mathrm{L,g}}_{p,q} := \{s \in \mathbf{S} : \pi^{\mathrm{L}}_{p,q} \leq \pi_{s,q}\},$$
$$\mathbf{S}^{\mathrm{U,g}}_{p,q} := \{s \in \mathbf{S} : \pi_{s,q} \leq \pi^{\mathrm{U}}_{p,q}\},$$

and similarly, sets of *bad* streams are defined as:

$$\mathbf{S}^{\mathrm{L,b}}_{p,q} := \{s \in \mathbf{S} : \pi^{\mathrm{L}}_{p,q} > \pi_{s,q}\},$$
$$\mathbf{S}^{\mathrm{U,b}}_{p,q} := \{s \in \mathbf{S} : \pi_{s,q} > \pi^{\mathrm{U}}_{p,q}\}.$$

Based on the fact that good streams are required for each product and assuming no initial inventory in blenders, they proposed the following valid constraints:

$$\begin{aligned} X_{b,p,t} &\leq \sum_{s \in \mathbf{S}^{\mathrm{U,g}}_{p,q}} \sum_{t' \leq t} \tilde{X}_{s,b,t'}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall q \in \mathbf{Q}, \forall t \in \mathbf{T}, \\ X_{b,p,t} &\leq \sum_{s \in \mathbf{S}^{\mathrm{L,g}}_{p,q}} \sum_{t' \leq t} \tilde{X}_{s,b,t'}, \quad \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall q \in \mathbf{Q}, \forall t \in \mathbf{T}. \end{aligned} \tag{40}$$

Note that constructing (40) is more efficient since it does not require solving any LPs. However, our new Constraints (39) offer the following benefits. Firstly, (39) applies to nonzero initial inventory of blenders while (40) does not. Secondly, in the case of no initial inventory of blenders, the new constraints are guaranteed to be at least as tight as (40). For each $(p, q) \in \mathbf{P} \times \mathbf{Q}$, there must be at least one $s \in \mathbf{S}^{\mathrm{U,g}}_{p,q}$ such that $\theta^{\mathrm{L}}_{s,p} > 0$ and at least one $s \in \mathbf{S}^{\mathrm{L,g}}_{p,q}$ such that $\theta^{\mathrm{L}}_{s,p} > 0$. This makes (39) no less tight than (40). Moreover, (39) is tighter than (40) in certain situations. For example, even if a stream s is a bad stream for all $q \in \mathbf{Q}$ for a product p, $\theta^{\mathrm{L}}_{s,p}$ may still be nonzero due to the demand requirements of p. In this situation, (39) is tighter than (40) because it does not consider bad streams. Besides, for a product p, if more than one good streams s have $\theta^{\mathrm{L}}_{s,p} > 0$, then (39) is also tighter than (40). Furthermore, note that (39) employs strict inequality $t < t'$ in the summation sign, in contrast to $t \leq t'$ in (40). The addend for the summation in (39) is due to the fact that a blender cannot be charged and discharged simultaneously, which makes our new constraints potentially tighter as well.

4.3 Group 3

The tightening constraints proposed in the previous two subsections are applicable to both the P- and S-formulations. In this subsection, we propose new tightening constraints (42) involving the proportion variables $C_{s,b,t}$ that are present in the P-formulation. Analogously to the bounds $\theta^{\mathrm{L}}_{s,p}$ and $\theta^{\mathrm{U}}_{s,p}$, we define $\upsilon^{\mathrm{L}}_{s,p}$ and $\upsilon^{\mathrm{U}}_{s,p}$ which are (s,p)-dependent bounds for $C_{s,b,t-1}$ if $X_{b,p,t} = 1$. These bounds are computed by solving the following LPs. For each $(s,p) \in \mathbf{S} \times \mathbf{P}$:

$$
\begin{aligned}
\upsilon^{\mathrm{L(U)}}_{s,p} := \min(\max) \quad & \bar{C}_{s,p} \\
\text{s.t.} \quad & \sum_{s' \in \mathbf{S}} \bar{C}_{s',p} = 1, && \text{(41a)} \\
& \pi^{\mathrm{L}}_{p,q} \leq \sum_{s' \in \mathbf{S}} \pi_{s',q} \bar{C}_{s',p} \leq \pi^{\mathrm{U}}_{p,q}, \quad \forall q \in \mathbf{Q}. && \text{(41b)}
\end{aligned}
$$

where the auxiliary variable $\bar{C}_{s',p}$ represents the possible proportion of stream s' in product p. Constraint (41a) ensures that the proportions of all streams summing up to one, and Constraints (41b) ensure that the product p's property specification is satisfied. Then, we have the following new tightening constraints:

$$
X_{b,p,t}\upsilon^{\mathrm{L}}_{s,p} \leq C_{s,b,t-1} \leq X_{b,p,t}\upsilon^{\mathrm{U}}_{s,p} + (1 - X_{b,p,t}), \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \qquad (42)
$$

A total of $2|\mathbf{S}||\mathbf{P}|$ LPs need to be solved to construct (42).

4.4 Group 4

This subsection proposes new tightening Constraints (44), (47)–(49), and (52)–(54) derived from the logistic constraints between blenders and product sinks. The new constraints do not require solving any LP and are inspired from Velez and Maravelias,[50] who proposed similar tightening constraints for general chemical scheduling problems.

As discussed in Section 3, for each $(p,t) \in \mathbf{P} \times \mathbf{T}$, the range of $\sum_{b \in \mathbf{B}} \sum_{t' \leq t} F_{b,p,t'}$ can be disjoint. Based on this fact, we can tighten the right-hand side of Constraints (5) and (6) as follows. Consider arbitrary $(p,t) \in \mathbf{P} \times \mathbf{T}$, and for each $b \in \mathbf{B}$, let $\epsilon_{b,p,t}$ be a non-negative integer, which denotes the number of material transfers from blender b to product sink p before time point t. Define a set $\mathbf{K}_{p,t}$ (see the Supporting Information for an illustration) as:

$$\mathbf{K}_{p,t} := \left\{ \begin{array}{c} (\epsilon_{b_1,p,t}, \epsilon_{b_2,p,t}, ..., \epsilon_{b_{|\mathbf{B}|},p,t}) : \delta_{p,t}^{\min} \leq \sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{U}} \epsilon_{b,p,t}, \delta_p^{\max} \geq \sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{L}} \epsilon_{b,p,t}, \\ \text{and } \epsilon_{b,p,t} \leq \mathrm{ord}(t), \forall b \in \mathbf{B} \end{array} \right\}, \tag{43}$$

where $\mathrm{ord}(t)$ denotes the order of time point t. Observe that for each $(\epsilon_{b_1,p,t}, \epsilon_{b_2,p,t}, ..., \epsilon_{b_{|\mathbf{B}|},p,t}) \in \mathbf{K}_{p,t}$, a continuous subrange of $\sum_{b \in \mathbf{B}} \sum_{t' \leq t} F_{b,p,t'}$ is the interval from $\sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{L}} \epsilon_{b,p,t}$ to $\sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{U}} \epsilon_{b,p,t}$, denoted as $[\sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{L}} \epsilon_{b,p,t}, \sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{U}} \epsilon_{b,p,t}]$. Then, the potentially disjoint range of $\sum_{b \in \mathbf{B}} \sum_{t' \leq t} F_{b,p,t'}$ is given by:

$$\bigcup_{(\epsilon_{b_1,p,t}, \epsilon_{b_2,p,t}, ..., \epsilon_{b_{|\mathbf{B}|},p,t}) \in \mathbf{K}_{p,t}} \left[\sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{L}} \epsilon_{b,p,t}, \sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{U}} \epsilon_{b,p,t} \right]$$

Thus, a lower bound of the range above can be described as:

$$\hat{\delta}_{p,t}^{\min} := \min \left\{ \sum_{b \in \mathbf{B}} \zeta_{b,p}^{\mathrm{L}} \epsilon_{b,p,t} : (\epsilon_{b_1,p,t}, \epsilon_{b_2,p,t}, ..., \epsilon_{b_{|\mathbf{B}|},p,t}) \in \mathbf{K}_{p,t} \right\}.$$

If we have $\hat{\delta}_{p,t}^{\min} > \delta_{p,t}^{\min}$, then the following new constraints are tighter than (6):

$$\sum_{b \in \mathbf{B}} \sum_{t' \leq t} F_{b,p,t'} \geq \hat{\delta}_{p,t}^{\min}, \quad \forall p \in \mathbf{P}, \forall t \in \mathbf{T}. \tag{44}$$

Note that to compute $\hat{\delta}^{\min}_{p,t}$, we may not need to examine every element of $\mathbf{K}_{p,t}$. For each $b \in \mathbf{B}$, let $\hat{\epsilon}_{b,p,t} := \lceil \frac{\delta^{\min}_{p,t}}{\zeta^{\mathrm{U}}_{b,p}} \rceil$. It is easily verified that:

$$\hat{\delta}^{\min}_{p,t} \equiv \min\left\{\sum_{b\in\mathbf{B}} \zeta^{\mathrm{L}}_{b,p}\epsilon_{b,p,t} : (\epsilon_{b_1,p,t}, \epsilon_{b_2,p,t}, ..., \epsilon_{b_{|\mathbf{B}|},p,t}) \in \mathbf{K}_{p,t} \text{ and } \epsilon_{b,p,t} \leq \hat{\epsilon}_{b,p,t}, \forall b \in \mathbf{B}\right\}. \quad (45)$$

Similarly to Constraints (44), we can also tighten the right-hand side of Constraints (5). For each $p \in \mathbf{P}$, let

$$\check{\delta}^{\max}_{p} := \max\left\{\sum_{b} \zeta^{\mathrm{U}}_{b,p}\epsilon_{b,p,|\mathbf{T}|} : (\epsilon_{b_1,p,|\mathbf{T}|}, \epsilon_{b_2,p,|\mathbf{T}|}, ..., \epsilon_{b_{|\mathbf{B}|},p,|\mathbf{T}|}) \in \mathbf{K}_{p,|\mathbf{T}|}\right\}, \quad (46)$$

which is the upper bound of the disjoint range of $\sum_{b\in\mathbf{B}}\sum_{t\in\mathbf{T}} F_{b,p,t}$. If $\check{\delta}^{\max}_{p} < \delta^{\max}_{p}$, then the following new constraints are tighter than (5):

$$\sum_{b\in\mathbf{B}}\sum_{t\in\mathbf{T}} F_{b,p,t} \leq \check{\delta}^{\max}_{p}, \quad \forall p \in \mathbf{P}. \quad (47)$$

The following new constraints bound the total number of material transfer decisions from all blenders to each product sink based on the tightened products' demand bounds $(\hat{\delta}^{\min}_{p,t}, \check{\delta}^{\max}_{p})$ and the extremal flowrates $(\zeta^{\mathrm{L}}_{b,p}, \zeta^{\mathrm{U}}_{b,p})$:

$$\sum_{b\in\mathbf{B}}\sum_{t'\leq t} X_{b,p,t} \geq \left\lceil \frac{\hat{\delta}^{\min}_{p,t}}{\max\{\zeta^{\mathrm{U}}_{b,p} : b \in \mathbf{B}\}} \right\rceil, \quad \forall p \in \mathbf{P}, \forall t \in \mathbf{T}, \quad (48)$$

$$\sum_{b\in\mathbf{B}}\sum_{t\in\mathbf{T}} X_{b,p,t} \leq \left\lfloor \frac{\check{\delta}^{\max}_{p}}{\min\{\zeta^{\mathrm{L}}_{b,p} : b \in \mathbf{B}\}} \right\rfloor, \quad \forall p \in \mathbf{P}. \quad (49)$$

Moreover, new Constraints (52) and (53) below also bound the number of material transfers from all blenders to each product sink, but in a different way. For each $(p,t) \in \mathbf{P} \times \mathbf{T}$, we define the minimum upper bound of all continuous subranges of $\sum_{b\in\mathbf{B}}\sum_{t'\leq t} F_{b,p,t}$ as:

$$\phi_{p,t} := \min\left\{\sum_{b} \zeta^{\mathrm{U}}_{b,p}\epsilon_{b,p,t} : (\epsilon_{b_1,p,t}, \epsilon_{b_2,p,t}, ..., \epsilon_{b_{|\mathbf{B}|},p,t}) \in \mathbf{K}_{p,t}\right\}, \quad (50)$$

and the maximum lower bound of all continuous subranges of $\sum_{b\in\mathbf{B}}\sum_{t\in\mathbf{T}} F_{b,p,t}$ as:

$$\lambda_p := \max\left\{\sum_b \zeta^{\mathrm{L}}_{b,p}\epsilon_{b,p,|\mathbf{T}|} : (\epsilon_{b_1,p,|\mathbf{T}|}, \epsilon_{b_2,p,|\mathbf{T}|}, ..., \epsilon_{b_{|\mathbf{B}|},p,|\mathbf{T}|}) \in \mathbf{K}_{p,|\mathbf{T}|}\right\}. \tag{51}$$

Then, the following new tightening constraints are valid:

$$\sum_{b\in\mathbf{B}}\sum_{t'\leq t} X_{b,p,t}\zeta^{\mathrm{U}}_{b,p} \geq \phi_{p,t}, \quad \forall p \in \mathbf{P}, \forall t \in \mathbf{T}, \tag{52}$$

$$\sum_{b\in\mathbf{B}}\sum_{t\in\mathbf{T}} X_{b,p,t}\zeta^{\mathrm{L}}_{b,p} \leq \lambda_p, \quad \forall p \in \mathbf{P}. \tag{53}$$

Finally, the following new constraints fix $X_{b,p,1}$ to be zero, if blender b cannot deliver product p during the first time period:

$$\begin{aligned} X_{b,p,1} = 0, \quad \forall (b,p) \in \mathbf{B}\times\mathbf{P} : \sum_{s\in\mathbf{S}} \eta_{s,b} < \zeta^{\mathrm{L}}_{b,p}, \\ \text{or } \sum_{s\in\mathbf{S}} \pi_{s,q}\eta_{s,b} > \pi^{\mathrm{U}}_{p,q}\sum_{s\in\mathbf{S}} \eta_{s,b}, \quad \text{for some } q \in \mathbf{Q}, \\ \text{or } \sum_{s\in\mathbf{S}} \pi_{s,q}\eta_{s,b} < \pi^{\mathrm{L}}_{p,q}\sum_{s\in\mathbf{S}} \eta_{s,b}, \quad \text{for some } q \in \mathbf{Q}. \end{aligned} \tag{54}$$

All parameters for bound tightening, e.g. $\hat{\delta}^{\min}_{p,t}$ defined in (45), $\check{\delta}^{\max}_{p}$ defined in (46), $\phi_{p,t}$ defined in (50), and λ_p defined in (51) are computed using an iterative routine adapted from Velez et al. [50].

4.5 Group 5

Similarly to the new constraints in the previous subsection, this subsection presents new Constraints (56)–(58) derived from the logistic constraints between stream sources and blenders.

For each $t \in \mathbf{T}$ and $s \in \mathbf{S}$, we first compute $\chi_{s,t}$ by solving LPs, where conceptually, $\chi_{s,t}$ is the least amount of stream s that needs to be used prior to time point t. For each $t \in \mathbf{T}$

and each $s \in \mathbf{S}$:

$$\chi_{s,t} := \min \quad \sum_{p \in \mathbf{P}} \bar{F}_{s,p,t} \tag{55a}$$

$$\text{s.t.} \quad \sum_{p \in \mathbf{P}} \bar{F}_{s',p,t} \leq \min\{\xi_{s'}, (\text{ord}(t)-1)\sum_{b \in \mathbf{B}} \tilde{\xi}^{\text{U}}_{s',b}\} + \sum_{b \in \mathbf{B}} \eta_{s',b}, \quad \forall s' \in \mathbf{S}, \tag{55b}$$

$$\delta^{\min}_{p,t} \leq \sum_{s' \in \mathbf{S}} \bar{F}_{s',p,t} \leq \delta^{\max}_{p}, \quad \forall p \in \mathbf{P}, \tag{55c}$$

$$\pi^{\text{L}}_{p,q} \sum_{s' \in \mathbf{S}} \bar{F}_{s',p,t} \leq \sum_{s' \in \mathbf{S}} \pi_{s',q} \bar{F}_{s',p,t} \leq \pi^{\text{U}}_{p,q} \sum_{s' \in \mathbf{S}} \bar{F}_{s',p,t}, \quad \forall q \in \mathbf{Q}, \forall p \in \mathbf{P}, \tag{55d}$$

where the auxiliary variable $\bar{F}_{s',p,t}$ represents the amount of stream s' used for producing product p by time point t. Constraints (55b) describe an upper bound of each stream's amount that is available for producing products by time point t. Since all blenders cannot be charged and discharged simultaneously, streams entering the blenders during time period t cannot be used for delivering products by time point t. Thus, the term $(\text{ord}(t)-1)\sum_{b \in \mathbf{B}} \tilde{\xi}^{\text{U}}_{s',b}$ in (55b) is a valid upper bound for stream s' from outside of blenders that can be used for delivering products by time point t. Constraints (55c) and (55d) ensure that the products' property requirement and demand requirement by time point t are satisfied. Then, a lower bound $\tau_{s,t}$ of stream s entering all blenders prior to time point t is given by:

$$\tau_{s,t} := \max\{0, \chi_{s,t} - \sum_{b \in \mathbf{B}} \eta_{s,b}\}, \quad \forall s \in \mathbf{S}, \forall t \in \mathbf{T}.$$

Next, our new tightening constraints for bounding the number of material transfer operations from each stream source to all blenders are similar to Constraints (48) and (52) as follows:

$$\sum_{b \in \mathbf{B}} \sum_{t' < t} \tilde{X}_{s,b,t} \geq \left\lceil \frac{\tau_{s,t}}{\max\{\tilde{\xi}^{\text{U}}_{s,b} : b \in \mathbf{B}\}} \right\rceil, \quad \forall s \in \mathbf{S}, \forall t \in \mathbf{T}, \tag{56}$$

$$\sum_{b \in \mathbf{B}} \sum_{t' < t} \tilde{X}_{s,b,t} \tilde{\xi}^{\text{U}}_{s,b} \geq \psi_{s,t}, \quad \forall s \in \mathbf{S}, \forall t \in \mathbf{T}, \tag{57}$$

where $\psi_{s,t}$ is the minimum upper bound of all continuous subranges of $\sum_{b\in\mathbf{B}}\sum_{t'<t}\tilde{F}_{s,b,t}$. The method for calculating $\psi_{s,t}$ is similar to the method for calculating $\phi_{p,t}$ in the previous subsection, and is adapted from Velez et al.[50]. Note that (56) and (57) employ a strict inequality $t' < t$ instead of a weak inequality $t' \leq t$. This is for the same reason as we use $(\mathrm{ord}(t)-1)\sum_{b\in\mathbf{B}}\tilde{\xi}^{\mathrm{U}}_{s',b}$ instead of $\mathrm{ord}(t)\sum_{b\in\mathbf{B}}\tilde{\xi}^{\mathrm{U}}_{s',b}$ in (55b) as explained above. In the worst case, constructing Constraints (56) and (57) requires solving $|\mathbf{T}||\mathbf{S}|$ LPs (55a)–(55d) to compute $\chi_{s,t}$; with the exception that arises when the LPs at some t and $t-1$ are identical.

Finally, in a globally optimal solution of the original MINLP, streams will not enter blenders during the final time period, since such amounts cannot be delivered as products. This leads to the following new constraints:

$$\tilde{X}_{s,b,|\mathbf{T}|} = 0, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}. \tag{58}$$

5 Numerical testing

We test our new preprocessing algorithm and tightening constraints using 86 numerical instances of the MBSP as defined in Section 2.1. For reference, Table 3 summarizes the new tightening constraints proposed in Section 4 and the number of LPs that need to be solved during preprocessing. The numerical instances have roughly 4–8 time periods, 4–10 streams, 2–6 blenders, 4–8 products, and 6–10 properties. These instances and the generation method are available in the Supporting Information. All models are implemented in GAMS v42.1.0.[51] We use CPLEX v22.1.1.0[52] to solve LPs during preprocessing and use BARON v23.1.5[45] and GUROBI v10.0.0[46] to solve the MINLP models. For both BARON and GUROBI, the CPU time limit `reslim` is set to 7,200 seconds, the relative optimality gap `optcr` is set to 1%, and all other default settings are used. All numerical tests are conducted on Princeton University's Della cluster (https://researchcomputing.princeton.edu/systems/ della). Each job is submitted to the processor Intel Cascade Lake with 2.8 GHz with a request of 4GB of memory. Such amount of memory is verified to be sufficient for all jobs. Our numerical tests have three goals, for both BARON and GUROBI:

- compare computational performance of the P- and S-formulations,
- examine the effectiveness of each group of new constraints for enhancing the P- and S-formulations, and
- find further improved formulations employing combinations of the new constraints.

Table 3: New constraint groups and the number of LPs need to be solved during preprocessing

Group No.	New constraints	# LPs need to be solved
1	(36)	$0.5\lvert\mathbf{P}\rvert(\lvert\mathbf{P}\rvert - 1)$ at most
2	(38),(39)	$2\lvert\mathbf{S}\rvert\lvert\mathbf{P}\rvert$
3	(42)	$2\lvert\mathbf{S}\rvert\lvert\mathbf{P}\rvert$
4	(44), (47)–(49), (52)–(54)	None
5	(56)–(58)	$\lvert\mathbf{T}\rvert\lvert\mathbf{S}\rvert$ at most

We employ the following notation to represent different MINLP models solved by different solvers. We use “b” and “g” to respectively indicate the solvers BARON and GUROBI, “P” and “S” to respectively indicate the P- and S-formulations, and digits consistent with Table 3 to indicate the employed groups of new constraints. For example, $\mathrm{bM}^{\mathrm{P}}_{1,2}$ represents a run, using BARON, to solve an MINLP model of the P-formulation with the new constraint Groups 1 and 2.

We use performance graphs[53] to perform the comparisons, where the CPU times include both the preprocessing time and MINLP solver solution time. Since the goal of this study is to explore the effectiveness of the proposed constraints, we make comparisons across variants of the same formulation (P- or S-) using the same solver (GUROBI or BARON). For a set of runs that correspond to the same combination of formulation and solver, we generate a performance graph using the data of instances that are not trivial (i.e., solved by all models in less than 15 seconds) nor intractable (i.e., no model solved the instance to optimality within 7,200 seconds). Please note that, for example, an instance that is intractable for models corresponding to one formulation-solver combination can be solvable for another combination. Thus, each performance graph is generated using a different set of instances. The numbers of trivial, intractable, and instances used to generate each performance graph are summarized in Table 4.

Table 4: Number of instances used to generate each performance graph

	Instance types		
	Trivial	Intractable	Used in the figure
Figure 2	0	24	62
Figure 3A	0	34	52
Figure 3B	1	34	51
Figure 3C	3	21	62
Figure 3D	4	22	60
Figure 4A	0	35	51
Figure 4B	3	33	50
Figure 4C	2	18	66
Figure 4D	5	20	61

Note that the longest preprocessing time in our numerical tests occurs when applying all new tightening constraints on the largest numerical instance. In this situation, hundreds of LPs are solved during preprocessing. However, these LPs, as described in Section 4, are easy to solve; the longest preprocessing time in our numerical tests is less than ten seconds. The data used for generating all performance graphs is given in the Supporting Information.

Firstly, we compare the performance of the base P- and S-formulations. As discussed earlier, it is not currently clear which formulation is in general faster. Figure 2 depicts the performance graph comparing gM^P, gM^S, bM^P, and bM^S. It is shown that when using BARON, the S-formulation is significantly more efficient than the P-formulation, and whereas when using GUROBI, the S-formulation is still better but the performance gap between the two formulations becomes closer. However, we do not draw a general conclusion here that the S-formulation is superior to the P-formulation for the MBSP, since a more rigorous comparison between these two formulations is needed. Moreover, Figure 2 shows that

GUROBI is more efficient than BARON for solving the MBSP. This observation coincides with the results in Cheng and Li[28] which also studies MBSP, and in Karia et al.[54] which studies general MIQCP. This may because GUROBI is tailored for MIQCP, while BARON is a general global optimization solver.

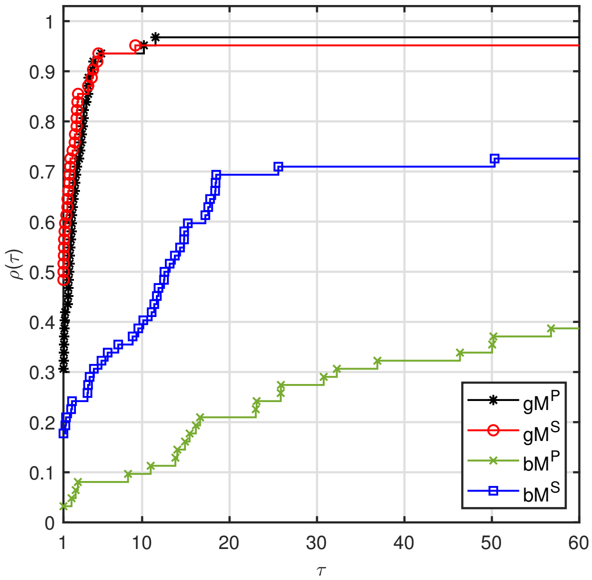


Figure 2: Performance graph comparing the base formulations solved using BARON or GUROBI

Next, we examine the effectiveness of each individual group of new constraints. All new constraints are applicable to the P-formulation, and all except Group 3 are applicable to the S-formulation. As shown in Figure 3, each of the new constraint Groups 1, 2, and 3 can lead to improvement for solving both formulations using both MINLP solvers. Compared to Groups 1, 2, and 3, Groups 4 and 5 tend to have smaller improvement. Notably, Group 4 has a negative impact when applying it to solve the S-formulation using BARON. We note that adding new constraints can tighten the relaxation of a model, but it simultaneously increases the relaxation's size.

Next, we are interested in whether using multiple groups of new constraints can further improve computational performance. For example, Groups 1, 2, 4, and 5 can all be applied on the S-formulation, which leads to a total of ten possible combinations. However, using

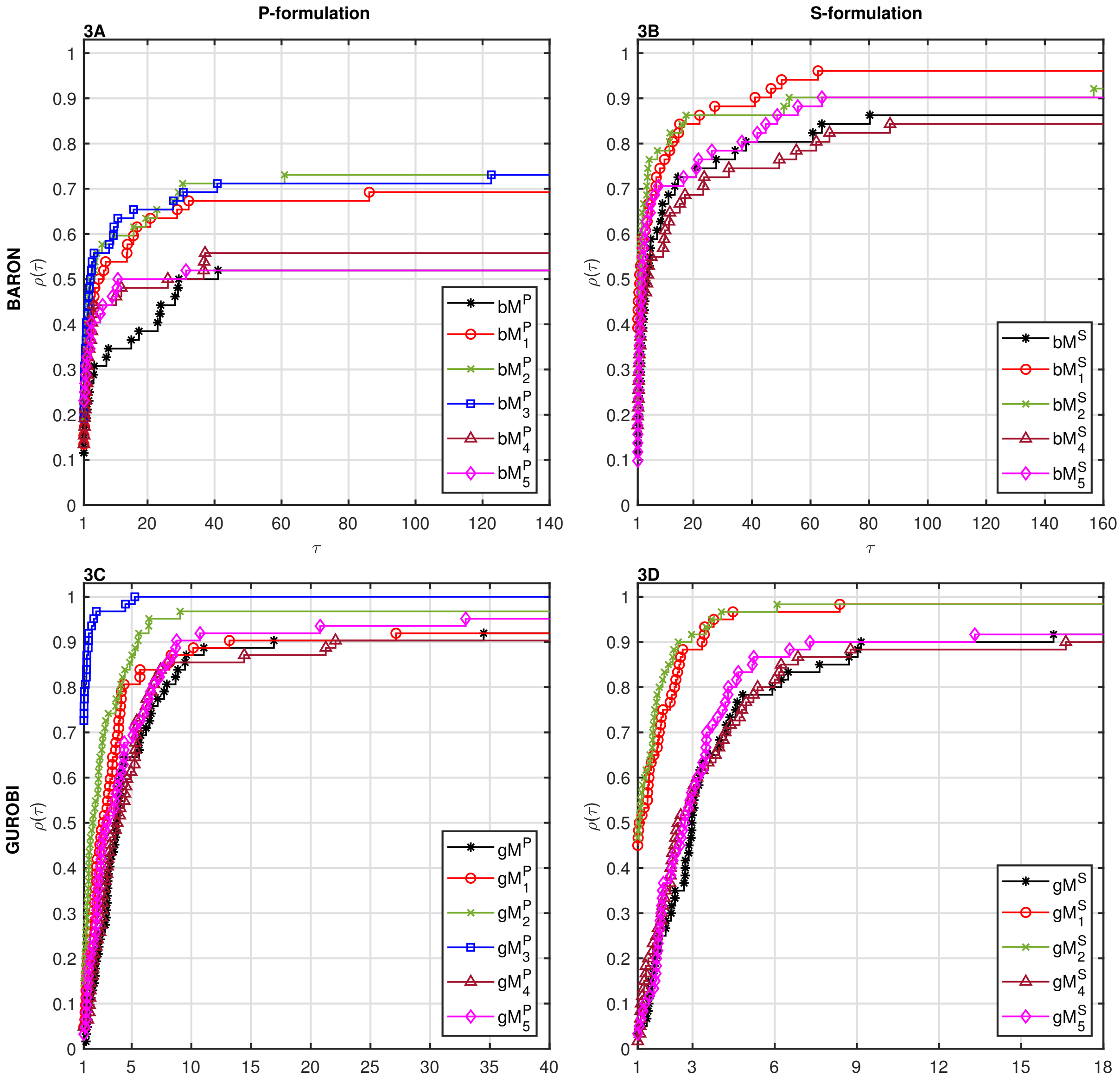


Figure 3: Performance graphs examining the effectiveness of each group of new constraints, using BARON or GUROBI

more tightening constraints does no necessarily lead to better performance, so we test all combinations of new constraints that are applicable to the two formulations. For brevity, in Figure 4, we compare the base formulation, the best formulation when using only one group of new constraints, and the best formulation when using multiple groups of new constraints. It is shown that for all formulations and solvers, using multiple groups of new constraints can improve the performance significantly. When using GUROBI, all new constraints (if applicable) are recommended to be added simultaneously. When using BARON, the best performance is obtained when using all new constraints except Group 4.

6 Extension to other types of MBSP

This section discusses how our new tightening constraints can be applied to other variants of the MBSP studied in literature.

First, in MBSPs with a cost minimization objective, there are typically no product maximum demand requirements $\delta_p^{\max}$. Instead, the goal is to find the least cost production schedule to fulfill minimum product demand. Since there are no $\delta_p^{\max}$, our new Constraints (47), (49), and (53) are not applicable while all other tightening constraints remain valid. Second, if flowrate lower bounds from stream sources to blenders are available, then with $\tau_{s,t}$ in the role of $\delta_{p,t}^{\min}$, Constraints (44) can be adapted to bound material transfer decisions between streams and blenders.

A third variant arises when instead of having $\delta_{p,t}^{\min}$ and $\delta_p^{\max}$, product demand bounds are imposed for each time period. Moreover, for each product, a tank is employed to temporarily store the product received from the blenders, and to deliver the product when necessary. Let $\sigma_{p,t}^{\min}$ and $\sigma_{p,t}^{\max}$ denote the minimum and maximum demand for product p during time period t, respectively. For this type of problem, even though $\delta_{p,t}^{\min}$ and $\delta_p^{\max}$ defined in this article

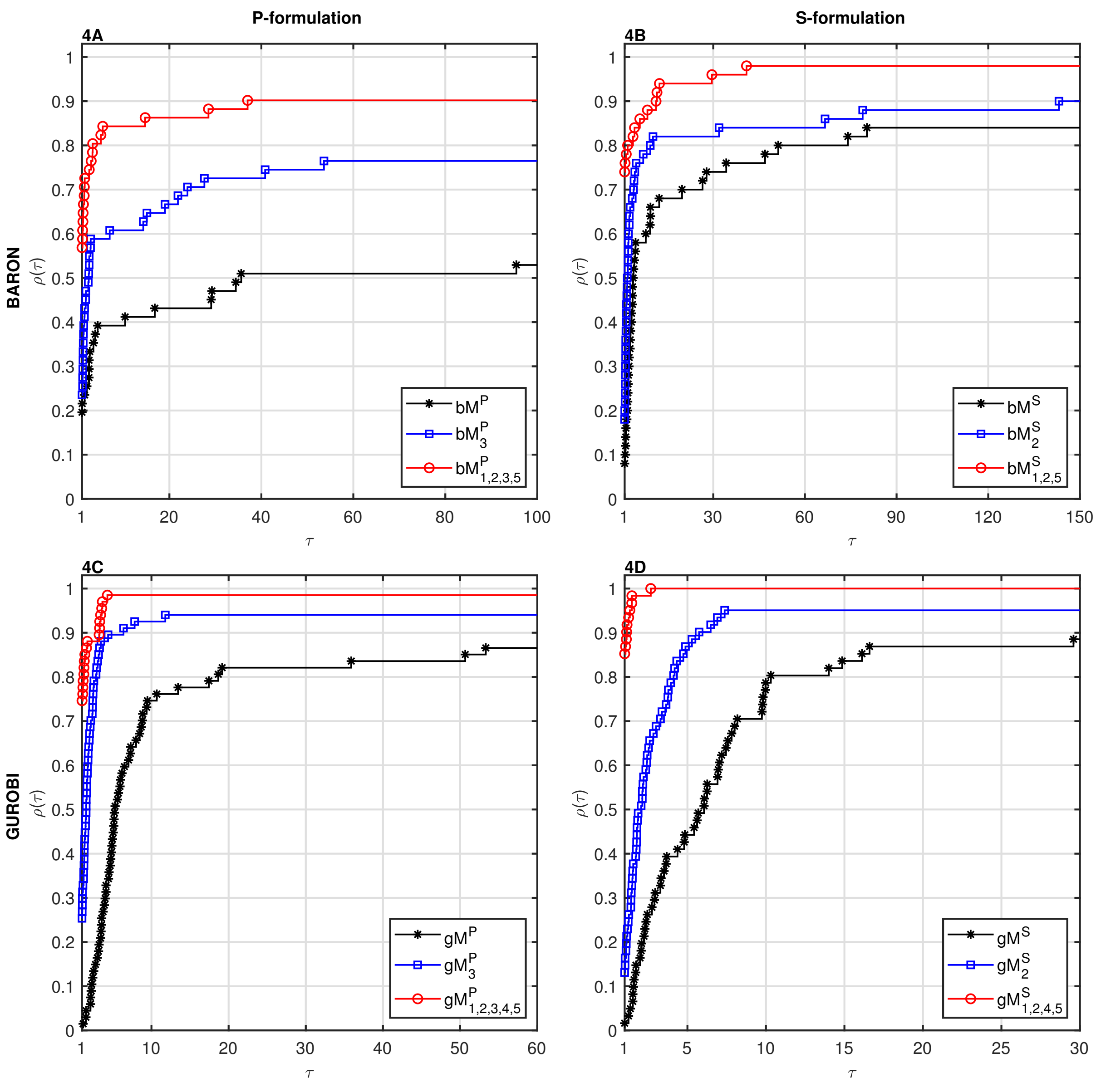


Figure 4: Performance graphs including formulations with multiple groups of new constraints

are not directly given, they can be easily derived from $\sigma_{p,t}^{\min}$ and $\sigma_{p,t}^{\max}$ as follows:

$$\delta_{p,t}^{\min} := \sum_{t' \leq t} \sigma_{p,t}^{\min}, \quad \forall p \in \mathbf{P}, \forall t \in \mathbf{T},$$
$$\delta_{p}^{\max} := \sum_{t \in \mathbf{T}} \sigma_{p,t}^{\max}, \quad \forall p \in \mathbf{P}.$$

Fourth, we may also have streams deliveries. In this case, the overall availability ξ_s used in the new constraints can be set to be the summation of all arrivals of streams throughout the scheduling horizon. Moreover, the authors expect that such time-dependent availability can be utilized to derive further tightened bounds or constraints, which constitutes a future work.

Fifth, another variant arises when an operational rule for blenders that is more restrictive than (7) is applicable, namely, that a blender cannot be charged and discharged simultaneously and can feed at most one product tank at any time, which can be modeled as:

$$\tilde{X}_{s,b,t} \leq 1 - \sum_{p \in \mathbf{P}} X_{b,p,t}, \quad \forall s \in \mathbf{S}, \forall b \in \mathbf{B}, \forall t \in \mathbf{T}.$$

In this case, new constraint Group 1 cannot tighten relaxations of MINLP models, while all other new constraints remain applicable.

Finally, the proposed constraints can be modified to account for special network connectivity (e.g. some (s, b) and (b, p) arcs may be forbidden, arcs between blenders may be allowed, and certain blenders may only charge other blenders but cannot deliver products). In such general case, all new tightening constraints are applicable to existing (s, b) and (b, p) arcs.

7 Conclusions

This article has proposed new constraints for tightening mixed-integer nonlinear programming (MINLP) models used to solve the multiperiod blend scheduling problem (MBSP). These new constraints are derived from the physical information lost due to relaxation, and typically require solving auxiliary linear programs (LPs) during preprocessing. We have tested the effectiveness of the new tightening constraints for enhancing the widely-used proportion-based (P-)formulation and split-fraction-based (S-)formulation of the MBSP. Even though hundreds of LPs may be solved during preprocessing, the preprocessing takes no more than ten seconds for all instances. It is shown that for both P- and S-formulations, our new tightening constraints significantly reduce the overall computational time, which includes both preprocessing and MINLP solver time. Finally, we discuss how to apply the new tightening constraints to various other settings of the MBSP.

Acknowledgement

The authors acknowledge financial support from the National Science Foundation under grant CBET-2026980.

Data availability statement

The data underlying this study are available in the published article and its Supporting Information.

Supporting Information Available

The Supporting Information includes the following files:

- `supporting_information.pdf`: a graphic illustration of the set $\mathbf{K}_{p,t}$ defined in Section 4.4 and a brief description of how the numerical instances were generated,
- `results.xlsx`: data for generating all performance graphs in this article,
- a folder `instances`: all numerical instances as `.gdx` files.

This information is available free of charge via the Internet at `https://nam12.safelinks.protection.outlook.com/?url=http%3A%2F%2Fpubs.acs.org%2F&data=05%7C01%7Cmaravelias%40princeton.edu%7C3f906d4610824d139d1b08dbce28aca8%7C2ff601167431425db5af077d7791bda4%7C0%7C0%7C638330446386189367%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzI3D%7C3000%7C%7C%7C&sdata=NlPAA05IfpAmh%2Fw09iRz9HvtIYbq3DEDAfGANnjTZM8%3D&reserved=0`.

Nomenclature

We use uppercase letters for nonnegative variables, lowercase letters for indices, boldface uppercase letters for sets, and lowercase Greek letters for parameters. We use calligraphic uppercase letters to denote collection of index subsets.

Sets

$s \in \mathbf{S}$	Streams or stream sources
$q \in \mathbf{Q}$	Properties
$b \in \mathbf{B}$	Blenders
$p \in \mathbf{P}$	Products or product sinks
$t \in \mathbf{T}$	Time periods (ordered set): $\{1, 2, ..., \lvert\mathbf{T}\rvert\}$. The end of a time period t will be referred as the time point t. With slight abuse of notation, define time point 0 as the beginning of the first time period.

$\mathbf{P}^{\max} \in \boldsymbol{\mathcal{P}}^{\max}$ A collection of certain product subsets, for use in the new tightening Constraints (36)

Parameters in problem statement

$\delta_p^{\max}$ Maximum demand for product p throughout the scheduling horizon

$\delta_{p,t}^{\min}$ Minimum accumulated demand for product p by time point t

ξ_s Available amount of stream s throughout the scheduling horizon

$\pi_{s,q}$ Quantity of property q of stream s

$\pi_q^{\max}$ $:= \max\{\pi_{s,q} : \forall s \in \mathbf{S}\}$

$\pi_{p,q}^{\mathrm{U}}$ Quantity upper bound of property q of product p

$\pi_{p,q}^{\mathrm{L}}$ Quantity lower bound of property q of product p

α_s Unit cost of stream s

β_p Unit revenue of product p

ω_b Capacity of blender b

$\zeta_{b,p}^{\mathrm{U}}$ Flowrate upper bound from blender b to product sink p

$\zeta_{b,p}^{\mathrm{L}}$ Flowrate lower bound from blender b to product sink p

$\tilde{\zeta}_{s,b}^{\mathrm{U}}$ Flowrate upper bound from stream source s to blender b

$\gamma_{b,p}$ Fixed cost for transferring material from blender b to product sink p

$\tilde{\gamma}_{s,b}$ Fixed cost for transferring material from stream source s to blender b

$\eta_{s,b}$ Initial inventory of stream s in blender b

Parameters for bound tightening

$\theta^{\mathrm{L}}_{s,p}$ Lower bound for $I_{s,b,t-1}$ if $X_{b,p,t} = 1$

$\theta^{\mathrm{U}}_{s,p}$ Upper bound for $I_{s,b,t-1}$ if $X_{b,p,t} = 1$

$\upsilon^{\mathrm{L}}_{s,p}$ Lower bound for $C_{s,b,t-1}$ if $X_{b,p,t} = 1$

$\upsilon^{\mathrm{U}}_{s,p}$ Upper bound for $C_{s,b,t-1}$ if $X_{b,p,t} = 1$

$\hat{\delta}^{\min}_{p,t}$ Tightened product minimum demand that satisfies $\hat{\delta}^{\min}_{p,t} \geq \delta^{\min}_{p,t}$

$\check{\delta}^{\max}_{p}$ Tightened product maximum demand that satisfies $\check{\delta}^{\max}_{p} \leq \delta^{\max}_{p}$

$\phi_{p,t}$ Minimum upper bound of all continuous subranges of $\sum_{b\in\mathbf{B}} \sum_{t'\leq t} F_{b,p,t'}$

λ_p Maximum lower bound of all continuous subranges of $\sum_{b\in\mathbf{B}} \sum_{t\in\mathbf{T}} F_{b,p,t}$

$\tau_{s,t}$ Lower bound of stream s entering all blenders prior to time point t

$\psi_{s,t}$ Minimum upper bound of all continuous subranges of $\sum_{b\in\mathbf{B}} \sum_{t'<t} \tilde{F}_{s,b,t'}$

Nonnegative continuous variables

$I^{\mathrm{S}}_{s,b,t}$ Inventory of stream s in blender b at time point t

$I_{b,t}$ Total inventory of blender b at time point t

$\tilde{F}_{s,b,t}$ Flowrate from stream source s to blender b during time period t

$F^{\mathrm{S}}_{s,b,p,t}$ Flowrate of stream s from blender b to product sink p during time period t

$F_{b,p,t}$ Total flowrate from blender b to product sink p during time period t

$C_{s,b,t}$ Proportion of stream s in blender b at time point t

$R_{b,p,t}$ Split fraction from blender b to product p during time period t

Binary variables

$\tilde{X}_{s,b,t}$ 1 if stream source s is charging blender b during time period t

$X_{b,p,t}$ 1 if blender b is delivering product p during time period t

Auxiliary variables for computing the tightening parameters

$\bar{I}_{s,p}$ Amount of stream s used for producing product p within the range $[\min\{\zeta^{\mathrm{L}}_{b,p} : b \in \mathbf{B}\}, \max\{\omega_b : b \in \mathbf{B}\}]$, used for computing $\theta^{\mathrm{L}}_{s,p}$ and $\theta^{\mathrm{U}}_{s,p}$

$\bar{C}_{s,p}$ Proportion of stream s in product p, used for computing $\upsilon^{\mathrm{L}}_{s,p}$ and $\upsilon^{\mathrm{U}}_{s,p}$

$\bar{F}_{s,p,t}$ Amount of stream s used for producing product p by time point t, used for computing $\chi_{s,t}$

Models

$\mathrm{M}^{\mathrm{P(S)}}$ The proportion-based (P-)formulation or the split-fraction-based (S-)formulation

$\mathrm{RM}^{\mathrm{P(S)}}$ The convex relaxation of the P- or S-formulation

$\mathrm{RM}^{\mathrm{P}}_{\mathrm{x}}$ A model consisting of RM^{P} and a tightening constraint x

$\mathrm{b(g)M}^{\mathrm{P(S)}}$ Solving the P- or S- formulation using the solver BARON (b) or GUROBI (g)

$\mathrm{b(g)M}^{\mathrm{P(S)}}_{\mathrm{x}}$ A model consisting of $\mathrm{b(g)M}^{\mathrm{P(S)}}$ and a tightening constraint x

References

(1) Chen, Y.; Maravelias, C. T. Preprocessing algorithm and tightening constraints for multiperiod blend scheduling: cost minimization. *Journal of Global Optimization* **2020**, *77*, 603–625.

(2) Kolodziej, S. P.; Grossmann, I. E.; Furman, K. C.; Sawaya, N. W. A discretization-based approach for the optimization of the multiperiod blend scheduling problem. *Computers & Chemical Engineering* **2013**, *53*, 122–142.

(3) Misener, R.; Floudas, C. A. Advances for the pooling problem: Modeling, global optimization, and computational studies. *Applied and Computational Mathematics* **2009**, *8*, 3–22.

(4) Fragkogios, A.; Saharidis, G. K. Modeling and solution approaches for crude oil scheduling in a refinery. In *Energy Management—Collective and Computational Intelligence with Theory and Applications*; Kahraman, C., Kayakutlu, G., Eds.; Springer: Cham, 2018; pp 251–275.

(5) Cerdá, J.; Pautasso, P. C.; Cafaro, D. C. Optimizing gasoline recipes and blending operations using nonlinear blend models. *Industrial & Engineering Chemistry Research* **2016**, *55*, 7782–7800.

(6) Neiro, S. M.; Murata, V. V.; Pinto, J. M. Hybrid time formulation for diesel blending and distribution scheduling. *Industrial & Engineering Chemistry Research* **2014**, *53*, 17124–17134.

(7) Blom, M. L.; Pearce, A. R.; Stuckey, P. J. A decomposition-based algorithm for the scheduling of open-pit networks over multiple time periods. *Management Science* **2016**, *62*, 3059–3084.

(8) Boland, N.; Kalinowski, T.; Rigterink, F.; Savelsbergh, M. A special case of the generalized pooling problem arising in the mining industry. *Optimization Online e-prints* **2016**, Available from `https://optimization-online.org/wp-content/uploads/2015/07/5025.pdf`.

(9) Gouws, J. F.; Majozi, T. Usage of inherent storage for minimisation of wastewater in multipurpose batch plants. *Chemical Engineering Science* **2009**, *64*, 3545–3554.

(10) Lu, B.; Huang, S.; Grossmann, I. E. Optimal synthesis and operation of wastewater treatment process with dynamic influent. *Industrial & Engineering Chemistry Research* **2017**, *56*, 8663–8676.

(11) Cheng, P.; Garcia-Herreros, P.; Lalpuria, M.; Grossmann, I. E. Optimal scheduling of copper concentrate operations under uncertainty. *Computers & Chemical Engineering* **2020**, *140*, 106919.

(12) Song, Y.; Menezes, B. C.; Garcia-Herreros, P.; Grossmann, I. E. Scheduling and feed quality optimization of concentrate raw materials in the copper refining industry. *Industrial & Engineering Chemistry Research* **2018**, *57*, 11686–11701.

(13) Beach, B.; Hildebrand, R.; Ellis, K.; Lebreton, B. An approximate method for the optimization of long-horizon tank blending and scheduling operations. *Computers & Chemical Engineering* **2020**, *141*, 106839.

(14) Dewitt, C. W.; Lasdon, L. S.; Waren, A. D.; Brenner, D. A.; Melhem, S. A. OMEGA: An improved gasoline blending system for Texaco. *Interfaces* **1989**, *19*, 85–101.

(15) Kelly, J. D.; Mann, J. L. Crude oil blend scheduling optimization: An application with multimillion dollar benefits. *Hydrocarbon Processing* **2003**, *82*, 47–53.

(16) Li, J.; Misener, R.; Floudas, C. A. Continuous-time modeling and global optimization approach for scheduling of crude oil operations. *AIChE Journal* **2012**, *58*, 205–226.

(17) Floudas, C. A.; Lin, X. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers & Chemical Engineering* **2004**, *28*, 2109–2129.

(18) Su, L.; Bernal, D. E.; Grossmann, I. E.; Tang, L. Modeling for integrated refinery planning with crude-oil scheduling. *Chemical Engineering Research and Design* **2023**, *192*, 141–157.

(19) Falk, J. E.; Soland, R. M. An algorithm for separable nonconvex programming problems. *Management Science* **1969**, *15*, 550–569.

(20) McCormick, G. P. Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. *Mathematical Programming* **1976**, *10*, 147–175.

(21) Chen, Y.; Maravelias, C. T. Variable Bound Tightening and Valid Constraints for Multiperiod Blending. *INFORMS Journal on Computing* **2022**, *34*, 2073–2090.

(22) Castro, P. M. Tightening piecewise McCormick relaxations for bilinear problems. *Computers & Chemical Engineering* **2015**, *72*, 300–311.

(23) Misener, R.; Thompson, J. P.; Floudas, C. A. APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. *Computers & Chemical Engineering* **2011**, *35*, 876–892.

(24) Vielma, J. P.; Nemhauser, G. L. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming* **2011**, *128*, 49–72.

(25) Wicaksono, D. S.; Karimi, I. A. Piecewise MILP under-and overestimators for global optimization of bilinear programs. *AIChE Journal* **2008**, *54*, 991–1008.

(26) Andrade, T.; Oliveira, F.; Hamacher, S.; Eberhard, A. Enhancing the normalized multiparametric disaggregation technique for mixed-integer quadratic programming. *Journal of Global Optimization* **2019**, *73*, 701–722.

(27) Castro, P. M. Normalized multiparametric disaggregation: an efficient relaxation for mixed-integer bilinear problems. *Journal of Global Optimization* **2016**, *64*, 765–784.

(28) Cheng, X.; Li, X. Discretization and global optimization for mixed integer bilinear programming. *Journal of Global Optimization* **2022**, *84*, 843–867.

(29) Castro, P. M.; Liao, Q.; Liang, Y. Comparison of mixed-integer relaxations with linear and logarithmic partitioning schemes for quadratically constrained problems. *Optimization and Engineering* **2021**, 1–31.

(30) Anstreicher, K. M.; Burer, S.; Park, K. Convex hull representations for bounded products of variables. *Journal of Global Optimization* **2021**, *80*, 757–778.

(31) Belotti, P.; Miller, A. J.; Namazifar, M. Linear inequalities for bounded products of variables. *SIAG/OPT Views-and-News* **2011**, *22*, 1–8.

(32) Chen, Y.; Maravelias, C. T. Tightening methods based on nontrivial bounds on bilinear terms. *Optimization and Engineering* **2022**, *23*, 1217–1254.

(33) Dey, S. S.; Kocuk, B.; Santana, A. Convexifications of rank-one-based substructures in QCQPs and applications to the pooling problem. *Journal of Global Optimization* **2020**, *77*, 227–272.

(34) Kimizuka, M.; Kim, S.; Yamashita, M. Solving pooling problems with time discretization by LP and SOCP relaxations and rescheduling methods. *Journal of Global Optimization* **2019**, *75*, 631–654.

(35) Luedtke, J.; d'Ambrosio, C.; Linderoth, J.; Schweiger, J. Strong convex nonlinear relaxations of the pooling problem. *SIAM Journal on Optimization* **2020**, *30*, 1582–1609.

(36) Lotero, I.; Trespalacios, F.; Grossmann, I. E.; Papageorgiou, D. J.; Cheon, M.-S. An MILP-MINLP decomposition method for the global optimization of a source based model of the multiperiod blending problem. *Computers & Chemical Engineering* **2016**, *87*, 13–35.

(37) Sherali, H. D.; Adams, W. P. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*; Springer: Dordrecht, 1999.

(38) Tawarmalani, M.; Sahinidis, N. V. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*; Springer: Dordrecht, 2002.

(39) Karuppiah, R.; Furman, K. C.; Grossmann, I. E. Global optimization for scheduling refinery crude oil operations. *Computers & Chemical Engineering* **2008**, *32*, 2745–2766.

(40) Papageorgiou, D. J.; Toriello, A.; Nemhauser, G. L.; Savelsbergh, M. W. Fixed-charge transportation with product blending. *Transportation Science* **2012**, *46*, 281–295.

(41) D'Ambrosio, C.; Linderoth, J.; Luedtke, J. Valid inequalities for the pooling problem with binary variables. In *Integer Programming and Combinatoral Optimization*; Günlük, O., Woeginger, G. J., Eds.; Springer: Berlin, 2011; pp 117–129.

(42) Padberg, M. W.; van Roy, T. J.; Wolsey, L. A. Valid linear inequalities for fixed charge problems. *Operations Research* **1985**, *33*, 842–861.

(43) Kesavan, P.; Allgor, R. J.; Gatzke, E. P.; Barton, P. I. Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs. *Mathematical Programming* **2004**, *100*, 517–535.

(44) Li, X.; Chen, Y.; Barton, P. I. Nonconvex generalized benders decomposition with piecewise convex relaxations for global optimization of integrated process design and operation problems. *Industrial & Engineering Chemistry Research* **2012**, *51*, 7287–7299.

(45) Sahinidis, N. V. BARON 23.1.5: Global Optimization of Mixed-Integer Nonlinear Programs, User's Manual. 2023; Available from `https://www.minlp.com/downloads/docs/baron\%20manual.pdf`.

(46) Gurobi Optimization, LLC Gurobi Optimizer Reference Manual. 2023; Available from `https://www.gurobi.com/documentation/10.0/`.

(47) Cheng, X.; Li, X. A multi-commodity flow formulation for the optimal design of wastewater treatment networks. *Computers & Chemical Engineering* **2020**, *134*, 106681.

(48) Cheng, X.; Li, X. On the P-formulation and the split-fraction-formulation for the generalized pooling problem. *Computers & Chemical Engineering* **2022**, *164*, 107893.

(49) Castro, P. M. New MINLP formulation for the multiperiod pooling problem. *AIChE Journal* **2015**, *61*, 3728–3738.

(50) Velez, S.; Sundaramoorthy, A.; Maravelias, C. T. Valid inequalities based on demand propagation for chemical production scheduling MIP models. *AIChE Journal* **2013**, *59*, 872–887.

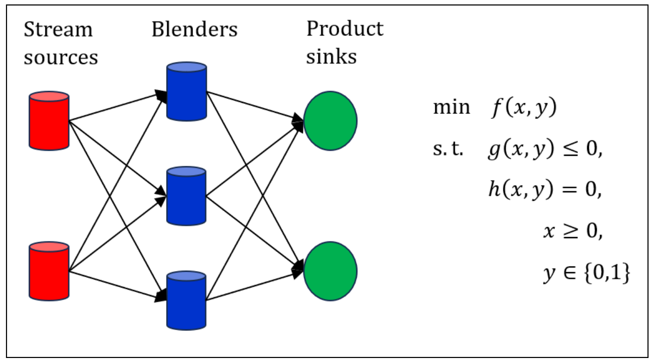
(51) GAMS GAMS documentation 42. 2023; Available from `https://www.gams.com/42/docs/index.html`.

(52) IBM-ILOG V22.1.1: User's Manual for CPLEX. 2022; Available from `https://www.ibm.com/docs/en/icos/22.1.1?topic=optimizers-users-manual-cplex`.

(53) Dolan, E. D.; Moré, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming* **2002**, *91*, 201–213.

(54) Karia, T.; Adjiman, C. S.; Chachuat, B. Assessment of a two-step approach for global optimization of mixed-integer polynomial programs using quadratic reformulation. *Computers & Chemical Engineering* **2022**, *165*, 107909.

TOC Graphic


Stream sources
Blenders
Product sinks
min f(x, y)
s. t. g(x, y) ≤ 0,
h(x, y) = 0,
x ≥ 0,
y ∈ {0,1}