

Mixed-Integer Nonlinear Optimization Methods for Pooling and Multiperiod Blending Problems

By

Yifu Chen

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Chemical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2021

Date of final oral examination: 01/25/2021

The dissertation is approved by the following members of the Final Oral Committee:

Christos Maravelias, Professor, Chemical and Biological Engineering

Jeffrey Linderth, Professor, Industrial and Systems Engineering

James Luedtke, Professor, Industrial and Systems Engineering

Ross Swaney, Associate Professor, Chemical and Biological Engineering

Abstract

Pooling and multiperiod blending problems are common in many industrial sectors, from oil refining to mining and wastewater management. Global optimization for such problems remains challenging due to the presence of bilinear terms. Binary variables may also be introduced to model certain operational constraints.

To address the computational challenges, we develop mixed-integer nonlinear optimization methods for such problems. We first consider the multiperiod blending problem with minimizing cost objective. We develop a novel preprocessing algorithm to calculate lower bounds on stream flows. We define product dedicated flow variables to address product specific features involved in multiperiod blending problem. The bounds on stream flows and new product dedicated flow variables are then used to generate tightening constraints.

For multiperiod blending problem with maximizing profit objective, we first propose a reformulation of the constraints involving bilinear terms using lifting. We introduce an algorithm that returns tight bounds on the lifted variables calculated by aggregating multiple constraints. We propose valid constraints derived from Reformulation-Linearization Technique that utilize the bounds on the lifted variables to further tighten the formulation.

Finally, we develop tightening and solution methods based on nontrivial bounds on bilinear terms. We derive a family of valid linear constraints and further 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 and test them on the pooling problem.

Acknowledgments

First, I would like to express my sincere gratitude to my advisor, Professor Christos Maravelias, for his guidance over the years, especially on conducting research, presenting results, and becoming a scholar.

I want to also thank my dissertation committee members, Professor Jeffrey Lindereth, Professor James Luedtke, and Professor Ross Swaney, for their time reading my thesis, attending meetings, and providing valuable feedbacks.

Many thanks to former and current members of the Maravelias Group for the friendly atmosphere.

I enjoyed the time with my friends in Madison, as well as my internships with ExxonMobil and Amazon, which inspired some of my research ideas. Finally, I really appreciate the emotional support from my parents.

Table of Contents

Abstract	i
Acknowledgments.....	ii
List of Figures	vi
List of Tables	viii
List of Symbols	ix
Chapter 1 Introduction.....	1
1.1 Pooling and multiperiod blending problems	1
1.2 Solution and tightening methods.....	2
1.3 Variable bounds tightening methods	4
1.4 Thesis outline	5
Chapter 2 Preprocessing algorithm and tightening constraints for multiperiod blending: cost minimization.....	7
2.1 Problem statement.....	8
2.2 MINLP and MILP models.....	9
2.3 Motivating example.....	12
2.4 Preprocessing algorithm.....	15
2.4.1 Demand for “good” streams.....	16
2.4.2 Demand updating.....	17
2.4.3 Complete algorithm.....	20
2.5 Product dedicated flow.....	22
2.6 Valid constraints.....	23
2.6.1 Valid constraints with flow variables only	23
2.6.2 Valid constraints with binary variables.....	24
2.6.3 Specifications for product dedicated flows	25
2.7 Computational results.....	26
2.7.1 Problem instances.....	27
2.7.2 Case study	27
2.7.3 Results for MINLP models	29
2.7.4 Results for MILP models.....	30
2.8 Conclusion	31
Chapter 3 Variable bound tightening and valid constraints for multiperiod blending	32

3.1 Reformulation of bilinear terms.....	33
3.2 Preprocessing method for variable bounds tightening.....	33
3.2.1 Bounds tightening using a pair of constraints	35
3.2.2 Bounds updating.....	38
3.2.3 Complete procedure for bound tightening.....	46
3.3 Valid constraints.....	47
3.4 Computational results.....	49
3.4.1 Case study	49
3.4.2 MINLP models	51
3.4.3 MILP models.....	52
3.4.4 Decomposition method.....	53
3.5 Conclusion	55
Chapter 4 Tightening methods based on nontrivial bounds on bilinear terms.....	56
4.1 Introduction	56
4.2 Background	58
4.2.1 Problem statement	58
4.2.2 Nonlinear models for the pooling problem.....	59
4.2.3 Nontrivial bounds on bilinear terms	62
4.2.4 Convex relaxation of bilinear terms.....	65
4.3 Valid constraints.....	66
4.3.1 A family of valid constraints	66
4.3.2 Generation of strong valid constraints	70
4.4 Solution methods	72
4.4.1 Methods for model with only continuous variables	72
4.4.2 Methods for model with semi-continuous variables.....	81
4.5 Computational results.....	85
4.5.1 Models with only continuous variables.....	85
4.5.2 Model with semi-continuous variables.....	89
4.6 Conclusion	90
Chapter 5 Summary.....	91
Appendix	93
A1 Explanations to Chapter 3	93
A1.1 Solving LP3	93

A1.2 Illustrative example	94
A2 Explanations to Chapter 4	98
A2.1 Solving the minimum distance problem	98
A2.2 Details of B&C algorithm.....	101
A2.3 B&B algorithm.....	102
Bibliography	103

List of Figures

Figure 2-1. Illustrative graph showing flows for one period.....	9
Figure 2-2. Illustrative graph for the feasible space of MILP models based on Kolodziej et al.(S. P. Kolodziej, Castro, and Grossmann 2013)	12
Figure 2-3. Motivating example with one period.....	13
Figure 2-4. Illustrative graph for lower bounding flow for S1 (Eqn. (2-18)) and enforcing specification for flows (Eqn. (2-19)).	15
Figure 2-5. Illustrative example for demand updating via algebraic equations; pattern filled bars indicate feasible property domains.	18
Figure 2-6. Illustrative example for demand updating via solving LP.....	20
Figure 2-7. Flowchart of preprocessing algorithm for each product.....	21
Figure 2-8. Illustrative example for the preprocessing algorithm with one product with $\omega = 1$ (index k is dropped for simplicity).	22
Figure 2-9. Illustrative graph for product dedicated flows.....	23
Figure 2-10. (A). Network configuration for the case study (dashed lines indicate connectivity between streams, blenders, and products). (B). Gantt chart for an optimal solution.	27
Figure 2-11. Performance profiles for concentration-based model (A) and source-based model (B).	30
Figure 2-12. Performance profiles for linear relaxation of the concentration-based model (A) and source-based model (B).	31
Figure 3-1. An illustrative example for parameters.....	35
Figure 3-2. An optimal schedule for Instance 7.....	50
Figure 3-3. Inventory profile for the schedule shown in Figure 3-2.	50
Figure 3-4. Performance profile for different MINLP models.....	52
Figure 3-5. Performance profile for two MILP models with $\delta = 0.01$ (left) and $\delta = 0.1$ (right)	53
Figure 4-1. Illustrative graph for bilinear terms xy with $x \in 1,3$, $y \in 1/3,1$ and its relaxation when the nontrivial upper bound $w = 2$ is active.....	66
Figure 4-2. Illustrative graph for bilinear terms $w = xy$ with $x \in 1,3$, $y \in 1/3,1$ when one of its nontrivial bounds $w = 2$ or $w = 1$ is active.....	69
Figure 4-3. The optimal solution to the illustrative example from solving the first relaxation	75
Figure 4-4. Tightening constraint for the illustrative example.....	76

Figure 4-5. Flowchart of the customized B&C algorithm.....	80
Figure 4-6. Illustrative graph for points of tangency on bilinear curve generated from Eqn. (4-50) with seven intervals (indices j and k are dropped in the graph for simplicity).....	81
Figure 4-7. Performance profile for model with only continuous variable and its variants solved with BARON (left) and SCIP (right)	86
Figure 4-8. Performance profile for model based on pq-formulation and its variants solved with BARON with maximizing profit objective.	88
Figure 4-9. Performance profile for model with semi-continuous variables and its variants solved with BARON with maximizing profit objective.	89
Figure 4-10. Performance profile for model with semi-continuous variables and its variants solved with BARON with minimizing cost objective.	90

List of Tables

Table 2-1. Major constraints in the concentration model for the motivating example	13
Table 2-2. Relaxation of Eqn. (2-6) using McCormick envelopes for the motivating example	14
Table 2-3. Valid constraints for the example shown in Figure 2-8 (index k is dropped).....	26
Table 2-4. Size of tested instances	27
Table 2-5. Parameters for streams and products for the case study.....	28
Table 2-6. Model statistics for the case study.....	28
Table 2-7. Model description	29
Table 2-8. CPU time in seconds for the case study for concentration-based model	29
Table 2-9. CPU time in seconds for the case study for source-based model	29
Table 2-10. CPU time for the case study with different linear models	29
Table 2-11. Percentages of instances solved to global optimality in 2 hours	30
Table 3-1. Bounds calculated by aggregating pair of constraints.....	38
Table 3-2. Bounds calculated by different methods	46
Table 3-3. Model description	49
Table 3-4. Model and solution statistics for Instance 7	50
Table 3-5. Size of tested instances and CPU time for different MINLP models	51
Table 3-6. Size of Instance 16 - 20	53
Table 3-7. Size of tested instances and CPU time for MINLP and decomposition.....	54
Table 4-1. Parameters for the illustrative example	75
Table 4-2. Solution statistics for B&B and B&C algorithms over select instances	87

List of Symbols

Indices/Sets

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

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

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

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

$t \in \mathbf{T}$: Time points: $\{0, 1, \dots, |\mathbf{T}|\}$ /time periods: $\{1, 2, \dots, |\mathbf{T}|\}$

Subsets

$\mathbf{S}_{kl}^{\mathbf{U}}$: Streams that satisfy the upper bound on property l for product k

$\mathbf{S}_{kl}^{\mathbf{L}}$: Streams that satisfy the lower bound on property l for product k

$\mathbf{L}_{ik}$: Properties for product k whose specification is violated by stream i

$\mathbf{L}_{ik}^{\mathbf{S}}$: Properties for which stream i is the only stream that satisfies the specification for product k

$\mathbf{L}_k^{\mathbf{M}}$: Properties for which multiple (but not all) streams satisfy the specification for product k

$\mathbf{L}^{\mathbf{L}}$: Set of properties that have lower bounding specification

$\mathbf{L}^{\mathbf{U}}$: Set of properties that have upper bounding specification

Parameters for multiperiod blending problem

Problem data

β_k : Price of product k

$\gamma_i^{\mathbf{I}}$: Inventory capacity for stream i

γ_j : Inventory capacity for blender j

$\gamma_k^{\mathbf{K}}$: Inventory capacity for product k

δ_{pt} : Amount of product k due at time point t

ξ_{it} : Supply for stream i at time point t

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

$\pi_{kl}^{\mathbf{U}}$: Upper bounding specification on property l for product k

$\pi_{kl}^{\mathbf{L}}$: Lower bounding specification on property l for product k

ω_k : Maximum(cumulative) demand for product k

Parameters calculated by preprocessing algorithm (for minimizing cost objective)

$\hat{\pi}_{kl}^{\mathbf{U}}$: Value of property l that violates the upper bound on product k by the least margin.

$\hat{\pi}_{kl}^{\mathbf{L}}$: Value of property l that violates the lower bound on product k by the least margin.

θ_{kl} : (Estimated) Value of property l for product k

$\hat{\omega}_{ik}$: Demand for stream i derived from product k

$\bar{\omega}_{ikl}$: Demand for stream i derived from property l for product k

$\bar{\omega}'_{ikl}$: Updated demand for stream i derived from property l for product k

Parameters calculated by preprocessing algorithm (for maximizing profit objective)

$\hat{\gamma}_{ijkl}$: Tightened bound on inventory of stream i in blender j when it is feeding product k derived from property l

$\bar{\gamma}_{ijk}$: Tightened bound on inventory of stream i in blender j when it is feeding product k

μ_{ikl} : Violation of specification for property l for product k from stream i

μ_l^* : Value of property l of the “best” stream for property l

μ_l^+ : Value of property l of the “second best” stream for property l

Parameters for pooling problem

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

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

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

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

φ_k : Minimum demand for product k

ω_k : Maximum demand for product k

Chapter 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; S. P. Kolodziej, Castro, and Grossmann 2013; Gupte et al. 2017) 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; J. D. Kelly and Mann 2003).

1.1 Pooling and multiperiod blending problems

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). It can be briefly stated as follows: multiple streams with different properties are blended in pools before sent to produce products. The combined flows from pools to a product must meet the corresponding specifications.

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.

While the pooling problem does not account for time varying supply of streams and demand for products, in practice, such features are important. The aforementioned features give rise to the multiperiod blending problem, where we not only make decisions on the proportion of streams to be blended in the pools, but also when to send streams to pools, and when to withdraw products from pools. In the multiperiod setting, binary variables are often introduced to enforce additional operating rules, leading to a nonconvex Mixed-Integer Nonlinear Program (MINLP).

1.2 Solution and tightening methods

Researchers have proposed novel ways to tackle the bilinear terms in the context of pooling through discretization (Gupte et al. 2013; S. Kolodziej, Castro, and Grossmann 2013), piecewise linear approximation (Meyer and Floudas 2006; Wicaksono and Karimi 2008; Misener, Thompson, and Floudas 2011; Misener and Floudas 2012), as well as identifying parametric structure (Ceccon, Kouyialis, and Misener 2016; Baltean-Lugoian and Misener 2017). A heuristic and two global optimization algorithms based on discretization of variables involved in bilinear terms have been presented by Kolodziej et al. (S. P. Kolodziej et al. 2013). An inventory pinch based algorithm for gasoline blending planning and scheduling has been proposed by Castillo et al. (Castillo, Mahalec, and Kelly 2013; Castillo and Mahalec 2014b; 2014a). A branch-and-bound algorithm for global optimization of crude

oil unloading and blending operations, based on refining the solution pool obtained from piecewise linear approximation, has been proposed by Li et al. (Li, Misener, and Floudas 2012). Decomposition-based heuristics and algorithms for the scheduling of open-pit networks, which includes blending of different grades of minerals, have been proposed by Blom et al. (Blom et al. 2014; Blom, Pearce, and Stuckey 2016). A model based on floating time slots for gasoline blend scheduling has been proposed by Cerdá et al. (Cerdá, Pautasso, and Cafaro 2016). Finally, a successive approximation method to handle intensive properties in blending process, was proposed by Kelly et al. (Jeffrey D. Kelly, Menezes, and Grossmann 2018).

Tightening methods for nonconvex optimization problems with bilinear terms have been studied extensively. 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 2015a), Dey and Gupte analyzed mixed-integer linear programming (MILP) techniques to address bilinear terms (Dey and Gupte 2015). 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).

Tightening methods based on strong valid inequalities and reformulations have been proven to be effective in addressing industrial-scale chemical production scheduling instances (Burkard and Hatzl 2005; Janak and Floudas 2008; Velez, Sundaramoorthy, and Maravelias 2013; Velez and Maravelias 2013a; 2013b; Merchan, Velez, and Maravelias 2013; Merchan,

Lee, and Maravelias 2016). Compared to the abundant studies focusing on formulations and solution methods, valid inequalities for blending process have received less attention. Papageorgiou et al. (Papageorgiou et al. 2012) studied the fixed-cost transportation problem with product blending. The problem was formulated as a mixed-integer linear programming (MILP) model considering one property, and facet-defining inequalities were introduced. D'Ambrosio et al. (D'Ambrosio, Linderoth, and Luedtke 2011) studied the pooling problem with binary variables and proposed four classes of valid inequalities derived from a mixed-integer linear relaxation of the problem. Both works exploited product specification.

1.3 Variable bounds tightening methods

Global optimization of nonconvex optimization problem is performed using branch-and-bound algorithms which involve solving convex relaxations of the original problem. The tightness of the convex relaxation plays an important role in the performance of the algorithms.

The tightness of the convex relaxation strongly depends on variable bounds. Various bounds tightening methods have been proposed (Belotti et al. 2009; Puranik and Sahinidis 2017), including, for example, methods based on reduced cost (Ryoo and Sahinidis 1996), which utilizes the optimal solution to the relaxed problem. Bounds tightening techniques that do not require such information have also been proposed. A well-known technique is Optimality Based Bound Tightening (OBBT) which typically relies on solving linear programs (LP) (Quesada and Grossmann 1995; Maranas and Floudas 1997; Sheckman and Sahinidis 1998; Smith and Pantelides 1999). OBBT can be computationally expensive, and methods aim to increase its efficiency have been studied (Gleixner et al. 2017). Feasibility Based Bound

Tightening (FBBT), which considers a single constraint at a time and utilizes interval arithmetic to infer variable bounds, has been employed in solving both MILP (Savelsbergh 1994; Achterberg et al. 2020) and MINLP (Achterberg 2007). FBBT has received considerable attention in both mathematical programming and artificial intelligence communities (Street 1989) Though computationally inexpensive, FBBT is known to be less effective compared to OBBT in terms of the tightness of the bounds found.

Tightening methods that utilize information from multiple constraints at a time have also been studied. For example, Achterberg et al. (Achterberg et al. 2020) studied presolve methods for MILP that consider multiple constraints simultaneously. Specifically, for variable bounds tightening purpose, their methods are based on special block structure in the problem matrix. Domes and Neumaier (Domes and Neumaier 2016) proposed constraint aggregation method for rigorous global optimization that utilizes information from local solutions. Belotti (Belotti 2013) proposed a procedure that infers variable bounds using a pair of constraints. Aggregating multiple constraints can lead to tighter variable bounds compared to FBBT, while it is computationally inexpensive compared to OBBT. However, which constraints to be aggregated and their weights require further investigation.

1.4 Thesis outline

This thesis focuses on solution methods, in particular tightening methods, for pooling and multiperiod blending problem. In Chapter 2, we first consider the multiperiod blending problem with minimizing cost objective and present tightening methods based on stream properties, product demand and specifications (contents of this chapter are from our published paper (Chen and Maravelias 2020)). In Chapter 3, we consider a variable bound

tightening method for multiperiod blending problem which incorporates the understanding of the physical system (contents of this chapter are from our submitted manuscript that is currently under revision (Chen and Maravelias 2021b)). In Chapter 4, we present tightening methods based on nontrivial bounds on bilinear terms for the pooling problem (contents of this chapter are from our submitted manuscript that is currently under review (Chen and Maravelias 2021a)). In Chapter 5, we summarize the thesis. Throughout the thesis, 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.

Chapter 2

Preprocessing algorithm and tightening constraints for multiperiod blending: cost minimization

Scheduling problems containing blending processes arise in many industries (Baker and Lasdon 1985; Blom et al. 2014). The multiperiod blending problem considers time varying stream availability and product demand. To some extent, multiperiod blending problem can be viewed as the scheduling extension of the pooling problem (S. P. Kolodziej et al. 2013; Lotero et al. 2016), or a time-indexed pooling problem (Gupte et al. 2017). In general, multiperiod blending problem is formulated as a MINLP model, where binary variables are used to enforce operating rules, and nonlinear constraints contain bilinear terms to model property or composition. Several formulations for multiperiod blending problem have been proposed, including a concentration-based model (S. P. Kolodziej et al. 2013) and source-based models (Lotero et al. 2016; Castro 2015b).

The convex relaxation of bilinear terms using McCormick envelopes (McCormick 1976) has been the basis of many global optimization techniques.

In this chapter we present solution methods for multiperiod blending problem focusing on the cost minimization objective. We develop a novel preprocessing algorithm to calculate lower bounds on stream flows. We define product dedicated flow variables to address product specific features involved in multiperiod blending problem. The bounds on stream flows and new product dedicated flow variables are then used to generate tightening

constraints which significantly improve the solution time of the MINLP models as well as models based on linear approximations.

2.1 Problem statement

The problem we consider is defined in terms of the following sets:

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

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

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

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

$t \in \mathbf{T}$: Time points: $\{0, 1, \dots, |\mathbf{T}|\}$ /time periods: $\{1, 2, \dots, |\mathbf{T}|\}$

And can be stated as follows:

Given are:

δ_{pt} : Amount of product k due at time point t

ξ_{it} : Supply for stream i at time point t

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

π_{kl}^U : Upper bounding specification on property l for product k

π_{kl}^L : Lower bounding specification on property l for product k

Our goal is to find a blend schedule with the lowest cost while satisfying product demand and specifications. We assume that all product properties are the average of the properties of the streams blended weighted by volume fraction. We also assume there is no initial inventory in the blenders.

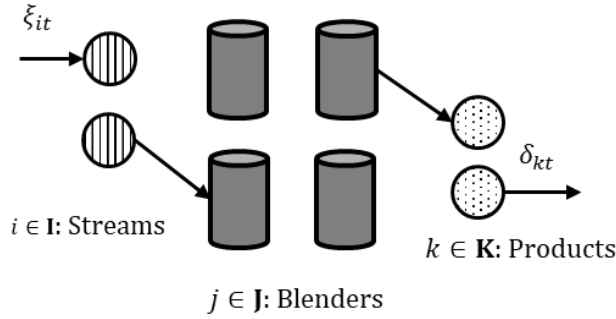


Figure 2-1. Illustrative graph showing flows for one period.

2.2 MINLP and MILP models

We define the following nonnegative continuous variables:

- $\tilde{F}_{ijt}$: Flow from stream i to blender j at time point t
- $F_{j,j',t}$: Flow from blender j to blender j' at time point t
- $\bar{F}_{jkt}$: Flow from blender j to product k at time point t
- $\tilde{I}_{it}$: Inventory of stream i during time period t
- I_{jt} : Inventory in blender j during time period t
- $\bar{I}_{kt}$: Inventory of product p during time period t

Eqns.(2-1) - (2-3) enforce material balances:

$$\tilde{I}_{i,t+1} = \tilde{I}_{it} + \xi_{it} - \sum_j \tilde{F}_{ijt}, \quad i, t \quad (2-1)$$

$$I_{j,t+1} = I_{jt} + \sum_i \tilde{F}_{ijt} + \sum_{j'} F_{j',j,t} - \sum_{j'} F_{j,j',t} - \sum_k \bar{F}_{jkt}, \quad j, t \quad (2-2)$$

$$\bar{I}_{p,t+1} = \bar{I}_{p,t} + \sum_j \bar{F}_{j,p,t} - \delta_{p,t}, \quad p, t \quad (2-3)$$

We also define the following binary variables:

- $\tilde{X}_{ijt}$: = 1 if stream i is fed into blender j at time point t
- $X_{j,j',t}$: = 1 if blender j feeds blender j' at time point t

$\bar{X}_{jkt}$: = 1 if blender j sends product k at time point t

The binary variables listed above allow us to model fixed costs, and are also used to enforce, for example, the operating rule that blender feeding and withdrawing cannot occur simultaneously. Eqns. (2-4) – (2-5) can be used to enforce such rule:

$$\tilde{F}_{ijt} \leq M\tilde{X}_{ijt}, \quad i, j, t \quad (2-4)$$

$$\tilde{X}_{ijt} \leq 1 - \bar{X}_{jkt}, \quad i, j, k, t \quad (2-5)$$

In the concentration-based model, we introduce a nonnegative continuous variable C_{ljt} to model value of property (concentration):

C_{ljt} : Value of property l of the inventory inside blender j during time period t

Eqn. (2-6) keeps track of the “amount” of property within a blender over time:

$$I_{j,t+1}C_{lj,t+1} = I_{jt}C_{ljt} + \sum_i \pi_{il}\tilde{F}_{ijt} + \sum_{j'} C_{l,j',t}F_{j',j,t} - \sum_{j'} C_{ljt}F_{j,j',t} - \sum_k C_{ljt}\bar{F}_{jkt}, \quad l, j, t \quad (2-6)$$

When a product is withdrawn, we enforce the specifications using:

$$\pi_{kl}^L - M(1 - \bar{X}_{jkt}) \leq C_{ljt} \leq \pi_{kl}^U + M(1 - \bar{X}_{jkt}), \quad l, k, j, t \quad (2-7)$$

Eqns. (2-1) - (2-7) comprise the concentration-based formulation, henceforth referred to as M^C .

In the source-based formulation the following nonnegative continuous variables are defined:

$F_{i,j,j',t}^S$: Flow of stream i from blender j to blender j' at time point t

$\bar{F}_{ijkt}^S$: Flow of stream i from blender j to product k at time point t

I_{ijt}^S : Inventory of stream i in blender j during time period t

The above variables should satisfy:

$$F_{j,j',t} = \sum_i F_{i,j,j',t}^S, \quad j, j', t \quad (2-8)$$

$$\bar{F}_{jkt} = \sum_i \bar{F}_{i,j,p,t}^S, \quad j, k, t \quad (2-9)$$

$$I_{j,t} = \sum_i I_{ijt}^S, \quad j, t \quad (2-10)$$

We also enforce the material balance for each stream in each blender:

$$I_{i,j,t+1}^S = I_{ijt}^S + \tilde{F}_{ijt} + \sum_{j'} F_{i,j',j,t}^S - \sum_{j'} F_{i,j,j',t}^S - \sum_k \bar{F}_{ijkt}^S, \quad i, j, t \quad (2-11)$$

When inventory is withdrawn from a blender, all streams are withdrawn at the same ratio:

$$F_{j,j',t} = R_{j,j',t}^J I_{jt}, \quad j, j', t \quad (2-12)$$

$$F_{i,j,j',t}^S = R_{j,j',t}^J I_{ijt}^S, \quad i, j, j', t \quad (2-13)$$

$$\bar{F}_{jkt} = R_{jkt}^P I_{jt}, \quad j, k, t \quad (2-14)$$

$$\bar{F}_{ijkt}^S = R_{jkt}^P I_{ijt}^S, \quad i, j, k, t \quad (2-15)$$

where $R_{j,j',t}^J \in [0,1]$ and $R_{j,p,t}^P \in [0,1]$ represent the ratio of flow over the starting inventory.

Eqn. (2-16) enforces product specifications:

$$\pi_{kl}^L \bar{F}_{jkt} \leq \sum_i \pi_{il} \bar{F}_{ijkt}^S \leq \pi_{kl}^U \bar{F}_{jkt}, \quad l, k, j, t \quad (2-16)$$

Eqns. (2-1) - (2-5), (2-8) - (2-16) comprise the source-based formulation, henceforth referred to as M^{SB} . Eqns. (2-1) - (2-16) are all based on Lotero et al. (Lotero et al. 2016), where more details about M^C , M^{SB} , as well as other alternative MINLP models, can be found.

In this work, we consider the cost minimization objective:

$$\min \sum_j \sum_t \left[\sum_i (\tilde{\alpha}_{ij}^F \tilde{X}_{ijt} + \tilde{\alpha}_{ij}^V \tilde{F}_{ijt}) + \sum_{j'} (\alpha_{j,j'}^F X_{j,j',t} + \alpha_{j,j'}^V F_{j,j',t}) \right] + \sum_k \bar{\alpha}_{jk}^F \bar{X}_{jkt} + \bar{\alpha}_{jk}^V \bar{F}_{jkt} \quad (2-17)$$

with positive fixed and variable cost coefficients.

The MINLP models can be approximated using a radix based discretization. The resulting MILP is guaranteed to return only feasible solutions to the original MINLP. Such MILP is referred to as L_1M in this work. A relaxation of L_1M , referred to as L_2M , is also a relaxation of the original MINLP. More details about the L_1M and L_2M can be found in the paper by

Kolodziej et al. (S. P. Kolodziej et al. 2013), in which they are called MPBP' and MPBPR, respectively.

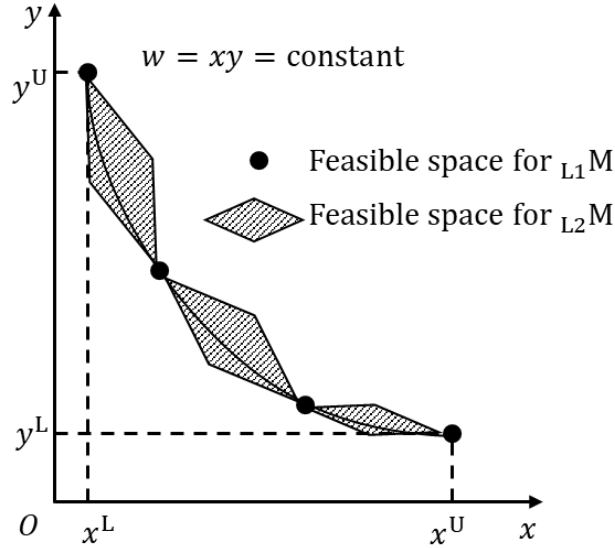


Figure 2-2. Illustrative graph for the feasible space of MILP models based on Kolodziej et al.(S. P. Kolodziej, Castro, and Grossmann 2013)

2.3 Motivating example

Consider the example with two streams (S1 and S2), one product (P1), one property (Q1) and 1 period shown in Figure 2-3. We show the major constraints in M^C for the motivating example in Table 2-1.

When we use M^C to solve the motivating example we obtain an optimal solution with $Z^* = 7.5$, $\tilde{F}_{S1,J1,0}^* = 0.5$, $\tilde{F}_{S2,J1,0}^* = 0.5$. We use McCormick envelopes to relax the bilinear terms in Eqn. (2-6). Let $U_1 = I_{J1,1}C_{Q1,J1,1}$, $U_2 = I_{J1,2}C_{Q1,J1,2}$, and $W_1 = \bar{F}_{J1,P1,1}C_{Q1,J1,1}$. We use the following upper and lower variable bounds: $I_{jt} \in [0, 2]$, $C_{l,jt} \in [0.8, 1]$, and $\bar{F}_{jkt} \in [0, 2]$. The resulting MILP model is referred to as M_cM .

When we use M_cM to solve the motivating example we obtain an optimal solution with $Z^* =$

$$5, \tilde{F}_{S1,J1,1}^* = 0, \tilde{F}_{S2,J1,1}^* = 1.$$

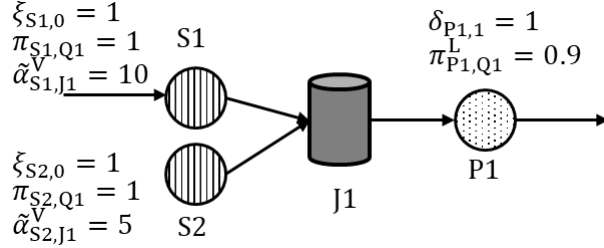


Figure 2-3. Motivating example with one period

Table 2-1. Major constraints in the concentration model for the motivating example

	Objective and constraints in M^C	Description
min	$Z = 10\tilde{F}_{S1,J1,0} + 10\tilde{F}_{S1,J1,1} + 5\tilde{F}_{S2,J1,0} + 5\tilde{F}_{S2,J1,1}$	Objective
s.t	$I_{J1,1} = \tilde{F}_{S1,J1,0} + \tilde{F}_{S2,J1,0} - \bar{F}_{J1,P1,0}$ $I_{J1,2} = I_{J1,1} + \tilde{F}_{S1,J1,1} + \tilde{F}_{S2,J1,1} - \bar{F}_{J1,P1,1}$	Eqn. (2-1)
	$\tilde{I}_{S1,1} = 1 - \tilde{F}_{S1,J1,0}$ $\tilde{I}_{S1,2} = \tilde{I}_{S1,1} - \tilde{F}_{S1,J1,1}$ $\tilde{I}_{S2,1} = 1 - \tilde{F}_{S2,J1,0}$ $\tilde{I}_{S2,2} = \tilde{I}_{S2,1} - \tilde{F}_{S2,J1,1}$	Eqn. (2-2)
	$\bar{I}_{P1,1} = \bar{F}_{J1,P1,0}$ $\bar{I}_{P1,2} = \bar{I}_{P1,1} + \bar{F}_{J1,P1,1} - 1$	Eqn. (2-3)
	$I_{J1,1}C_{Q1,J1,1} = \tilde{F}_{S1,J1,0} + 0.8\tilde{F}_{S2,J1,0}$ $I_{J1,2}C_{Q1,J1,2} = I_{J1,1}C_{Q1,J1,1} + \tilde{F}_{S1,J1,1} + 0.8\tilde{F}_{S2,J1,1} - \bar{F}_{J1,P1,1}C_{Q1,J1,1}$	Eqn. (2-6)
	$C_{Q1,J1,1} \geq 0.9 - M(1 - \bar{X}_{J1,P1,1})$	Eqn. (2-7)

Table 2-2. Relaxation of Eqn. (2-6) using McCormick envelopes for the motivating example

Constraints	Description
$U_1 = \tilde{F}_{S1,J1,0} + 0.8\tilde{F}_{S2,J1,0}$ $U_2 = U_1 + \tilde{F}_{S1,J1,1} + 0.8\tilde{F}_{S2,J1,1} - W_1$	Replacing bilinear terms in Eqn. (2-6) with reformulated variables
$U_1 \geq 0.8I_{J1,1}$ $U_1 \geq 2C_{Q1,J1,1} + I_{J1,1} - 2$ $U_1 \leq 2C_{Q1,J1,1} + 0.8I_{J1,1} - 1.6$ $U_1 \leq I_{J1,1}$	McCormick envelope for U_1
$U_2 \geq 0.8I_{J1,2}$ $U_2 \geq 2C_{Q1,J1,2} + I_{J1,2} - 2$ $U_2 \leq 2C_{Q1,J1,2} + 0.8I_{J1,2} - 1.6$ $U_2 \leq I_{J1,2}$	McCormick envelope for U_2
$W_1 \geq 0.8\bar{F}_{J1,P1,1}$ $W_1 \geq 2C_{Q1,J1,1} + \bar{F}_{J1,P1,1} - 2$ $W_1 \leq 2C_{Q1,J1,1} + 0.8\bar{F}_{J1,P1,1} - 1.6$ $W_1 \leq \bar{F}_{J1,P1,1}$	McCormick envelope for W_1

We next tighten $_{Mc}M$ using valid inequalities based on product demand and specifications.

One observation is that S1 is required to produce P1, since S1 is the only stream satisfying the specification. To produce 1 unit of P1, we need at least 0.5 unit of S1 (since the blend contains 0.5 unit of S1 and 0.5 unit of S2 will satisfy the specification exactly). Thus, we have:

$$\tilde{F}_{S1,J1,0} + \tilde{F}_{S1,J1,1} \geq 0.5 \quad (2-18)$$

which, when added to $_{Mc}M$, yields an optimal solution with $Z^* = 7.5$, $\tilde{F}_{S1,J1,1}^* = 0.5$, $\tilde{F}_{S2,J1,1}^* = 0.5$.

Another idea is to enforce the specification for P1 on the streams fed into the blenders. If we assume, for now, that all streams fed into blenders will be transferred into P1, we can write:

$$\tilde{F}_{S1,J1,0} + \tilde{F}_{S1,J1,1} + 0.8(\tilde{F}_{S2,J1,0} + \tilde{F}_{S2,J1,1}) \geq 0.9(\bar{F}_{J1,P1,0} + \bar{F}_{J1,P1,1}) \quad (2-19)$$

which, when added to $_{Mc}M$, yields an optimal solution with $Z^* = 6$, $\tilde{F}_{S1,J1,1}^* = 0.1$, $\tilde{F}_{S2,J1,1}^* = 1$.

Both Eqn. (2-18) and Eqn. (2-19) cut off the solution obtained from $_{Mc}M$. The effectiveness of those two constraints is illustrated in Figure 2-4, which shows the feasible space for $\tilde{F}_{S1,J1,0}$ and $\tilde{F}_{S2,J1,0}$ when $\tilde{X}_{S1,J1,0} = 1, \tilde{X}_{S2,J1,0} = 1, \tilde{X}_{S1,J1,1} = 0, \tilde{X}_{S2,J1,1} = 0, \bar{X}_{J1,P1,0} = 0, \bar{X}_{J1,P1,1} = 1, \bar{F}_{J1,P1,1} = 1$.

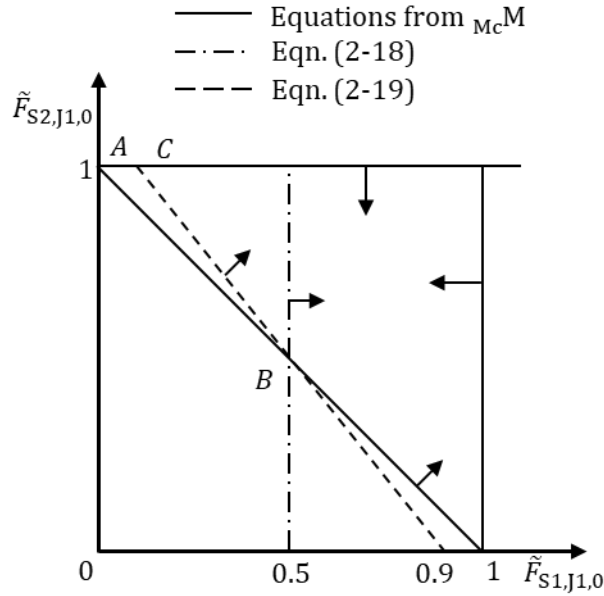


Figure 2-4. Illustrative graph for lower bounding flow for S1 (Eqn. (2-18)) and enforcing specification for flows (Eqn. (2-19)).

2.4 Preprocessing algorithm

We develop a preprocessing algorithm to calculate lower bounds on stream flows based on product demand and specifications. Given a product p and its specification π_{kl}^U/π_{kl}^L , the total flow of streams that satisfy the specification (henceforth referred to as “good” streams) should be positive, as discussed by Greenburg (Greenberg 1995) and Papageorgiou et al. (Papageorgiou et al. 2012). The preprocessing algorithm systematically calculates lower bounds on stream flows, which are then used to generate tightening constraints. We first focus on the case in which we have only one specification.

2.4.1 Demand for “good” streams

Given a product k , a property l , and a lower bounding specification π_{kl}^L , let $\mathbf{S}_{kl}^L$ denote the set of streams that satisfy such specification: $\mathbf{S}_{kl}^L = \{i \in \mathbf{I} \mid \pi_{il} \geq \pi_{kl}^L\}$. We define parameter $\hat{\pi}_{kl}^L$ as: $\hat{\pi}_{kl}^L = \max_{i \in \mathbf{S}_{kl}^L} \{\pi_{il}\}$. The lower bounds on the “good” stream flows can be obtained by considering the blend that contains those “good” streams and the stream that violates the specification by the least margin. Such blend should satisfy the specification exactly. Let ω_k denote the demand for product k , $\bar{\omega}_{ikl}$ denote the demand for stream i derived from ω_k and π_{kl}^L . We have:

$$\sum_{i \in \mathbf{S}_{kl}^L} \pi_{il} \bar{\omega}_{ikl} + \hat{\pi}_{kl}^L \left(\omega_k - \sum_{i \in \mathbf{S}_{kl}^L} \bar{\omega}_{ikl} \right) = \pi_{kl}^L \omega_k, \quad k, l \quad (2-20)$$

In general, we cannot directly propose nonzero demand for individual “good” stream. However, in the special case of one “good” stream, by considering the binary blend of the only “good” stream and the stream that violates the specification by the least margin, we have:

$$\bar{\omega}_{ikl} = \frac{(\pi_{kl}^L - \hat{\pi}_{kl}^L) \omega_k}{(\pi_{il} - \hat{\pi}_{kl}^L)}, \quad i, k, l \in \mathbf{L}_{ik}^S \quad (2-21)$$

where $\mathbf{L}_{ik}^S$ denotes the set of properties for which stream i is the only stream that satisfies the specification for product k .

Similarly, for upper bounding specification π_{kl}^U , the set for “good” streams is $\mathbf{S}_{kl}^U = \{i \in \mathbf{I} \mid \pi_{il} \leq \pi_{kl}^U\}$, and $\hat{\pi}_{kl}^U = \min_{i \in \mathbf{S}_{kl}^U} \{\pi_{il}\}$. We use the following equation, which is similar to Eqn. (2-21), to calculate demand for the only “good” stream:

$$\bar{\omega}_{ikl} = \frac{(\hat{\pi}_{kl}^U - \pi_{kl}^U) \omega_k}{(\hat{\pi}_{kl}^U - \pi_{il})}, \quad i, k, l \in \mathbf{L}_{ik}^U \quad (2-22)$$

We have derived demand for a “good” stream from one specification. In many multiperiod blending instances, there are multiple specifications that we need to consider. We introduce a procedure to update the demand obtained from one specification using other specifications.

2.4.2 Demand updating

Assume that we already have valid $\bar{\omega}_{ikl} > 0$, and let $\hat{\omega}_{ik}$ denote the demand for stream i for product k . If there is only one specification for product k , then it is clear that $\hat{\omega}_{ik} = \bar{\omega}_{ikl}$. Now, assume there is another property l' . From the specification for property l' , we aim to update $\hat{\omega}_{ik}$ to make it greater than $\bar{\omega}_{ikl}$. In general, we can initialize $\hat{\omega}_{ik}$ using $\hat{\omega}_{ik} = \max_l \bar{\omega}_{ikl}$. We will briefly go through several cases where we are able to update $\hat{\omega}_{ik}$.

2.4.2.1. Specifications for different properties

When there are multiple specifications, it is important to note that a “good” stream for one specification may not be a “good” stream for other specifications. Also, in Eqn. (2-21), we obtain the demand for one “good” stream by assuming a certain binary blend. However, such blend may violate other specifications. Our preprocessing algorithm can identify the aforementioned cases and update the previously obtained $\hat{\omega}_{ik}$ accordingly, through algebraic equations and/or solving linear programming (LP) problems.

2.4.2.2. Demand updating via algebraic equations

We first consider an example with three streams (S1 to S3), two properties (Q1 and Q2), and one product (P1) shown in Figure 2-5. Since $\pi_{S1,Q1} > \pi_{P1,Q1}^L > \pi_{S2,Q1} > \pi_{S3,Q1}$, S1 is the only “good” stream for Q1 and its demand can be obtained from Eqn. (2-21) by considering the binary blend of S1 and S2: $\bar{\omega}_{S1,P1,Q1} = 0.25$. Similarly, S3 is the only “good” stream for Q2, and by considering the binary blend of S3 and S2, we calculate: $\bar{\omega}_{S3,P1,Q2} = 0.15$. However,

since S3 is required, the binary blend of S1 and S2 violates the specification for Q2. In other words, we have to use S3, which is a lower quality stream for Q1 compared to S2, and thus will lead to a higher demand for S1. Such demand updating can be done by an algebraic equation which is a modification of Eqn. (2-21). Instead of the binary blend, we now consider the blend that contains all streams with $\hat{\omega}_{ik} > 0$. If $\bar{\omega}'_{ikl}$ denotes the updated demand, Eqn. (2-21) becomes:

$$\bar{\omega}'_{ikl} = \frac{[(\pi_{kl}^L - \hat{\pi}_{kl}^L)\omega_k + \sum_{i' \in S_{kl}^L} (\hat{\pi}_{kl}^L - \pi_{i',l})\hat{\omega}_{i',l}]}{(\pi_{il} - \hat{\pi}_{kl}^L)}, \quad i, k, l \quad (2-23)$$

Once we finish updating the demand for one stream, we update the demand for another stream. We iterate until no further improvement can be achieved.

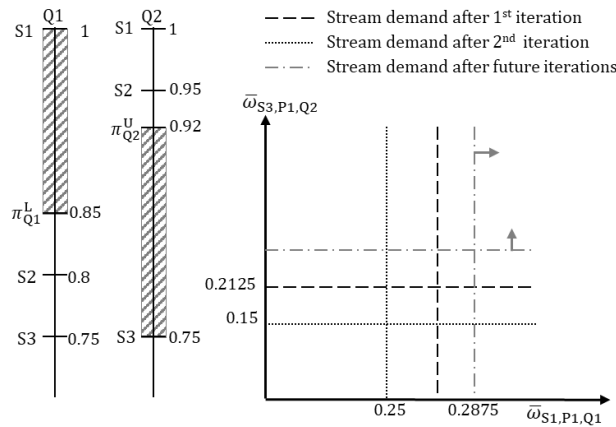


Figure 2-5. Illustrative example for demand updating via algebraic equations; pattern filled bars indicate feasible property domains.

2.4.2.3. Demand updating via solving LP

In Figure 2-6, we show an example with three streams (S1 to S3), two properties (Q1 and Q2), and one product (P1). Stream S1 is the only “good” stream for Q1 and its demand is again obtained from Eqn. (2-21) by considering the binary blend of S1 and S2: $\bar{\omega}_{S1,P1,Q1} =$

0.25. From Q2, we cannot propose nonzero demand for a single stream, since both S1 and S3 satisfy the specification.

However, based on specification for Q2 we can update demand for S1. Note that the binary blend with 0.25 units of S1 and 0.75 units of S2, which we used to obtain the demand for S1, violates the specification for Q2. Thus, we either increase the fraction of S1 in the binary blend, or introduce S3 into the blend, which is a lower quality stream for Q1 compared to S2. In both cases, we will end up with more S1.

Unlike the previous case, instead of updating the demand for S1 through an algebraic equation, we solve the following LP:

$$\begin{aligned}
 \min \quad & \hat{\omega}_{S1,P1} \\
 \text{s. t} \quad & \hat{\omega}_{S1,P1} + \hat{\omega}_{S2,P1} + \hat{\omega}_{S3,P1} \geq 1 \\
 & \hat{\omega}_{S1,P1} + 0.8\hat{\omega}_{S2,P1} + 0.75\hat{\omega}_{S3,P1} \geq 0.85(\hat{\omega}_{S1,P1} + \hat{\omega}_{S2,P1} + \hat{\omega}_{S3,P1}) \\
 & 0.8\hat{\omega}_{S1,P1} + 0.7\hat{\omega}_{S2,P1} + \hat{\omega}_{S3,P1} \geq 0.75(\hat{\omega}_{S1,P1} + \hat{\omega}_{S2,P1} + \hat{\omega}_{S3,P1})
 \end{aligned} \tag{LP1}$$

which returns an optimal objective function value of 0.269.

In general, given a product k , we can obtain the demand for a stream i' by solving the following LP:

$$\begin{aligned}
 \min \quad & \hat{\omega}_{i',k} \\
 \text{s. t} \quad & \sum_i \hat{\omega}_{ik} \geq \omega_k \\
 & \pi_{kl}^L \sum_i \hat{\omega}_{ik} \leq \sum_i \pi_{ik} \hat{\omega}_{ik} \leq \pi_{kl}^U \sum_i \hat{\omega}_{ik}, \quad l
 \end{aligned} \tag{LP2}$$

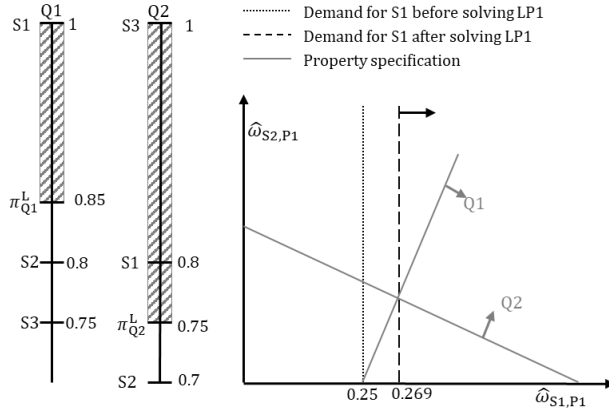


Figure 2-6. Illustrative example for demand updating via solving LP.

2.4.3 Complete algorithm

Figure 2-7 shows the flow chart of the complete algorithm for the calculation of demand for streams based on product demand and specifications. We introduce subsets $\mathbf{L}^L/\mathbf{L}^U$ for properties that have lower/upper bounding specification. The structure of the algorithm, assuming lower bounding specifications only, is as follows. For each product and specification, we check if there exists exactly one stream that satisfies the specification. If this is the case, we calculate its demand using Eqn. (2-21). After checking all specifications, we evaluate Eqn. (2-23) to see if a higher demand is obtained. If this is the case, we update demands by iteratively using Eqn. (2-23) until no improvements can be achieved; otherwise, we proceed to the next step. We then estimate the value of property l of the blend (denoted by θ_{kl}), by considering all nonzero $\hat{\omega}_{ik}$ and the stream that violates π_{kl}^L by the least margin. If $\pi_{kl}^L \leq \theta_{kl} \forall l$, the algorithm terminates; otherwise, we solve LP2 to update all nonzero $\hat{\omega}_{ik}$, one at a time. In all instances we tested, the algorithm runs in less than 3 seconds, which is negligible compared to the solution time of the MINLP models for the same instance.

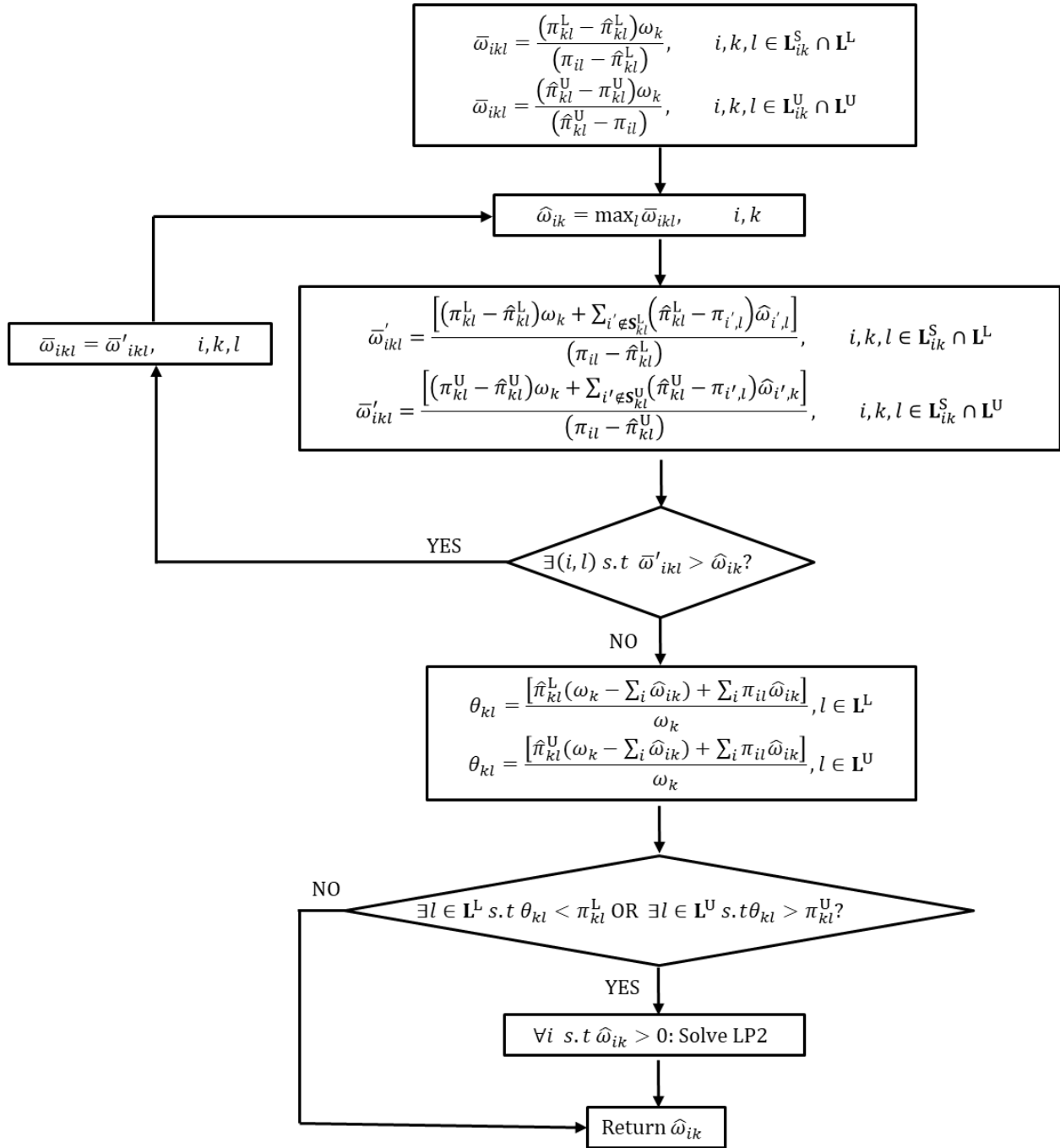


Figure 2-7. Flowchart of preprocessing algorithm for each product.

We illustrate the computing sequence of the preprocessing algorithm using the example shown in Figure 2-8 with four streams (S1 to S4), three properties (Q1 to Q3), and one product.

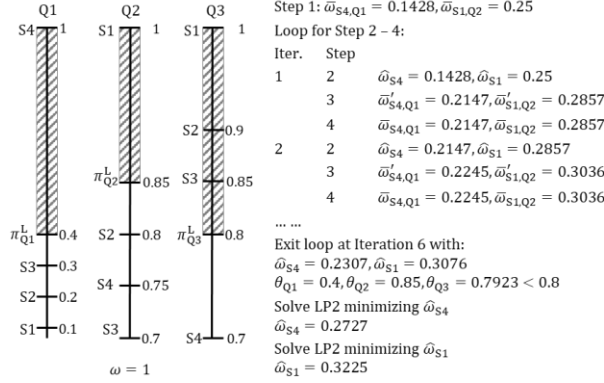


Figure 2-8. Illustrative example for the preprocessing algorithm with one product with $\omega = 1$ (index k is dropped for simplicity).

2.5 Product dedicated flow

We introduce a new nonnegative continuous variable $\hat{F}_{ik}$ to model the flow from a stream i dedicated to a product k . We first consider the overall flow balance:

$$\sum_i \sum_j \sum_t \tilde{F}_{ijt} = \sum_k \sum_j \sum_t \bar{F}_{jkt} + \sum_j I_j^F \quad (2-24)$$

where I_j^F denotes the inventory in blender j at the end of the scheduling horizon.

One observation is that the RHS contains one term for production and another term for final inventory. We partition the LHS into two parts similar to the RHS: flows that are dedicated to certain products and flows that will remain in the blenders. After the partition, we write valid constraints for the flows that are dedicated to products using demand for streams we obtained from the preprocessing algorithm.

Let F_i^R denote the final inventory of stream i in blenders. We have:

$$\sum_j \sum_t \tilde{F}_{ijt} = \sum_k \hat{F}_{ik} + F_i^R, \quad i \quad (2-25)$$

We match $\hat{F}_{ik}$ with the production of product k .

$$\sum_i \hat{F}_{ik} = \sum_j \sum_t \bar{F}_{jkt}, \quad k \quad (2-26)$$

Product dedicated flow variables introduced in Eqns. (2-25) - (2-26) can be easily defined for different blend scheduling models.

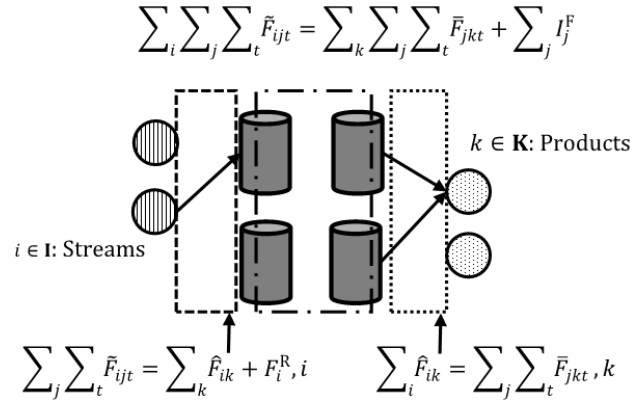


Figure 2-9. Illustrative graph for product dedicated flows.

2.6 Valid constraints

We present three types of valid constraints based on demand for streams and product dedicated flow variables.

2.6.1 Valid constraints with flow variables only

We first enforce demand satisfaction for each stream:

$$\hat{F}_{ik} \geq \hat{\omega}_{ik}, \quad i, k \quad (2-27)$$

More generally, for each specification for a product, the combined demand for “good” streams should be nonzero. Different from the demand for each “good” stream, we enforce:

$$\sum_{i \in \mathbf{S}_{kl}^U} (\hat{\pi}_{kl}^U - \pi_{il}) \hat{F}_{ik} \geq (\hat{\pi}_{kl}^U - \pi_{kl}^U) \omega_k + \sum_{i \notin \mathbf{S}_{kl}^U} (\pi_{il} - \hat{\pi}_{kl}^U) \hat{\omega}_{ik}, \quad k, l \in \mathbf{L}_k^M \quad (2-28)$$

$$\sum_{i \in \mathbf{S}_{kl}^L} (\pi_{il} - \hat{\pi}_{kl}^L) \hat{F}_{ik} \geq (\pi_{kl}^L - \hat{\pi}_{kl}^L) \omega_k + \sum_{i \notin \mathbf{S}_{kl}^L} (\hat{\pi}_{kl}^L - \pi_{il}) \hat{\omega}_{ik}, \quad k, l \in \mathbf{L}_k^M \quad (2-29)$$

Eqns. (2-28) – (2-29) are written for every product, and every property with specification satisfied by at least two streams (denoted by $\mathbf{L}_k^M$). We enforce the combined demand

satisfaction for the “good” streams for such specification by considering the blend that may contain: (1) every “good” stream for that specification, (2) streams violate that specification with $\hat{\omega}_{ik} > 0$, and (3) the stream violates that specification by the least margin.

Eqns. (2-27) - (2-29) employ parameter $\hat{\omega}_{ik}$ obtained from the preprocessing algorithm. If $\hat{\omega}_{ik} = 0$, Eqn. (27) will be trivially satisfied, while Eqns. (2-28) – (2-29) may still lead to a tighter relaxation.

When there are multiple due times for orders and backlogging is not allowed, we introduce the parameter $\hat{\omega}_{ikt}$, which is time indexed, to denote the cumulative demand for streams until time point t . The preprocessing algorithm calculates the cumulative demand at different time points accordingly. Variable $\hat{F}_{ikt}$ includes a time index t to denote the cumulative product dedicated flow until time point t , and Eqns. (2-27) - (2-29) are written at every time point when an order is due.

2.6.2 Valid constraints with binary variables

We can also express tightening constraints using binary variables by recognizing that for each product, “good” streams are required:

$$\bar{X}_{jkt} \leq \sum_{i \in \mathbf{S}_{kl}^U} \sum_{j'} \sum_{t' \leq t} \tilde{X}_{i,j',t'}, \quad j, k, l, t \quad (2-30)$$

$$\bar{X}_{jkt} \leq \sum_{i \in \mathbf{S}_{kl}^L} \sum_{j'} \sum_{t' \leq t} \tilde{X}_{i,j',t'}, \quad j, k, l, t \quad (2-31)$$

We can also incorporate demand for streams and binary variables $\tilde{X}_{s,j,t}$. We have:

$$\sum_{i \in \mathbf{S}_{kl}^U \setminus \{i'\}} (\hat{\pi}_{kl}^U - \pi_{il}) \hat{F}_{ik} \geq \left(1 - \sum_j \sum_t \tilde{X}_{i',j,t}\right) \left[\hat{\pi}_{kl}^U - \pi_{kl}^U \right] \omega_k + \sum_{i \notin \mathbf{S}_{kl}^U} (\pi_{il} - \hat{\pi}_{kl}^U) \hat{\omega}_{ik}, \quad k, l \in \mathbf{L}_k^M, i' \in \mathbf{S}_{kl}^U \quad (2-32)$$

$$\sum_{i \in \mathbf{S}_{kl}^L \setminus \{i'\}} (\pi_{il} - \hat{\pi}_{kl}^L) \hat{F}_{ik} \geq \quad (2-33)$$

$$\left(1 - \sum_j \sum_t \tilde{X}_{i',j,t}\right) \left[(\pi_{kl}^L - \hat{\pi}_{kl}^L) \omega_k + \sum_{i \notin \mathbf{S}_{kl}^L} (\hat{\pi}_{kl}^L - \pi_{il}) \hat{\omega}_{ik} \right], \quad k, l \in \mathbf{L}_k^M, i' \in \mathbf{S}_{kl}^L$$

Eqns. (2-32) – (2-33) enforce lower bound for a subset of “good” streams when one “good” stream, denoted as i' , has zero cumulative flow (since $\sum_j \sum_t \tilde{X}_{i',j,t} = 0$).

Similar to Eqns. (2-28) – (2-29), when there are multiple due times for orders and backlogging is not allowed, Eqns. (2-32) - (2-33) can be written at every time point when an order is due, with the previously mentioned modifications. We also note that Eqns. (2-30) - (2-33) are inspired by the facet-defining inequalities proposed by Papageorgiou et al. (Papageorgiou et al. 2012) for the fixed-charge transportation problem with product blending, in which they are called “lifted blending facets”.

2.6.3 Specifications for product dedicated flows

Finally, we write the following constraints that enforce the specifications based on product dedicated flow variables:

$$\pi_{kl}^L \sum_j \sum_t \bar{F}_{jkt} \leq \sum_i \pi_{il} \hat{F}_{ik} \leq \pi_{kl}^U \sum_j \sum_t \bar{F}_{jkt}, \quad k, l \quad (2-34)$$

Note that unlike Eqns. (2-27) - (2-29), Eqn. (2-34) is written for the entire scheduling horizon.

If stream i has initial inventory in the blender j , then we consider it as a separate stream in the preprocessing algorithm. The corresponding $\tilde{X}_{ijt}$ will be fixed to 1 (Eqn. (2-5) will not be enforced for such (j, t) combination), and $\tilde{F}_{ijt}$ will be fixed to the initial inventory.

In Table 2-3, we list the valid inequalities for the example shown in Figure 2-8.

Table 2-3. Valid constraints for the example shown in Figure 2-8 (index k is dropped)

Constraints	Description
$\sum_t \hat{F}_{S1,t} \geq 0.3225$	Eqn. (2-27)
$0.2 \sum_t \hat{F}_{S1,t} + 0.1 \sum_t \hat{F}_{S2,t} + 0.05 \sum_t \hat{F}_{S3,t} \geq 0.1 \times 1 + 0.1 \times 0.2727$	Eqn. (2-29)
$\bar{X}_{j,t} \leq \sum_{j'} \sum_{t' \leq t} \tilde{X}_{S1,j',t'}, \quad j, t$	Eqn. (2-31)
$\bar{X}_{j,t} \leq \sum_{j'} \sum_{t' \leq t} \tilde{X}_{S4,j',t'}, \quad j, t$	
$\bar{X}_{j,t} \leq \sum_{j'} \sum_{t' \leq t} (\tilde{X}_{S1,j',t'} + \tilde{X}_{S2,j',t'} + \tilde{X}_{S3,j',t'}), \quad j, t$	
$0.2 \sum_t \hat{F}_{S1,t} + 0.1 \sum_t \hat{F}_{S2,t} \geq (1 - \sum_j \sum_t \tilde{X}_{S3,j,t})(0.1 \times 1 + 0.1 \times 0.2727)$	Eqn. (2-33)
$0.1 \sum_t \hat{F}_{S2,t} + 0.05 \sum_t \hat{F}_{S3,t} \geq (1 - \sum_j \sum_t \tilde{X}_{S1,j,t})(0.1 \times 1 + 0.1 \times 0.2727)$	
$0.2 \sum_t \hat{F}_{S1,t} + 0.05 \sum_t \hat{F}_{S3,t} \geq (1 - \sum_j \sum_t \tilde{X}_{S2,j,t})(0.1 \times 1 + 0.1 \times 0.2727)$	
$0.1 \sum_t \hat{F}_{S1,t} + 0.2 \sum_t \hat{F}_{S2,t} + 0.3 \sum_t \hat{F}_{S3,t} + \sum_t \hat{F}_{S4,t} \geq 0.4 \sum_j \sum_t \bar{F}_{j,t}$	Eqn. (2-34)
$\sum_t \hat{F}_{S1,t} + 0.8 \sum_t \hat{F}_{S2,t} + 0.7 \sum_t \hat{F}_{S3,t} + 0.75 \sum_t \hat{F}_{S4,t} \geq 0.85 \sum_j \sum_t \bar{F}_{j,t}$	
$\sum_t \hat{F}_{S1,t} + 0.9 \sum_t \hat{F}_{S2,t} + 0.85 \sum_t \hat{F}_{S3,t} + 0.7 \sum_t \hat{F}_{S4,t} \geq 0.8 \sum_j \sum_t \bar{F}_{j,t}$	

2.7 Computational results

We test our methods using 20 instances. Computational experiments are conducted on a cluster running CentOS Linux 7 with Intel Xeon (E5520) processors at 2.27 GHz and 16 GB of RAM. The instances are coded in GAMS 24.7. We use 2 different MINLP solvers: BARON 16.3.4 and SCIP 3.2, and CPLEX 12.6 is used for solving the MILP models. Default options are used for all solvers.

2.7.1 Problem instances

Stream properties, product specifications, and blender network configurations are taken from published literature. Table 2-4 summarizes some key characteristics of the instances

Table 2-4. Size of tested instances

Instance	S	J	P	Q	T	Property source
1-5	7	2	3	7	6	Castillo and Mahalec (2014a)
6-12	9	3	4	9	6	Reddy, Karimi and Srinivasan (2004)
13-15	8	4	4	1	6	Castro and Grossmann (2014)
16-20	2	8	2	1	6	Lotero et al. (2016)

2.7.2 Case study

We consider an instance (Instance 17) with two streams, eight blenders, two products, one property and six periods, with the corresponding parameters given in Table 2-5, and the network configuration shown in Figure 2-10(A).

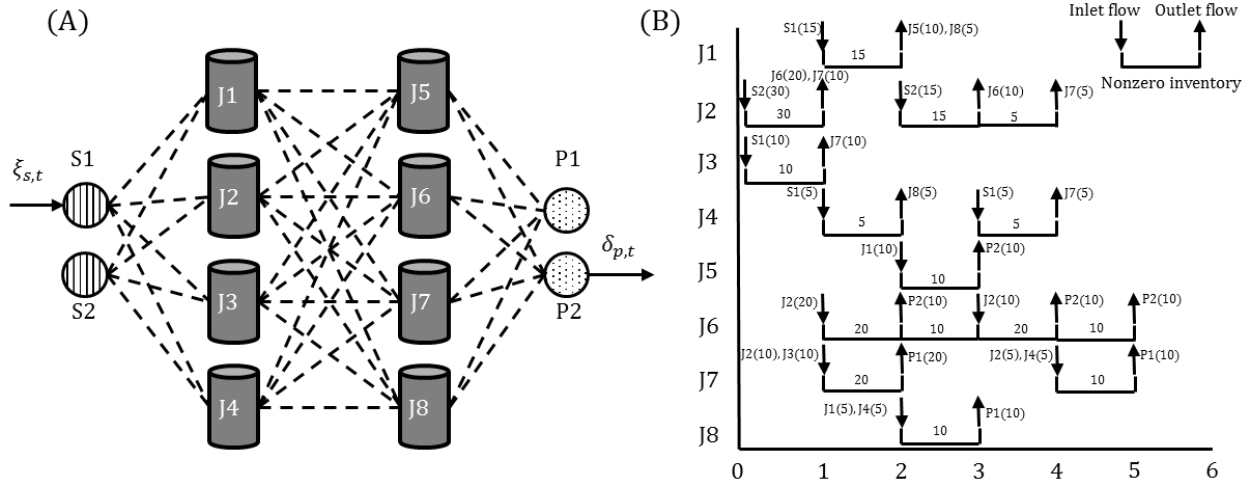


Figure 2-10. (A). Network configuration for the case study (dashed lines indicate connectivity between streams, blenders, and products). (B). Gantt chart for an optimal solution.

Table 2-5. Parameters for streams and products for the case study

	$\xi_{i,0}$	$\xi_{i,1}$	$\xi_{i,2}$	$\delta_{k,3}$	$\delta_{k,4}$	$\delta_{k,5}$	$\delta_{k,6}$	$\pi_{i,Q1}$	$\pi_{i,Q1}^U$	$\tilde{\alpha}_{i,j}^V$
S1	10	10	10	-	-	-	-	0.06	-	1
S2	30	30	30	-	-	-	-	0.26	-	2
P1	-	-	-	10	10	10	10	-	0.16	-
P2	-	-	-	10	10	10	10	-	1	-

Since S1 is the only stream that satisfies the specification for P1, demand for S1 is nonzero.

The preprocessing algorithm calculates the demand for S1 for $t = \{3,4,5,6\}$ as 5, 10, 15, and 20, respectively. Eqn. (2-30) yields four constraints, Eqn. (2-32) leads to 24 constraints, and Eqn. (2-33) leads to four constraints. Model statistics for the case study are given in Table 2-6. An optimal solution is shown in Figure 2-10(B) with an objective function value of 100.

Table 2-6. Model statistics for the case study

	M^C	M^{SB}
Linear constraints	5580	7562
Continuous variables	1118	2490
Nonlinear constraints	48	72
Discrete variables	192	192

The proposed tightening constraints are tested using different models as shown in Table 2-7.

The CPU time for the concentration-model and source-based models can be found in Table 2-8 and Table 2-9, respectively. For the concentration-based model, with the addition of the proposed constraints, the solvers find feasible solutions and solve the case study in less than 3 minutes for most models and combinations of added constraints. The addition of the tightening constraints enhances the solution of the source-based model as well.

We further test our methods on MILP models L_1M and L_2M . Computational results for selected models are given in Table 2-10. Notably, the proposed methods bring improvement to MILP models, with the addition of some constraints leading to one order of magnitude improvement in CPU time.

Table 2-7. Model description

Models	Description
M^x	Original MINLP, $x = \{C, SB\}$
M_I^x	M^x + Eqns. (2-27) - (2-29)
M_{II}^x	M^x + Eqns. (2-30) - (2-33)
M_P^x	M^x + Eqns. (2-14) - (2-15), (2-34)
$M_{I_{II}}^x$	M_I^x + Eqns. (2-30) - (2-33)
$M_{I_P}^x$	M_I^x + Eqns. (2-14) - (2-15), (2-34)
$M_{II_P}^x$	M_{II}^x + Eqns. (2-14) - (2-15), (2-34)
$M_{I_{II_P}}^x$	$M_{I_{II}}^x$ + Eqns. (2-14) - (2-15), (2-34)

Table 2-8. CPU time in seconds for the case study for concentration-based model

	M^C	M_I^C	M_{II}^C	M_P^C	$M_{I_{II}}^C$	$M_{I_P}^C$	$M_{II_P}^C$	$M_{I_{II_P}}^C$
BARON	-	39.91	4555.89	108.44	135.07	101.35	86.35	137.15
SCIP	-	(20%)*	(20%)*	30.93	(20%)*	30.86	193.99	194.03

Note: “-” indicates no solution found after 2 hours. “*” indicates instance not solved to global optimality, with optimality gap after 2 hours shown in brackets.

Table 2-9. CPU time in seconds for the case study for source-based model

	M^{SB}	M_I^{SB}	M_{II}^{SB}	M_P^{SB}	$M_{I_{II}}^{SB}$	$M_{I_P}^{SB}$	$M_{II_P}^{SB}$	$M_{I_{II_P}}^{SB}$
BARON	572.23	482.18	674.25	534.78	583.89	249.96	627.02	382.45
Ratio	(1)	(0.84)	(1.18)	(0.93)	(1.02)	(0.44)	(1.09)	(0.67)
SCIP	23.63	6.32	6.66	30.07	6.34	30.31	8.51	5.49
Ratio	(1)	(0.27)	(0.28)	(1.27)	(0.27)	(1.28)	(0.36)	(0.23)

Note: Numbers in the brackets are the ratio of CPU time of the corresponding model over CPU time of M^{SB} .

Table 2-10. CPU time for the case study with different linear models

x	xM^C	xM_P^C	$xM_{II_P}^C$	$xM_{I_{II_P}}^C$	xM^{SB}	xM_P^{SB}	$xM_{II_P}^{SB}$	$xM_{I_{II_P}}^{SB}$
L1	>7200(20%)	6623.49	6584.02	1958.10	578.72	226.20	230.96	569.68
L2	246.05	37.80	38.05	10.76	54.14	29.33	29.93	303.46

2.7.3 Results for MINLP models

Computational results for all 20 instances, using BARON, are shown in Figure 2-11 using performance profiles. The profiles for M_P , M_{II_P} and $M_{I_{II_P}}$, three models that have the best performance overall, are shown along with the original formulation. For the concentration-

based model, adding constraints on continuous variables brings significant improvement, while for the source-based model, adding constraints associated with binary variables improves the performance the most.

One observation is that adding the proposed constraints brings more significant improvements to M^C compared to M^{SB} . One explanation is that M^{SB} is a tighter formulation compared to M^C ; thus the benefits of adding the proposed constraints appears to be limited. Table 2-11 gives the percentage of instances solved to global optimality in 2 hours.

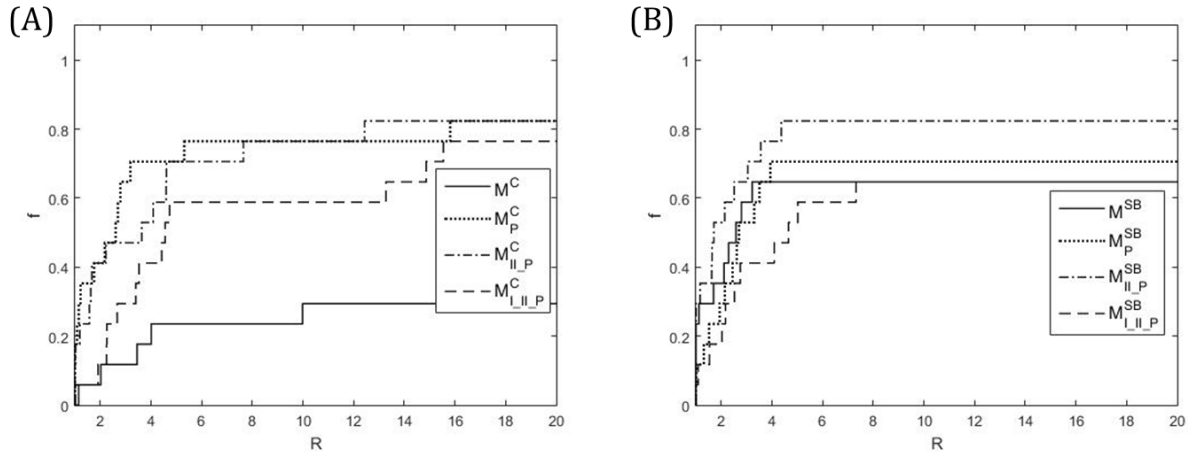


Figure 2-11. Performance profiles for concentration-based model (A) and source-based model (B).

Table 2-11. Percentages of instances solved to global optimality in 2 hours

x	M^x	M_I^x	M_{II}^x	M_P^x	$M_{I,II}^x$	$M_{I,P}^x$	$M_{II,P}^x$	$M_{I,II,P}^x$
C	29.4%	29.4%	17.6%	82.3%	35.3%	82.3%	82.3%	76.5%
SB	64.7%	64.7%	58.8%	70.6%	76.5%	64.7%	88.2%	70.6%

2.7.4 Results for MILP models

The performance profiles for L_2M for the same 20 instances are presented in Figure 2-12. As in the case of the MINLP models, the addition of constraints on continuous variables brings significant improvement to the solution of L_2M^C , while the addition of the constraints

expressed using binary variables enhance the solution of L_2M^{SB} . It is worth noting that many studies aim to solve the MINLP models for multiperiod blending problem by solving MILPs. Our method is applicable to those MILPs as well.

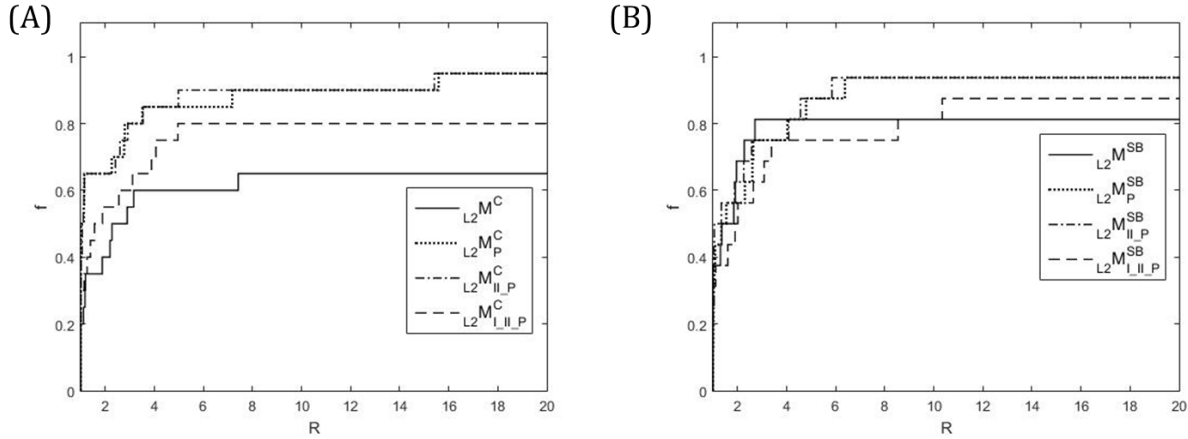


Figure 2-12. Performance profiles for linear relaxation of the concentration-based model (A) and source-based model (B).

2.8 Conclusion

We developed solution methods for multiperiod blending problem focusing on cost minimization problems. We first developed a preprocessing algorithm to calculate lower bounds on stream flows. The bounds obtained from this algorithm, along with the newly introduced product dedicated flow variables, are then used to generate tightening constraints. The proposed methods lead to significant improvement in the solution time of MINLP models for multiperiod blending problem as well as models based on linear approximations of these model.

Chapter 3

Variable bound tightening and valid constraints for multiperiod blending

In this chapter we focus on variable bound tightening methods as well as valid constraints derived from the tightened bounds for multiperiod blending problem. We assume no flows between blenders in this chapter, and without loss of generality, we only consider upper bounding specifications. To simplify notation, we reintroduce the following variables in the source-based model, with new symbols:

F_{ijt} : Flow of stream i to blender j at time point t

I_{ijt} : Inventory of stream i in blender j during time period t

R_{jkt} : Split fraction for inventory in blender j to product k at time point t

$\hat{F}_{ijkt}$: Flow of stream i from blender j to product k at time point t

We also define the following binary variable:

X_{jkt} : = 1 when blender j feeds product k at time point t

We focus on the following constraints in the source-based model:

$$\hat{F}_{ijkt} = I_{ijt}R_{jkt}, \quad i, j, k, t \quad (3-1)$$

$$\sum_i \pi_{il} I_{ijt} \leq \pi_{kl}^U \sum_i I_{ijt} + \gamma_j \pi_{kl}^U (1 - X_{jkt}), \quad j, k, l, t \quad (3-2)$$

We introduce a reformulation of the source-based model using lifting, and a preprocessing method to calculate tight bounds.

3.1 Reformulation of bilinear terms

We lift I_{ijt} , and partition it into nonnegative continuous variables U_{ijkt} and V_{ijkt} :

$$I_{ijt} = U_{ijkt} + V_{ijkt}, \quad i, j, k, t \quad (3-3)$$

$$\sum_i U_{ijkt} \leq \gamma_j(1 - X_{jkt}), \quad j, k, t \quad (3-4)$$

$$\sum_i V_{ijkt} \leq \gamma_j X_{jkt}, \quad j, k, t \quad (3-5)$$

where U_{ijkt} represents the inventory of stream i in blender j during time period t when there is no flow from blender j to product k ($X_{jkt} = 0$), and V_{ijkt} represents such inventory when $X_{jkt} = 1$.

Eqn. (3-1) now becomes:

$$\hat{F}_{ijkt} = V_{ijkt} R_{jkt}, \quad i, j, k, t \quad (3-6)$$

and Eqn. (3-2) can be re-written as:

$$\sum_i \pi_{il} V_{ijkt} \leq \pi_{kl}^U \sum_i V_{ijkt}, \quad j, k, l, t \quad (3-7)$$

The reformulated model, with variables U_{ijkt} and V_{ijkt} , henceforth referred to as M^{UV} . In M^{UV} , the variables involved in a bilinear term are V_{ijkt} and R_{jkt} . We aim to tighten bounds on V_{ijkt} .

3.2 Preprocessing method for variable bounds tightening

A relaxation of Eqn. (3-5) is:

$$\sum_i V_{ijkt} \leq \gamma_j, \quad j, k, t \quad (3-8)$$

The right hand side (RHS) parameter γ_j can be tightened. We first rewrite Eqn. (3-7) as:

$$\sum_i (\pi_{il} - \pi_{kl}^U) V_{ijkt} \leq 0, \quad j, k, l, t \quad (3-9)$$

We define a parameter μ_{ikl} to represent the margin by which stream i violates the specification for property l for product k : $\mu_{ikl} = \pi_{il} - \pi_{kl}^U$ (note that μ_{ikl} can be positive or negative). Eqn. (3-9) can thus be written as:

$$\sum_i \mu_{ikl} V_{ijkt} \leq 0, \quad j, k, l, t \quad (3-10)$$

We aim to calculate a tighter upper bound on V_{ijkt} using Eqn. (3-8) and (3-10). For simplicity, we drop indices j, k , and t for now, thus μ_{ikl} becomes $\mu_{il} = \pi_{il} - \pi_l^U$. We consider the following:

$$\sum_i V_i \leq \gamma \quad (3-11)$$

$$\sum_i \mu_{il} V_i \leq 0, \quad l \quad (3-12)$$

We define a parameter $\mu_l^* = \min_i \{\mu_{il}\}$ and a set function $b(l) = \arg \min_i \{\mu_{il}\}$ that returns the “best” stream for property l . It is possible that, for a property l , there are multiple streams with $\mu_{il} = \mu_l^*$ (i.e., multiple “best” streams). In that case, we consider $b(l)$ being the stream with the smallest index among all such streams. We assume $\mu_l^* < 0$ because (1) if $\mu_l^* > 0$ then $\mu_{il} > 0, \forall i$ and since $V_i \geq 0$, Eqn. (3-12) can be satisfied only if $V_i = 0, \forall i$; and (2) if $\mu_l^* = 0$, then Eqn. (3-12) can be satisfied only if $V_i = 0, \forall i: \mu_{il} \neq 0$.

We define subset $\mathbf{L}_i = \{l: \mu_{il} > 0\}$, that is, the set of properties with specification violated by stream i . Similarly, we define subset $\mathbf{I}_l = \{i: \mu_{il} > 0\}$, that contains streams that violate the specification for property l .

To illustrate, we consider an illustrative example with $\mathbf{I} = \{1,2,3\}$, $\mathbf{L} = \{L1, L2\}$. Parameters π_{il}, π_l^U and μ_{il} calculated from them are given in Figure 3-1.

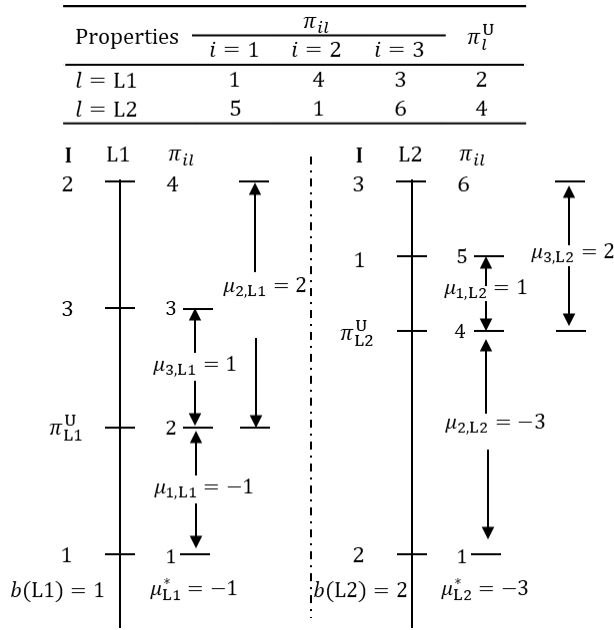


Figure 3-1. An illustrative example for parameters

3.2.1 Bounds tightening using a pair of constraints

From Eqn. (3-11) it is clear that γ is a valid upper bound on V_i . To tighten such upper bound, we combine Eqn. (3-11) with one constraint in Eqn.(3-12). For V_i with positive coefficient in at least one constraint in Eqn. (3-12) (i.e., streams that violates at least one specification), bounds derived from such pairs of constraints will be tighter than γ .

To calculate bounds using aforementioned pairs of constraints, we first multiply all inequalities in Eqn. (3-12) by $-\frac{1}{\mu_l^*}$ (recall that $\mu_l^* < 0$) to obtain:

$$\sum_i (-\frac{\mu_{il}}{\mu_l^*})V_i \leq 0, \quad l$$

Next, we combine Eqn. (3-11), with a weight equal to 1, with each individual constraint above,

$$\sum_i (1 - \frac{\mu_{il}}{\mu_l^*})V_i \leq \gamma, \quad l \quad (3-13)$$

Each constraint in Eqn. (3-13) is obtained by combing a pair of constraints: Eqn. (3-11) and one constraint in Eqn. (3-12). Next, we derive bounds on V_i from Eqn. (3-13).

After using i' instead of i , we obtain:

$$\sum_{i'} (1 - \frac{\mu_{i'l}}{\mu_l^*})V_{i'} \leq \gamma, \quad l$$

For each $l \in \mathbf{L}$, we consider streams in the set $\mathbf{I}_l$, and isolate such streams, one at a time, from the summation on the left hand side (LHS):

$$(1 - \frac{\mu_{il}}{\mu_l^*})V_i + \sum_{i' \neq i} (1 - \frac{\mu_{i'l}}{\mu_l^*})V_{i'} \leq \gamma, \quad l, i \in \mathbf{I}_l$$

We examine the second term on the LHS of the above equation. By the definition of μ_l^* we have $\mu_{i'l} \geq \mu_l^*$. Thus, if $\mu_{i'l} < 0$ then $\frac{\mu_{i'l}}{\mu_l^*} \in [0,1]$ and therefore $1 - \frac{\mu_{i'l}}{\mu_l^*} \geq 0$; and if $\mu_{i'l} \geq 0$, then $-\frac{\mu_{i'l}}{\mu_l^*} \geq 0$ and therefore $1 - \frac{\mu_{i'l}}{\mu_l^*} > 1 > 0$. Given that $V_{i'}$ is nonnegative, we have $\sum_{i' \neq i} (1 - \frac{\mu_{i'l}}{\mu_l^*})V_{i'} \geq 0$. Thus, the following inequality, obtained by dropping the summation on the LHS of the above equation, is valid:

$$(1 - \frac{\mu_{il}}{\mu_l^*})V_i \leq \gamma, \quad i, l \in \mathbf{L}_i \quad (3-14)$$

and since $1 - \frac{\mu_{il}}{\mu_l^*} > 0 \forall i, l \in \mathbf{L}_i$ we have:

$$V_i \leq \gamma / (1 - \frac{\mu_{il}}{\mu_l^*}), \quad i, l \in \mathbf{L}_i$$

Or

$$V_i \leq \hat{\gamma}_{il} = -\frac{\mu_l^* \gamma}{\mu_{il} - \mu_l^*}, \quad i, l \in \mathbf{L}_i \quad (3-15)$$

Note that $\hat{\gamma}_{il}$ is smaller than γ and serves as an upper bound on V_i derived from property l .

The physical interpretation of $\hat{\gamma}_{il}$ is as follows. Suppose we have to meet demand for volume γ for a product. Parameter $\hat{\gamma}_{il}$ represents the maximum volume of stream i that can be used towards volume γ based on property $l \in \mathbf{L}_i$. In other words, $\hat{\gamma}_{il}/\gamma$ is the maximum fraction of stream i that can be used for such product. This stream-specific volume, $\hat{\gamma}_{il}$, is derived by considering the binary mixture of streams i and $b(l)$ that satisfies the specification for property l exactly.

Once we calculated $\hat{\gamma}_{il}$ from Eqn. (3-15), the upper bound on V_i , denoted as $\bar{\gamma}_i$, is set to the smallest $\hat{\gamma}_{il}$, considering all properties that stream i violates (i.e., all $l \in \mathbf{L}_i$), $\bar{\gamma}_i = \min_{l \in \mathbf{L}_i} \{\hat{\gamma}_{il}\}$. For illustration purpose, we introduce a set function $m(i)$ that returns the property l from which $\bar{\gamma}_i$ is derived (i.e., $m(i) = \arg \min_{l \in \mathbf{L}_i} \{\hat{\gamma}_{il}\}$).

Consider the illustrative example shown in Figure 3-1 with $\gamma = 1$. Based on the calculated parameter μ_{il} shown in Figure 3-1, we have the following constraints for Eqn. (3-11) – (3-12):

$$V_1 + V_2 + V_3 \leq 1$$

$$-V_1 + 2V_2 + V_3 \leq 0$$

$$V_1 - 3V_2 + 2V_3 \leq 0$$

The calculations described above lead to bounds on V_i given in Table 3-1.

Table 3-1. Bounds calculated by aggregating pair of constraints

	$i = 1$	$i = 2$	$i = 3$
$\hat{\gamma}_{i,L1}$	—	1/3	1/2
$\hat{\gamma}_{i,L2}$	3/4	—	3/5
$\bar{\gamma}_i$	3/4	1/3	1/2

Note: “—” indicates the corresponding $\hat{\gamma}_{il}$ is not calculated since $L1 \notin \mathbf{L}_1$ and $L2 \notin \mathbf{L}_2$.

3.2.2 Bounds updating

In this subsection, we discuss how we can further tighten $\hat{\gamma}_{il}$. Recall that bounds on V_i are derived using pairs of constraints. For each such pair, we can derive bounds tighter than $\hat{\gamma}_{il}$ by considering one additional constraint in Eqn. (3-12) that is not included in such pair.

We elaborate the aforementioned idea in the context of blending. Recall that $\hat{\gamma}_{il}$ is based on the binary mixture of streams i and $b(l)$ with volume γ , which satisfies the specification for property l and contains $(\gamma - \hat{\gamma}_{il})$ volume of stream $b(l)$. It is possible that stream $b(l)$ violates specifications for other properties, and its maximum volume in γ volume of product is less than $(\gamma - \hat{\gamma}_{il})$. For all $(i, l \in \mathbf{L}_i, b(l))$ combinations, we check if the following holds:

$$\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il}.$$

If $\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il}$, then there exists a property $m[b(l)]$ (the property from which $\bar{\gamma}_{b(l)}$ is derived) whose specification is violated by the binary mixture of stream i and $b(l)$ that satisfies specification for property l exactly (i.e., $\mu_{i,m[b(l)]}\hat{\gamma}_{il} + \mu_{b(l),m[b(l)]}(\gamma - \hat{\gamma}_{il}) > 0$). Note

that property $m[b(l)]$ is not considered when deriving $\hat{\gamma}_{il}$; when taking it into account, the binary mixture of stream i and $b(l)$ will not be able to satisfy the specifications for property l and property $m[b(l)]$ simultaneously. In such case, we include one additional stream to the binary mixture. Note that by including one additional stream, $\hat{\gamma}_{il}$ will be tightened since it is previously obtained from the binary mixture of streams i and $b(l)$ that satisfies specification for property l exactly.

Specifically, we update $\hat{\gamma}_{il}$ by considering the “second best” stream for property l . We define $\mu_l^+ = \min_{i' \neq b(l)} \{\mu_{i'l}\}$. Let $b^+(l)$ be a set function that returns the “second best” stream: $b^+(l) = \arg \min_{i' \neq b(l)} \{\mu_{i'l}\}$, which implies $\mu_l^+ = \pi_{b^+(l),l} - \pi_{kl}^U$. If there are multiple “second best” streams, we proceed as follows: for the specific $(i, l \in \mathbf{L}_i, b(l))$ combination being considered, if $\mu_{il} = \mu_l^+$ (i.e., stream i is one of the “second best” streams), then $b^+(l) = i$; else, $b^+(l)$ is the stream with the smallest index among all such streams.

To tighten $\hat{\gamma}_{il}$, we prove three propositions. In Proposition 1, we consider a special case where $b^+(l) = i$, while in Propositions 2 and Propositions 3 we consider the more general case where $b^+(l) \neq i$. For Propositions 2 and Propositions 3, we consider a mixture with volume γ that contains stream i , $b(l)$, and $b^+(l)$, and satisfies the specification for property l . Note that for volume γ of such mixture, the (current) upper bound on volume of stream $b(l)$ is $\bar{\gamma}_{b(l)}$. Assume we have volume $\hat{\gamma}_{il}$ for stream i and volume $(\gamma - \bar{\gamma}_{b(l)} - \hat{\gamma}_{il})$ for stream $b^+(l)$. Then, for property l we have:

$$\mu_l^* \bar{\gamma}_{b(l)} + \mu_{il} \hat{\gamma}_{il} + \mu_l^+ (\gamma - \bar{\gamma}_{b(l)} - \hat{\gamma}_{il}) \leq 0, \quad i, l \in \mathbf{L}_i$$

which is equivalent to:

$$(\mu_{il} - \mu_l^+) \hat{\gamma}_{il} \leq \mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}, \quad i, l \in \mathbf{L}_i$$

For Propositions 2 and Propositions 3, since $b^+(l) \neq i$, by the definition of μ_l^+ it follows that $\mu_{il} - \mu_l^+ > 0$. Thus, we have:

$$\hat{\gamma}_{il} \leq \frac{\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)}, \quad i, l \in \mathbf{L}_i$$

Note that the RHS of the above equation can be nonpositive. Proposition 2 shows that in such case zero is a valid upper bound on V_i . If the RHS is positive, Proposition 3 shows that it is a valid upper bound on V_i .

Proposition 1 For $(i, l \in \mathbf{L}_i, b(l))$ with $\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il}$ and $b^+(l) = i$, if $\sum_{i' \in \mathbf{I}} \mu_{i'l} V_{i'} \leq 0$, then $\hat{\gamma}_{il} \leq 0$.

Proof (by contradiction).

Since $b^+(l) = i$ and $l \in \mathbf{L}_i$, it follows that the “second best” stream violates the specification for property l , thus $\mu_l^+ = \mu_{il} > 0$.

From Eqn. (3-15) we have $\hat{\gamma}_{il} = -\frac{\mu_l^* \gamma}{\mu_l^+ - \mu_l^*}$, which leads to $(\mu_l^+ - \mu_l^*) \hat{\gamma}_{il} = -\mu_l^* \gamma$. If we move all

terms to the LHS, we have $\mu_l^+ \hat{\gamma}_{il} - \mu_l^* \hat{\gamma}_{il} + \mu_l^* \gamma = 0$, and thus, $\mu_l^+ \hat{\gamma}_{il} + \mu_l^* (\gamma - \hat{\gamma}_{il}) = 0$

To simplify notation, we introduce $\varepsilon = \hat{\gamma}_{il}$, which means that above equation can be written as,

$$\mu_l^+ \varepsilon + \mu_l^* (\gamma - \varepsilon) = 0 \tag{3-16}$$

Next, to prove the result using contradiction, we assume that $\hat{\gamma}_{il} = \varepsilon > 0$.

Recall that $\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il} = \gamma - \varepsilon$, and thus if we multiply both sides of the inequality with $\mu_l^* < 0$ we obtain $\mu_l^* \bar{\gamma}_{b(l)} > \mu_l^* (\gamma - \varepsilon)$, and thus, from Eqn. (3-16), we have:

$$\mu_l^+ \varepsilon + \mu_l^* \bar{\gamma}_{b(l)} > 0 \quad (3-17)$$

We also have

$$\sum_{i' \in I} \mu_{i'l} V_{i'} = \mu_{il} V_i + \sum_{i' \neq i} \mu_{i'l} V_{i'} \quad (3-18)$$

If $V_i = \hat{\gamma}_{il} = \varepsilon$, then

$$\sum_{i' \in I} \mu_{i'l} V_{i'} = \mu_l^+ \varepsilon + \sum_{i' \neq i} \mu_{i'l} V_{i'} \quad (3-19)$$

Note that

$$\sum_{i' \neq i} \mu_{i'l} V_{i'} = \mu_l^* V_{b(l)} + \sum_{i' \notin \{b(l), i\}} \mu_{i'l} V_{i'} \quad (3-20)$$

with $\mu_l^* < 0$ and $V_{b(l)} \leq \bar{\gamma}_{b(l)}$.

Since the “second best” stream, in this case stream i , violates the specification for property l (i.e., $\mu_{il} > 0$), it follows that $\mu_{i'l} > 0, \forall i' \notin \{b(l), i\}$, while $\mu_l^* < 0$. Since $V_{i'}$ is nonnegative, the RHS of Eqn. (3-20) decreases as the value of $V_{b(l)}$ increases. With $V_{b(l)}$ upper bounded by $\bar{\gamma}_{b(l)}$, we have:

$$\sum_{i' \neq i} \mu_{i'l} V_{i'} \geq \mu_l^* \bar{\gamma}_{b(l)} + \sum_{i' \notin \{b(l), i\}} \mu_{i'l} V_{i'} \quad (3-21)$$

Combing Eqn. (3-19) and (3-21) we have:

$$\sum_{i' \in I} \mu_{i'l} V_{i'} = \mu_l^+ \varepsilon + \sum_{i' \neq i} \mu_{i'l} V_{i'} \geq \mu_l^+ \varepsilon + \mu_l^* \bar{\gamma}_{b(l)} + \sum_{i' \notin \{b(l), i\}} \mu_{i'l} V_{i'} \quad (3-22)$$

with $\sum_{i' \notin \{b(l), i\}} \mu_{i'l} V_{i'} \geq 0$ (since $\mu_{i'l} > 0, \forall i' \notin \{b(l), i\}$ and $V_{i'}$ is nonnegative) and $\mu_i^+ \varepsilon + \mu_i^* \bar{\gamma}_{b(l)} > 0$ (see Eqn. (3-17)).

Thus, from Eqn. (3-22) it follows that $\sum_{i' \in I} \mu_{i'l} V_{i'} > 0$, which leads to a contradiction. ■

Before presenting Proposition 2 and Proposition 3, we introduce some prerequisites. For $(i, l \in \mathbf{L}_i, b(l))$ with $\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il}$ and $b^+(l) \neq i$, to derive a valid upper bound on V_i , we again consider volume γ for a product (i.e., $\sum_{i' \in I} V_{i'} = \gamma$), where we assume $V_i = \hat{\gamma}_{il}$. Such assumptions imply (1) $\sum_{i' \neq i} V_{i'} = \gamma - \hat{\gamma}_{il}$ and (2)

$$\sum_{i' \in I} \mu_{i'l} V_{i'} = \mu_{il} \hat{\gamma}_{il} + \sum_{i' \neq i} \mu_{i'l} V_{i'} \quad (3-23)$$

The LHS of Eqn. (3-23) should be nonpositive (see Eqn. (3-12)). To prove Proposition 2 and Proposition 3 by contradiction, we are going to show that under certain conditions, the RHS of Eqn. (3-23) is positive. Here, we investigate the RHS of Eqn. (3-23). In particular, we are interested in the lower bound on $\sum_{i' \neq i} \mu_{i'l} V_{i'}$ subject to $\sum_{i' \neq i} V_{i'} = \gamma - \hat{\gamma}_{il}$ and $V_{b(l)} \leq \bar{\gamma}_{b(l)}$. In other words, we are interested in the solution of the following LP (LP3):

$$\begin{aligned} \min \quad & \sum_{i' \neq i} \mu_{i'l} V_{i'} \\ \text{s. t.} \quad & \sum_{i' \neq i} V_{i'} \leq \gamma - \hat{\gamma}_{il} \\ & - \sum_{i' \neq i} V_{i'} \leq \hat{\gamma}_{il} - \gamma \\ & V_{b(l)} \leq \bar{\gamma}_{b(l)} \\ & V_{i'} \geq 0 \end{aligned}$$

The objective function value for LP3 provides a lower bound on $\sum_{i' \neq i} \mu_{i'l} V_{i'}$. LP3 contains $(|I| - 1)$ variables and three inequality constraints. Here, we note that the optimal solution to LP3 is $V_{b^+(l)} = \gamma - \hat{\gamma}_{il} - \bar{\gamma}_{b(l)}, V_{b(l)} = \bar{\gamma}_{b(l)}$, and all other variables being zero. When $\mu_i^+ \leq$

0, the corresponding dual variables for the three constraints are $0, \mu_l^+$, and $(\mu_l^* - \mu_l^+)$; when $\mu_l^+ > 0$, the corresponding dual variables for the three constraints are $-\mu_l^+, 0$, and $(\mu_l^* - \mu_l^+)$. One can verify the optimality of such solution with strong duality. We show the optimal tableau for LP3 in Appendix A1.1.

The optimal solution mentioned above leads to the objective function value of $\mu_l^+(\hat{\gamma}_{il} - \gamma) + (\mu_l^* - \mu_l^+)\bar{\gamma}_{b(l)}$. Thus, from LP3 we have $\sum_{i' \neq i} \mu_{i'l} V_{i'} \geq \mu_l^+(\hat{\gamma}_{il} - \gamma) + (\mu_l^* - \mu_l^+)\bar{\gamma}_{b(l)}$.

We now revisit Eqn. (3-23). From LP3, we have a lower bound on the second term of its RHS, thus:

$$\sum_{i' \in I} \mu_{i'l} V_{i'} \geq \mu_{il} \hat{\gamma}_{il} + \mu_l^+(\hat{\gamma}_{il} - \gamma) + (\mu_l^* - \mu_l^+)\bar{\gamma}_{b(l)} \quad (3-24)$$

if $\sum_{i' \neq i} V_{i'} = \gamma - \hat{\gamma}_{il}$ and $V_{b(l)} \leq \bar{\gamma}_{b(l)}$ hold.

We next present Proposition 2 and Proposition 3.

Proposition 2 For $(i, l \in \mathbf{L}_i, b(l))$ with $\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il}$, $b^+(l) \neq i$, $\sum_{i' \in I} V_{i'} = \gamma$, $V_{b(l)} \leq \bar{\gamma}_{b(l)}$, and $\frac{\mu_l^+(\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)} \leq 0$, if $\sum_{i' \in I} \mu_{i'l} V_{i'} \leq 0$, then $\hat{\gamma}_{il} \leq 0$.

Proof (by contradiction)

If $V_i = \hat{\gamma}_{il}$, and since $\sum_{i' \in I} V_{i'} = \gamma$, then $\sum_{i' \neq i} V_{i'} = \gamma - \hat{\gamma}_{il}$. We also have $V_{b(l)} \leq \bar{\gamma}_{b(l)}$. Thus from Eqn. (3-24) we have:

$$\begin{aligned} \sum_{i'} \mu_{i'l} V_{i'} &\geq \mu_{il} \hat{\gamma}_{il} + \mu_l^+(\hat{\gamma}_{il} - \gamma) + (\mu_l^* - \mu_l^+)\bar{\gamma}_{b(l)} \\ &= \mu_l^* \bar{\gamma}_{b(l)} + \mu_l^+(\gamma - \bar{\gamma}_{b(l)}) + (\mu_{il} - \mu_l^+)\hat{\gamma}_{il} \end{aligned} \quad (3-25)$$

We examine the signs of $\mu_l^* \bar{\gamma}_{b(l)} + \mu_l^+ (\gamma - \bar{\gamma}_{b(l)})$ and $(\mu_{il} - \mu_l^+) \hat{\gamma}_{il}$ on the RHS of Eqn. (3-25) separately.

For $\mu_l^* \bar{\gamma}_{b(l)} + \mu_l^+ (\gamma - \bar{\gamma}_{b(l)})$: with $l \in \mathbf{L}_i$ and $b^+(l) \neq i$, it follows that $\mu_{il} > \mu_l^+$, thus $\mu_{il} - \mu_l^+ > 0$. Since $\frac{\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)} \leq 0$ and the denominator is positive, it follows that the numerator $\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)} \leq 0$, which is equivalent to $\mu_l^* \bar{\gamma}_{b(l)} + \mu_l^+ (\gamma - \bar{\gamma}_{b(l)}) \geq 0$.

For $(\mu_{il} - \mu_l^+) \hat{\gamma}_{il}$: we have $\mu_{il} - \mu_l^+ > 0$. To prove Proposition 2 using contradiction, we assume that $\hat{\gamma}_{il} > 0$, so it follows that $(\mu_{il} - \mu_l^+) \hat{\gamma}_{il} > 0$.

Thus, from Eqn. (3-25) we have $\sum_{i'} \mu_{i'l} V_{i'} > 0$, which leads to a contradiction. ■

Proposition 3 For $(i, l \in \mathbf{L}_i, b(l))$ with $\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il}$, $b^+(l) \neq i$, $\sum_{i' \in \mathbf{I}} V_{i'} = \gamma$, $V_{b(l)} \leq \bar{\gamma}_{b(l)}$, and $\frac{\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)} > 0$, if $\sum_{i' \in \mathbf{I}} \mu_{i'l} V_{i'} \leq 0$, then $\hat{\gamma}_{il} \leq \frac{\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)}$.

Proof (by contradiction)

If $V_i = \hat{\gamma}_{il}$, and since $\sum_{i' \in \mathbf{I}} V_{i'} = \gamma$, then $\sum_{i' \neq i} V_{i'} = \gamma - \hat{\gamma}_{il}$. We also have $V_{b(l)} \leq \bar{\gamma}_{b(l)}$. Thus from Eqn. (3-24) we have:

$$\begin{aligned} \sum_{i'} \mu_{i'l} V_{i'} &\geq \mu_{il} \hat{\gamma}_{il} + \mu_l^+ (\hat{\gamma}_{il} - \gamma) + (\mu_l^* - \mu_l^+) \bar{\gamma}_{b(l)} \\ &= \mu_l^* \bar{\gamma}_{b(l)} + \mu_l^+ (\gamma - \bar{\gamma}_{b(l)}) + (\mu_{il} - \mu_l^+) \hat{\gamma}_{il} \end{aligned} \tag{3-26}$$

To prove Proposition 3 using contradiction, we assume that $\hat{\gamma}_{il} = \frac{\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)} + \varepsilon$ with

$\varepsilon > 0$. From Eqn. (3-25) we have:

$$\sum_{i'} \mu_{i'l} V_{i'} \geq \mu_l^* \bar{\gamma}_{b(l)} + \mu_l^+ (\gamma - \bar{\gamma}_{b(l)}) + (\mu_{il} - \mu_l^+) \left(\frac{\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)} + \varepsilon \right)$$

or

$$\sum_{i'} \mu_{i'l} V_{i'} \geq \mu_l^* \bar{\gamma}_{b(l)} + \mu_l^+ (\gamma - \bar{\gamma}_{b(l)}) + \mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)} + (\mu_{il} - \mu_l^+) \varepsilon$$

After rearranging terms, we obtain,

$$\sum_{i'} \mu_{i'l} V_{i'} \geq \mu_l^* \bar{\gamma}_{b(l)} - \mu_l^* \bar{\gamma}_{b(l)} + \mu_l^+ (\gamma - \bar{\gamma}_{b(l)}) + \mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) + (\mu_{il} - \mu_l^+) \varepsilon$$

which leads to

$$\sum_{i'} \mu_{i'l} V_{i'} \geq (\mu_{il} - \mu_l^+) \varepsilon$$

Since $l \in \mathbf{L}_i$ and $b^+(l) \neq i$, it follows that $\mu_{il} > \mu_l^+$, thus $\mu_{il} - \mu_l^+ > 0$. With $\varepsilon > 0$ we have

$\sum_{i'} \mu_{i'l} V_{i'} > 0$, which leads to a contradiction. ■

From Proposition 2 and 3, it follows that for $(i, l \in \mathbf{L}_i, b(l))$ with $\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il}$ and $b^+(l) \neq$

i , $\hat{\gamma}_{il} = \max \left\{ 0, \frac{\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)} \right\}$ is a valid upper bound on V_i .

Utilizing the above results, we update bounds as follows: for $i \in \mathbf{I}$ and $l \in \mathbf{L}_i$, we first check if $\bar{\gamma}_{b(l)} < \gamma - \hat{\gamma}_{il}$ ($\hat{\gamma}_{il}$ is calculated from Eqn. (3-15)); if that is the case, we have:

$$\hat{\gamma}_{il} = \begin{cases} 0, & \text{if } b^+(l) = i \\ \max \left\{ 0, \frac{\mu_l^+ (\bar{\gamma}_{b(l)} - \gamma) - \mu_l^* \bar{\gamma}_{b(l)}}{(\mu_{il} - \mu_l^+)} \right\}, & \text{otherwise} \end{cases}$$

To illustrate, we consider the same example in Figure 3-1. Note that we have $b(\mathbf{L}_2) = 2$, $b^+(\mathbf{L}_2) = 1$, and from Table 3-1 we have $\bar{\gamma}_{b(\mathbf{L}_2)} = \bar{\gamma}_2 = 1/3$, and $1/3 < \gamma - \hat{\gamma}_{3,\mathbf{L}_2} = 2/5$. Thus, we update $\hat{\gamma}_{3,\mathbf{L}_2}$. Since $b^+(\mathbf{L}_2) \neq 3$, and $[\mu_{\mathbf{L}_2}^+ (\bar{\gamma}_2 - \gamma) - \mu_{\mathbf{L}_2}^* \bar{\gamma}_2] / (\mu_{3,\mathbf{L}_2} - \mu_{\mathbf{L}_2}^+) = 1/3 > 0$, we have $\hat{\gamma}_{3,\mathbf{L}_2} = 1/3$, and $\bar{\gamma}_3$ is updated to $1/3$.

The bounds calculated by our method are given in Table 3-2. For comparison, we also show the bounds which would have been obtained by FBBT and OBBT for the same example. We note that for this example, bounds on all V_i obtained by our method is tighter than bounds obtained from FBBT. For V_1 and V_2 , bounds obtained by our method is as tight as bounds obtained from OBBT.

Table 3-2. Bounds calculated by different methods

$\bar{\gamma}_i$	$i = 1$	$i = 2$	$i = 3$
FBBT	1	1/2	3/4
OBBT	3/4	1/3	1/11
Our method	3/4	1/3	1/3

Note: Calculation performed by FBBT and OBBT can be found in Appendix A1.2.

3.2.3 Complete procedure for bound tightening

The complete procedure, which combines the calculations described in the previous sections, is summarized below. The pseudocode, where we bring back indices j, k, t and thus $\mathbf{L}_{ik} = \{l: \mu_{ikl} > 0\}$, is as follows:

Complete procedure for bound tightening

```

For  $k \in \mathbf{K}$  do
  For  $j \in \mathbf{J}$  do
    For  $i \in \mathbf{I}$  do
       $\bar{\gamma}_{ijk} = \gamma_j$ 
      For  $l \in \mathbf{L}_{ik}$  do
         $\hat{\gamma}_{ijkl} = -\frac{\mu_l^* \gamma_j}{\mu_{ilk} - \mu_l^*}$ 
         $\bar{\gamma}_{ijk} = \min\{\bar{\gamma}_{ijk}, \hat{\gamma}_{ijkl}\}$ 
      End
    End
  For  $i \in \mathbf{I}$  do
    For  $l \in \mathbf{L}_{ik}$  do
      If  $\bar{\gamma}_{b(l),jk} < \gamma - \hat{\gamma}_{ijkl}$  then
        If  $b^+(l) = i$  then
           $\hat{\gamma}_{ijkl} = 0$ 
        Else
           $\hat{\gamma}_{ijkl} = \max\left\{0, \frac{\mu_{kl}^+(\bar{\gamma}_{b(l),jk} - \gamma) - \mu_l^* \bar{\gamma}_{b(l),jk}}{(\mu_{ilk} - \mu_{lk}^*)}\right\}$ 
        End
         $\bar{\gamma}_{ijk} = \min\{\bar{\gamma}_{ijk}, \hat{\gamma}_{ijkl}\}$ 
      End
    End
  End
End
Output:  $\bar{\gamma}_{ijk}$ 

```

3.3 Valid constraints

Since $R_{jkt} \leq [0,1]$, $(1 - R_{jkt}) \in [0,1]$, so multiplying $V_{ijkt} \leq \bar{\gamma}_{ijk}$ by $(1 - R_{jkt})$ yields:

$$(1 - R_{jkt})V_{ijkt} \leq (1 - R_{jkt})\bar{\gamma}_{ijk}, \quad i, j, k, t$$

and then:

$$V_{ijkt} - V_{ijkt}R_{jkt} \leq (1 - R_{jkt})\bar{\gamma}_{ijk}, \quad i, j, k, t$$

Note that $\hat{F}_{ijkt} = V_{ijkt}R_{jkt}$, thus:

$$V_{ijkt} - \hat{F}_{ijkt} \leq (1 - R_{jkt})\bar{\gamma}_{ijk}, \quad i, j, k, t \quad (3-27)$$

If we reintroduce indices j, k , and t , Eqn. (3-13) can be written as:

$$\sum_i (1 - \frac{\mu_{ikl}}{\mu_l^*}) V_{ijkt} \leq \gamma_j, \quad k, l, t$$

Multiplying both sides with $(1 - R_{jkt})$ leads to:

$$\sum_i (1 - \frac{\mu_{ikl}}{\mu_l^*}) V_{ijkt} (1 - R_{jkt}) \leq (1 - R_{jkt})\gamma_j, \quad j, k, l, t$$

or

$$\sum_i (1 - \frac{\mu_{ikl}}{\mu_l^*}) (V_{ijkt} - V_{ijkt} R_{jkt}) \leq (1 - R_{jkt})\gamma_j, \quad j, k, l, t \quad (3-28)$$

Since $\hat{F}_{ijkt} = V_{ijkt} R_{jkt}$, Eqn. (3-28) can be written as follows:

$$\sum_i (1 - \frac{\mu_{ikl}}{\mu_l^*}) (V_{ijkt} - \hat{F}_{ijkt}) \leq (1 - R_{jkt})\gamma_j, \quad j, k, l, t \quad (3-29)$$

Both Eqn. (3-27) and (3-29) are RLT constraints. Finally, we also have:

$$V_{ijkt} \leq \bar{\gamma}_{ijk}, \quad i, j, k, t \quad (3-30)$$

Eqn. (3-30) enforces upper bounds on V_{ijkt} which may be tighter than the bounds obtained through general purpose bound tightening techniques such as FBBT.

Eqn. (3-27) and (3-29) - (3-30) are added to model M^{UV} , resulting in model $M_{R,T}^{UV}$. We show an illustrative graph for our tightening methods using the example introduced in Figure 3-1 in Appendix A1.2. We also introduce model M_R^{UV} , which has the same constraints as $M_{R,T}^{UV}$, but without tightened bounds on V_{ijkt} (i.e., $\bar{\gamma}_{ijk} = \gamma_j$ in Eqn. (3-27), (3-29) - (3-30)). We summarize the models we consider in Table 3-3.

Table 3-3. Model description

Models	Description
M^{SB}	Original source-based model
M^{UV}	Source-based model reformulated with U_{ijkt} and V_{ijkt}
M_R^{UV}	M^{UV} + Eqn. (3-27), (3-29) - (3-30) $\bar{\gamma}_{ijk} = \gamma_j$
$M_{R,T}^{UV}$	M^{UV} + Eqn. (3-27), (3-29) - (3-30) $\bar{\gamma}_{ijk}$ obtained from our method

3.4 Computational results

We test our methods using 15 instances. Computational experiments are conducted on a Windows 10 machine with Intel Core i7 at 2.80 GHz and 8 GB of RAM. Models are coded in GAMS 28.2. We use BARON 19.7.13 with default options. Instances have five to eight streams, two to three blenders, four products, and four to six properties. Stream properties and product specifications are taken from Adhya et al. (Adhya, Tawarmalani, and Sahinidis 1999).

3.4.1 Case study

We first show the results for Instance 7 as a case study. It has eight streams, three blenders, four products, six properties and five time periods. An optimal schedule, with an objective function value of 3448.7, is shown in Figure 3-2 and the corresponding inventory profile in Figure 3-3. The model and solution statistics for different models for Instance 7 are given in Table 3-4. After 300 seconds M^{SB} has an optimality gap of 2.43% while $M_{R,T}^{UV}$ is solved to optimality in less than 50 seconds, indicating the effectiveness of the tighter bounds and RLT constraints.

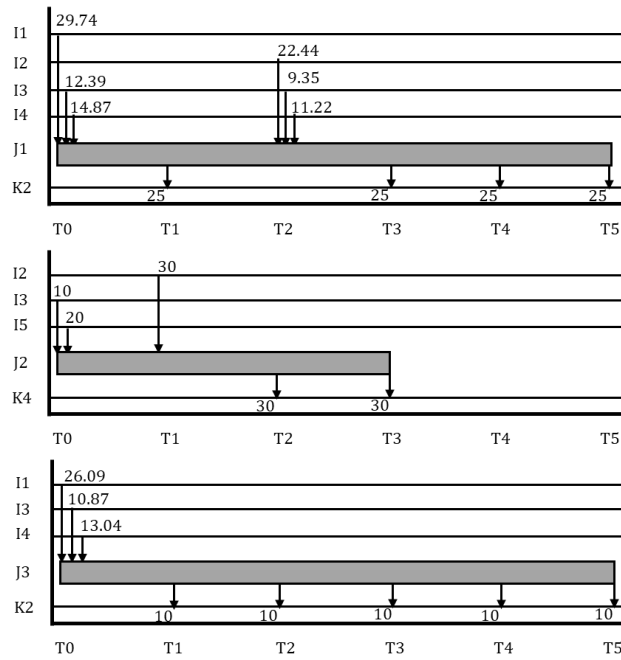


Figure 3-2. An optimal schedule for Instance 7.

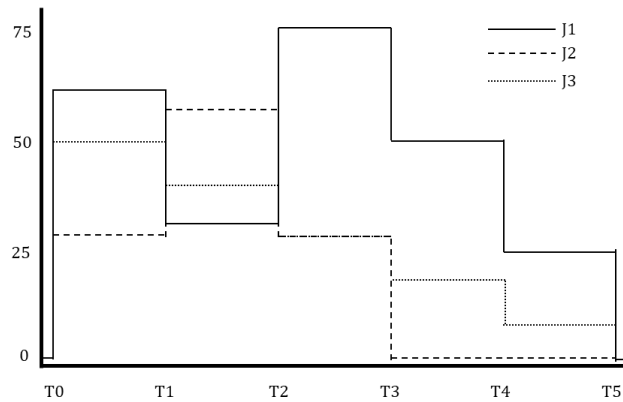


Figure 3-3. Inventory profile for the schedule shown in Figure 3-2.

Table 3-4. Model and solution statistics for Instance 7

	M^{SB}	M^{UV}	M_R^{UV}	$M_{R,T}^{UV}$
Con. Var.	1009	2161	2161	2161
Bin. Var.	72	72	72	72
Constraints	1921	2641	3649	2161
CPU Time (s)	>300	109.8	>300	47.7
Opt. Gap	2.4%	0	2.4%	0

3.4.2 MINLP models

We give the main characteristics of all 15 instances and the CPU time in Table 3-5. The CPU time for $M_{R,T}^{UV}$ includes run time for bound tightening method. Percentage in the parentheses indicates optimality gap.

Overall, we observe that $M_{R,T}^{UV}$ performs the best over the tested instances, with substantial improvement over the performance of M^{SB} for most instances. Further, the comparison of the CPU times between $M_{R,T}^{UV}$ and M^{UV} suggests that the proposed methods result in substantial computational improvement.

Table 3-5. Size of tested instances and CPU time for different MINLP models

Instance	Size					CPU Time (in seconds)			
	I	J	K	L	T	M^{SB}	M^{UV}	M_R^{UV}	$M_{R,T}^{UV}$
1	5	2	4	4	3	59	22	28	25
2	5	2	4	4	5	30.5	21.8	66.12	15.2
3	5	2	4	4	7	21.66	82.86	>300(0.18%)	71.66
4	5	2	4	6	7	211.4	287.46	>300(0.01%)	136.51
5	8	3	4	6	5	169.48	>300(0.01%)	282.83	137.63
6	5	2	4	6	3	35.4	55.76	118.71	34.95
7	8	3	4	6	5	>300(2.43%)	109.78	>300(2.41%)	47.72
8	5	2	4	6	3	57.55	83.43	43.6	50.72
9	8	3	4	6	3	7.89	70	27.72	11.68
10	8	3	4	6	5	17.56	95.16	231.19	74.77
11	5	2	4	4	7	41.88	8.1	9.8	9.1
12	5	2	4	4	7	38.8	7.8	20.3	8.8
13	5	2	4	4	9	69.02	15.83	14.2	14.97
14	5	2	4	4	9	56.35	21.2	13.95	21.49
15	5	2	4	4	3	104.08	19.53	26.21	22.78

We show a performance profile generated by the data above in Figure 3-4.

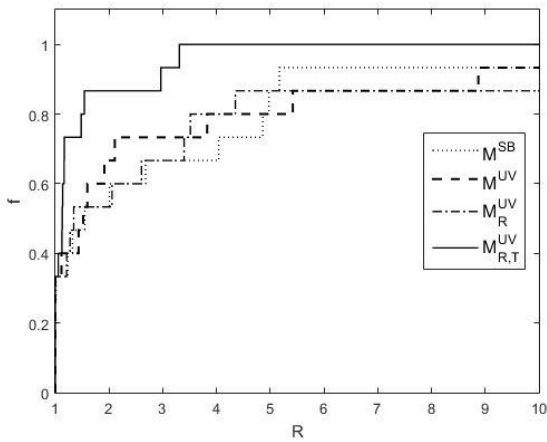


Figure 3-4. Performance profile for different MINLP models

3.4.3 MILP models

Mixed-integer linear models that approximate the MINLP models can be developed through discretization. In addition to providing approximate solutions, MILP models can also be used in solution methods (S. P. Kolodziej, Castro, and Grossmann 2013; S. P. Kolodziej et al. 2013; Gupte et al. 2017). Here, we allow the split fraction, R_{jkt} , to take values only from a discrete set $\mathbf{D}^R$, thereby linearizing nonlinear constraints Eqn. (3-1) and/or Eqn. (3-6). Specifically, we have $\mathbf{D}^R = \{0, \delta_1, \delta_2, \dots, \delta_n, 1\}$ with $\delta_1 = \delta_2 - \delta_1 = \dots = \delta_n - \delta_{n-1} = 1 - \delta_n = \delta$. The MILP obtained from such discretization, referred to as $_{L1}M$, is guaranteed to return only feasible solutions to the original MINLP. A relaxation of $_{L1}M$, referred to as $_{L2}M$, is obtained by introducing additional continuous variables to allow R_{jkt} to take any values in $[0,1]$. The resulting bilinear terms with two continuous variables are then relaxed using linear constraints. We test MILP models (both with and without our methods) over 20 instances including Instances 1 – 15, and five additional instances (Instances 16 - 20) modified from D’Ambrosio et al. (D’Ambrosio, Linderoth, and Luedtke 2011) with more streams, blenders

and products. The characteristics of Instances 16 – 20 are given in Table 3-6. Performance profiles for the MILP models are shown in Figure 3-5.

Table 3-6. Size of Instance 16 - 20

Instance	I	J	K	L	T
16	15	10	10	1	1
17	15	10	10	2	1
18	15	8	10	4	1
19	15	8	10	4	1
20	15	10	10	1	1

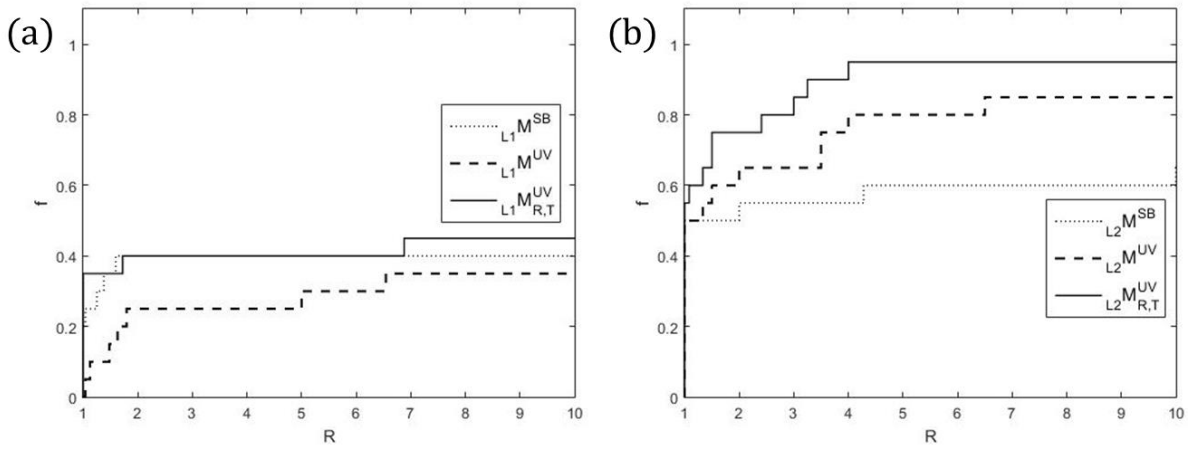


Figure 3-5. Performance profile for two MILP models with $\delta = 0.01$ (left) and $\delta = 0.1$ (right)

Overall we observe that $L1M^{UV}$ and $L2M^{UV}$ perform best over the tested instances, indicating the effectiveness of our method. For $L2M$, we see substantial improvement from the reformulation ($L2M^{UV}$) and bound tightening ($L2M^{UV,R,T}$) compared to the original model.

3.4.4 Decomposition method

We further test our methods on an MILP-MINLP decomposition method for multiperiod blending proposed by (Lotero et al. 2016). We briefly describe the method below: (1). A new binary variable Y_{jt} is introduced, which equals to 1 if blender j feeds products at time point t . (2). A relaxed problem (MILP) is solved in which Eqn. (3-6), the constraint that contains

bilinear term, is replaced using McCormick envelopes with tightened bounds. (3). Binary Y_{jt} is fixed to the value obtained from the solution to the relaxed problem, and a reduced problem (MINLP) containing all constraints in $M_{R,T}^{UV}$ is solved (“reduced” in the sense that after fixing Y_{jt} , some X_{jkt} are also fixed, resulting in a reduced feasible space compared to $M_{R,T}^{UV}$). Solving one relaxed problem and one reduced problem completes one iteration, from which an upper bound and a lower bound (if the reduced problem is feasible) are obtained. A feasibility or optimality cut is added to the relaxed problem after solving the reduced problem in each iteration. More details about the decomposition method can be found in (Lotero et al. 2016).

We show computational results for 5 instances (Instance 21 – 25) modified from (D’Ambrosio, Linderoth, and Luedtke 2011) in Table 3-7. We set the maximum number of iterations to five, and time limits for the relaxed problem and reduced problem are set at 10 seconds and 30 seconds, respectively. We use CPLEX 12.9 to solve the relaxed problem.

Table 3-7. Size of tested instances and CPU time for MINLP and decomposition

Inst.	Size					MINLP ($M_{R,T}^{UV}$)		Decomposition		
	I	J	K	L	T	Opt. Gap	CPU Time(s)	# of Iter.	Opt. Gap	CPU Time(s)
21	15	10	10	1	3	1.58%	300	2	0	50.8
22	15	10	10	2	3	11.79%	300	2	0	42.6
23	15	8	10	4	3	4.06%	300	5	0.1%	200
24	15	8	10	4	3	0.01%	300	1	0	16.23
25	15	10	10	1	3	0.23%	300	1	0	18.6

Model $M_{R,T}^{UV}$ does not solve Instances 21 – 25 to global optimality in 300 seconds, while the decomposition method solves four of them to global optimality in less than 60 seconds. For Instance 23, the decomposition method does not close the optimality gap after five iterations at 200 seconds. However, the optimality gap is smaller compared to the MINLP model after 300 seconds. We also note that decomposition method using $M_{R,T}^{UV}$ outperforms the same

method based on M^{SB} (data not shown). We observe that, with the addition of the RLT constraints with tightened variable bounds, decomposition method using M_{RT}^{UV} typically closes the optimality gap within fewer iterations.

3.5 Conclusion

We developed variable bound tightening methods, based on multiple constraints, for multiperiod blending. We first proposed a reformulation of the constraints involving bilinear terms using lifting. We introduced a preprocessing method to tighten the bounds on the lifted variables using multiple constraints. The reformulation and the selection of constraints to be considered for bound tightening are based on the understanding of the physical system. We proposed valid constraints derived from Reformulation-Linearization Technique (RLT) that utilize the bounds on the lifted variables to further tighten the formulation. Computational results indicate the effectiveness of our methods in reducing the computational requirements. Finally, the proposed methods can be coupled with other solution strategies for multiperiod blending problem.

Chapter 4

Tightening methods based on nontrivial bounds on bilinear terms

4.1 Introduction

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 xy with nonnegative variables $x \in [\underline{x}, \bar{x}]$ and $y \in [\underline{y}, \bar{y}]$ and the set $\mathbf{S} = \{(w, x, y) \in \mathbb{R}_+^3 : w = xy, \underline{x} \leq x \leq \bar{x}, \underline{y} \leq y \leq \bar{y}\}$. Using the method proposed by McCormick (McCormick 1976) leads to four linear inequalities containing $\underline{x}, \bar{x}, \underline{y}$, and $\bar{y}$ which describe the convex hull of $\mathbf{S}$.

If w is also lower bounded by a positive parameter $\underline{w} > \underline{x}\underline{y}$, then $\underline{w}$ is said to be a nontrivial lower bound on xy . Similarly, if w is upper bounded by a positive parameter $\bar{w} < \bar{x}\bar{y}$, then $\bar{w}$ is said to be a nontrivial upper bound on xy . For bilinear term with nontrivial bounds, consider the set $\mathbf{S}_1^+ = \{(w, x, y) \in \mathbb{R}_+^3 : w = xy, \underline{w} \leq w \leq \bar{w}, \underline{x} \leq x \leq \bar{x}, \underline{y} \leq y \leq \bar{y}\}$, whose convex hull has been studied by Belotti et al. (Belotti, Miller, and Namazifar 2010; 2011). Specifically, they showed that the convex hull of $\mathbf{S}_1^+$ 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 representations for bilinear terms with bounds on the product, and derived closed-form representations containing second-order cone constraints (Anstreicher, Burer, and Park 2020).

In this chapter, we study the following set:

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

which can be viewed as a generalization of $\mathbf{S}_1^+$ ($\mathbf{S}_1$ becomes $\mathbf{S}_1^+$ when $Z = 1$). we derive a family of valid linear constraints for $\mathbf{S}_1$, and further 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 chapter coincide with a subset of the “lifted tangent inequalities”. 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.

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 problem with binary variables has 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 for such problem (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.

4.2 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.

4.2.1 Problem statement

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

Given are:

- α_i : Unit cost of stream i
- β_k : Price of product k
- γ_j : Capacity of pool j
- v_{jk}^L : Lower bound on positive flow between pool j and product k
- v_{jk}^U : Capacity of the pipeline between pool j and product k
- π_{il} : Value of property l for stream i
- π_{kl}^U : 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.

4.2.2 Nonlinear models for the pooling problem

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). Here, 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:

$$\sum_i F_{ij} \leq \gamma_j, \quad j \quad (4-1)$$

Product demand:

$$\sum_i \sum_j \hat{F}_{ijk} \leq \omega_k, \quad k \quad (4-2)$$

Product specifications:

$$\sum_i \sum_j \pi_{il} \hat{F}_{ijk} \leq \pi_{kl}^U \sum_i \sum_j \hat{F}_{ijk}, \quad k, l \quad (4-3)$$

Upper bound on the flows from pools to products:

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

Stream splitting:

$$\hat{F}_{ijk} = F_{ij}R_{jk}, \quad i, j, k \quad (4-5)$$

Note that Eqn. (4-5) is an equality constraint with a bilinear term.

For split fraction R_{jk} we have:

$$\sum_k R_{jk} = 1, \quad j \quad (4-6)$$

Eqns. (4-5) and (4-6) enforce that there is no flow accumulation in pools.

Reformulation–Linearization Technique (RLT) constraints can be added to strengthen the formulation. Summing over index k on both sides of Eqn.(4-5), we have:

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

which, combined with Eqn. (4-6), leads to:

$$\sum_k \hat{F}_{ijk} = F_{ij}, \quad i, j \quad (4-7)$$

Another RLT constraint can be obtained by multiplying both sides of Eqn. (2-1) with R_{jk} (a nonnegative variable):

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

which, combined with Eqn. (4-5), leads to:

$$\sum_i \hat{F}_{ijk} \leq \gamma_j R_{jk}, \quad j, k \quad (4-8)$$

The objective function is profit maximization:

$$\max \sum_i \sum_j \left(\sum_k \beta_k \hat{F}_{ijk} - \alpha_i F_{ij} \right) \quad (4-9)$$

Eqns. (4-1)- (4-9) comprise a nonlinear model for the pooling problem which contains only continuous variables and is henceforth referred to as M^{CON} . In practice, in addition to the pipeline capacity modeled in Eqn. (4-4), 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 given parameter. Let v_{jk}^L denote such parameter ($v_{jk}^L < v_{jk}^U$). 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:

$$\bar{F}_{jk} = \sum_i \hat{F}_{ijk}, \quad j, k \quad (4-10)$$

$$v_{jk}^L Z_{jk} \leq \bar{F}_{jk} \leq v_{jk}^U Z_{jk}, \quad j, k \quad (4-11)$$

Eqn. (4-11) ensures that when $Z_{jk} = 0$, $\bar{F}_{jk} = 0$; when $Z_{jk} = 1$, $\bar{F}_{jk} \in [v_{jk}^L, v_{jk}^U]$.

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

$$\frac{v_{jk}^L}{\gamma_j} Z_{jk} \leq R_{jk} \leq Z_{jk}, \quad j, k \quad (4-12)$$

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 $\bar{F}_{jk} \geq v_{jk}^L$ so the lower bound on R_{jk} in this case should be v_{jk}^L/γ_j and by definition $R_{jk} \leq 1$. Thus, R_{jk} is now also a semi-continuous variable.

We again consider profit maximization with additional fixed cost terms:

$$\max \sum_i \sum_j \left(\sum_k \beta_k \hat{F}_{ijk} - \alpha_i F_{ij} \right) - \sum_j \sum_k \alpha_{jk}^F Z_{jk} \quad (4-13)$$

Eqns. (4-1) - (4-3), (4-5) and (4-7) - (4-13) comprise a mixed-integer nonlinear model for the pooling problem with semi-continuous flow, henceforth referred to as M^{SC} . We note that M^{SC} and M^{CON} define the same feasible space for variables F_{ij} , R_{jk} , and $\hat{F}_{ijk}$ when $v_{jk}^L = 0$ and $Z_{jk} = 1$ for each (j, k) pair.

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:

$$U_k \geq \varphi_k - \sum_j \bar{F}_{jk}, \quad k \quad (4-14)$$

and the objective function is:

$$\min \sum_i \sum_j \alpha_i F_{ij} + \sum_j \sum_k \alpha_{jk}^F Z_{jk} + \sum_k \beta_k^P U_k \quad (4-15)$$

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

4.2.3 Nontrivial bounds on bilinear terms

Summing over index i for the constraints in Eqn. (4-5), we obtain:

$$\sum_i \hat{F}_{ijk} = R_{jk} \sum_i F_{ij}, \quad j, k \quad (4-16)$$

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

$$\tilde{F}_j = \sum_i F_{ij}, \quad j \quad (4-17)$$

From Eqn. (4-7) and Eqn. (4-17), we can re-write Eqn. (4-16) as:

$$\bar{F}_{jk} = \tilde{F}_j R_{jk}, \quad j, k \quad (4-18)$$

Eqn.(4-18) is an equality constraint with a bilinear term; it is implied from constraints in both M^C and M^S . Note that $\tilde{F}_j$ is upper bounded by γ_j since $\sum_i F_{ij} \leq \gamma_j$ (see Eqn. (4-1)) and R_{jk} is upper bounded by 1. Thus, from the right-hand-side (RHS) of Eqn. (4-18) we know that $\bar{F}_{jk}$ is upper bounded by γ_j . However, $\bar{F}_{jk}$ is also upper bounded by v_{jk}^U since $\sum_i \hat{F}_{ijk} \leq v_{jk}^U$ (see Eqn. (4-4)), 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^{CON} and M^{SC} .

We next examine the lower bounds on both sides of Eqn. (4-18) for model M^S when $Z_{jk} = 1$. We note that in M^S , $\tilde{F}_j$ is also semi-continuous since we have:

$$\tilde{F}_j \geq v_{jk}^L Z_{jk}, \quad j, k \quad (4-19)$$

which is implied by Eqn. (4-11) and Eqn. (4-18). In this case, from Eqn. (4-11) we have $\bar{F}_{jk} \geq \iota_{jk}$; we also have $\tilde{F}_j \geq v_{jk}^L$, and from Eqn. (4-12) we have $R_{jk} \geq v_{jk}^L / \gamma_j$. We note that from the RHS of Eqn. (4-18) with bounds on $\tilde{F}_j$ and R_{jk} mentioned above, one can only derive the

lower bound on $\bar{F}_{jk}$ as $(v_{jk}^L)^2/\gamma_j$, which is smaller (thus less tight) than v_{jk}^L since typically we have $v_{jk}^L < \gamma_j$.

Definition 1 Consider a bilinear term $w = xy$ with $x \geq \underline{x}$, $y \geq \underline{y}$, and $w \geq \underline{w}$. $\underline{w}$ is said to be a nontrivial lower bound on w if $\underline{w} > \underline{x}\underline{y}$.

Definition 2 Consider a bilinear term $w = xy$ with $x \leq \bar{x}$, $y \leq \bar{y}$, and $w \leq \bar{w}$. $\bar{w}$ is said to be a nontrivial upper bound on w if $\bar{w} < \bar{x}\bar{y}$.

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

Here we are interested in the set defined as follow:

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

with $\underline{w} > \underline{x}\underline{y}$, $\bar{w} < \bar{x}\bar{y}$, and $\underline{w} < \bar{w}$. Set $\mathbf{S}_1$ contains structures in M^{SC} ; for a (j, k) pair one can consider $\bar{F}_{jk}$ as w , $\tilde{F}_j$ as x , R_{jk} as y , Z_{jk} as Z , and Eqns. (4-5), (4-11), (4-12), and (4-18) are similar to constraints that define $\mathbf{S}_1$.

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

$$\mathbf{S}_1^+ = \left\{ (w, x, y) \in \mathbb{R}_+^3: \underline{x} \leq x \leq \bar{x}, \underline{y} \leq y \leq \bar{y}, \underline{w} \leq w \leq \bar{w}, w = xy \right\} \quad (4-20)$$

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

4.2.4 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:

$$w \geq \underline{y}x + \underline{x}y - \underline{x}\underline{y} \quad (4-21)$$

$$w \geq \bar{y}x + \bar{x}y - \bar{x}\bar{y} \quad (4-22)$$

$$w \leq \underline{y}x + \bar{x}y - \bar{x}\underline{y} \quad (4-23)$$

$$w \leq \bar{y}x + \underline{x}y - \underline{x}\bar{y} \quad (4-24)$$

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

$$\mathbf{S}_2^+ = \{(w, x, y) \in \mathbb{R}_+^3: \underline{x} \leq x \leq \bar{x}, \underline{y} \leq y \leq \bar{y}, \underline{w} \leq w \leq \bar{w}, \text{Eqns. (4-21) - (4-24)}\}$$

and set $\mathbf{S}_3 = \{w \in \mathbb{R}_+: w = \bar{w}\}$. The feasible space defined by $\mathbf{S}_1^+ \cap \mathbf{S}_3$ and $\mathbf{S}_2^+ \cap \mathbf{S}_3$ are shown in Figure 4-1. The intersection of $\mathbf{S}_1^+$ and $\mathbf{S}_3$ is the solid curve $xy = \bar{w}$, 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 $xy = \bar{w}$, Eqns. (4-21) - (4-24) that define $\mathbf{S}_2^+$ are generated with $\underline{x} = 1, \bar{x} = 3, \underline{y} = 1/3$, and $\bar{y} = 1$. In the next section, we derive a family of valid constraints for $\mathbf{S}_1$ (thus valid for $\mathbf{S}_1^+$ as well) that tightens $\mathbf{S}_2^+$.

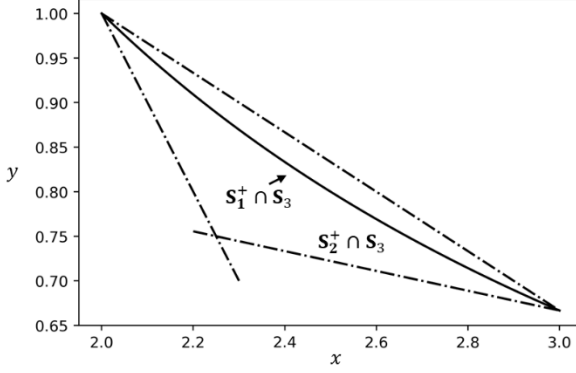


Figure 4-1. Illustrative graph for bilinear terms xy with $x \in [1,3], y \in [1/3,1]$ and its relaxation when the nontrivial upper bound $\bar{w} = 2$ is active.

4.3 Valid constraints

We first present a family of valid 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.

4.3.1 A family of valid constraints

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

Proposition 1 $\rho^2 x + \bar{w}y + 2\rho(\sigma_1 w + \sigma_2 Z) \geq 0$ with $\sigma_1 = (\sqrt{\underline{w}\bar{w}} - \bar{w})/(\bar{w} - \underline{w})$, $\sigma_2 = \bar{w}(\underline{w} - \sqrt{\underline{w}\bar{w}})/(\bar{w} - \underline{w})$, and parameter $\rho > 0$ 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:

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

which is equivalent to:

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

One can verify $\rho^2 x + \bar{w}y + 2\rho(\sigma_1 w + \sigma_2 Z) \geq 0$ is valid for $\mathbf{S}_1^0$ 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. (4-20):

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

and the proposed constraint becomes:

$$\rho^2 x + \bar{w}y + 2\rho(\sigma_1 w + \sigma_2) \geq 0 \quad (4-25)$$

Assume $(w, x, y) \in \mathbf{S}_1^+$, we first examine the terms $\rho^2 x + \bar{w}y$. Consider the valid inequality

$(\rho\sqrt{x} - \sqrt{\bar{w}y})^2 \geq 0$, which, after expanding the left-hand-side (LHS), we obtain

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

and thus

$$\rho^2 x + \bar{w}y \geq 2\rho\sqrt{\bar{w}xy} \quad (4-26)$$

Since $(w, x, y) \in \mathbf{S}_1^+$, we have $w = xy$. Thus, Eqn. (4-26) can be re-written as

$$\rho^2 x + \bar{w}y \geq 2\rho\sqrt{\bar{w}w} \quad (4-27)$$

With Eqn. (4-27), we know that the LHS of Eqn. (4-25) is lower bounded by the following:

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

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

$$\rho^2 x + \bar{w}y + 2\rho(\sigma_1 w + \sigma_2) \geq 2\rho(\sqrt{\bar{w}w} + \sigma_1 w + \sigma_2) \quad (4-28)$$

We next show that the RHS of Eqn. (4-28) is nonnegative by showing $\sqrt{\bar{w}w} + \sigma_1 w + \sigma_2 \geq 0$ (recall that we have $\rho > 0$). We first examine the zeros of the following quadratic function w.r.t $\sqrt{w}$:

$$\sigma_1(\sqrt{w})^2 + \sqrt{\bar{w}w} + \sigma_2 = \sqrt{\bar{w}w} + \sigma_1 w + \sigma_2 = 0 \quad (4-29)$$

We note that $\sqrt{\underline{w}}$ is one zero for such function, since

$$\sqrt{\underline{w}\bar{w}} + \sigma_1 \underline{w} + \sigma_2 = \sqrt{\underline{w}\bar{w}} + \frac{\underline{w}(\sqrt{\underline{w}\bar{w}} - \bar{w})}{\bar{w} - \underline{w}} + \frac{\bar{w}(\underline{w} - \sqrt{\underline{w}\bar{w}})}{\bar{w} - \underline{w}} = \frac{\sqrt{\underline{w}\bar{w}}(\bar{w} - \underline{w}) + \sqrt{\underline{w}\bar{w}}(\underline{w} - \bar{w}) + \bar{w}\underline{w} - \bar{w}\underline{w}}{\bar{w} - \underline{w}} = 0$$

and $\sqrt{\bar{w}}$ is the other zero for such function, since

$$\sqrt{\bar{w}w} + \sigma_1 \bar{w} + \sigma_2 = \bar{w} + \frac{\bar{w}(\sqrt{\underline{w}\bar{w}} - \bar{w})}{\bar{w} - \underline{w}} + \frac{\bar{w}(\underline{w} - \sqrt{\underline{w}\bar{w}})}{\bar{w} - \underline{w}} = \frac{\bar{w}(\bar{w} - \underline{w}) + \bar{w}(\underline{w} - \bar{w})}{\bar{w} - \underline{w}} = 0$$

We further note that the coefficient of the quadratic term, σ_1 , in Eqn. (4-29), is negative since $\underline{w} < \bar{w}$ (see Proposition 1 for the definition of σ_1). Thus, we have

$$\sqrt{\bar{w}w} + \sigma_1 w + \sigma_2 \geq 0 \quad (4-30)$$

for $\sqrt{\underline{w}} \leq \sqrt{w} \leq \sqrt{\bar{w}}$, which is equivalent to $\underline{w} \leq w \leq \bar{w}$. Combining Eqn. (4-28) and (4-30), we have:

$$\rho^2 x + \bar{w}y + 2\rho(\sigma_1 w + \sigma_2) \geq 2\rho(\sqrt{\bar{w}w} + \sigma_1 w + \sigma_2) \geq 0 \quad (4-31)$$

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

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

$$\rho^2 x + \bar{w}y + 2\rho(\sigma_1 w + \sigma_2 Z) \geq 0 \quad (4-32)$$

is valid for $\mathbf{S}_1$. ■

The family of constraints in Eqn. (4-32) can lead to some strong inequalities, particularly when $w = \underline{w}$ or $w = \overline{w}$ (i.e., when one of the nontrivial bounds is active). We show an illustrative graph in Figure 4-2 for bilinear terms $w = xy$ when one of its nontrivial bounds $\overline{w} = 2$ or $\underline{w} = 1$ is active. Blue curve represents $xy = 2$ and black curve represents $xy = 1$. Dashed blue and black lines represent the intersection of Eqn. (4-25) and $w = 2$ and $w = 1$, respectively, with $\rho = 1$. Dotted blue and black lines represent the intersection of Eqn. (4-25) and $w = 2$ and $w = 1$, respectively, with $\rho = 2/3$. Coordinates for points of tangency are shown in parentheses.

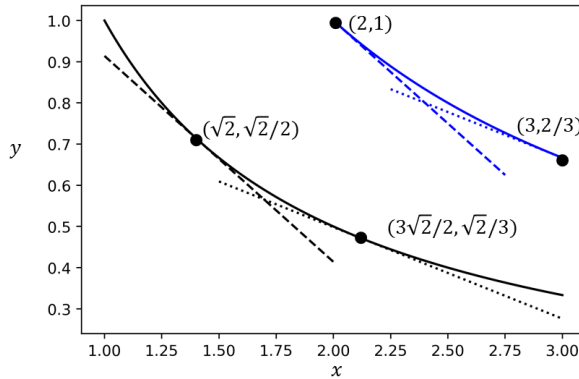


Figure 4-2. Illustrative graph for bilinear terms $w = xy$ with $x \in [1,3], y \in [1/3,1]$ when one of its nontrivial bounds $\overline{w} = 2$ or $\underline{w} = 1$ is active.

Remark 1 When $w = \overline{w}$, Eqn. (4-32) becomes $\rho^2 x + \overline{w}y - 2\rho\overline{w} \geq 0$. When $\rho > 0$, one can easily verify that for the branch of the $xy = \overline{w}$ hyperbola with both x and y positive, line $\rho^2 x + \overline{w}y - 2\rho\overline{w} = 0$ is tangent to such hyperbola at point $(\overline{w}/\rho, \rho)$ (the slope for the tangent line at such point is $(-\rho^2/\overline{w})$).

Remark 2 When $w = \underline{w}$, Eqn. (4-32) becomes $\rho^2 x + \overline{w}y - 2\rho\sqrt{\underline{w}\overline{w}} \geq 0$. When $\rho > 0$, one can easily verify that for the branch of the $xy = \underline{w}$ hyperbola with both x and y positive, line $\rho^2 x + \overline{w}y - 2\rho\sqrt{\underline{w}\overline{w}} = 0$ is tangent to such hyperbola at point $(\sqrt{\underline{w}\overline{w}}/\rho, \rho\sqrt{\underline{w}/\overline{w}})$.

Remark 3 As will be shown later, for the pooling problem studied here, we can identify bounds on variables involved in $\mathbf{S}_1^+$ as follows:

$$\mathbf{S}_1^+ = \{(w, x, y) \in \mathbb{R}_+^3: \underline{w} \leq x \leq \overline{w}, \underline{w}/\overline{w} \leq y \leq 1, \underline{w} \leq w \leq \overline{w}, w = xy\}$$

with $\gamma > \overline{w} > \underline{w}$. Thus, when $w = \overline{w}$ we have $x \in [\overline{w}, \gamma], y \in [\overline{w}/\gamma, 1]$; when $w = \underline{w}$ we have $x \in [\underline{w}, \gamma], y \in [\underline{w}/\gamma, 1]$. If we have a ρ with the corresponding $\rho^2 x + \overline{w}y - 2\rho\overline{w} = 0$ being tangent to $xy = \overline{w}$ at point $(\overline{w}/\rho, \rho) \in [\overline{w}, \gamma] \times [\overline{w}/\gamma, 1]$, then the line $\rho^2 x + \overline{w}y - 2\rho\sqrt{\overline{w}\underline{w}} = 0$ is tangent to $xy = \underline{w}$ at $(\sqrt{\overline{w}\underline{w}}/\rho, \rho\sqrt{\underline{w}/\overline{w}}) \in [\underline{w}, \gamma] \times [\underline{w}/\gamma, 1]$ since $\sqrt{\underline{w}/\overline{w}} < 1$.

Remark 4 By setting $\underline{w} = 0$ and $Z = 1$, from Proposition 1 we have

$$\rho^2 x + \overline{w}y - 2\rho w \geq 0 \quad (4-33)$$

which is valid for

$$\mathbf{S}_1^C = \{(w, x, y) \in \mathbb{R}_+^3: x \leq x^U, y \leq y^U, w \leq \overline{w}, w = xy\} \quad (4-34)$$

4.3.2 Generation of strong valid constraints

Eqn. (4-32) 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.

4.3.2.1 Generation based on constraint violation maximization

We consider the following optimization problem:

$$\min_{\rho} \rho^2 x^* + \overline{w}y^* + 2\rho(\sigma_1 w^* + \sigma_2 Z^*) \quad (4-35)$$

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

4.3.2.2 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 $(w^*, x^*, y^*, Z^*) \in \mathbf{S}_3 = \{w \in \mathbb{R}_+ : w = \bar{w}\}$, that is, the nontrivial upper bound is active, $w^* = \bar{w}$ (in this case $Z^* = 1$). Of particular interest is the point $(w^*, x^*, y^*, Z^*) \in \mathbf{S}_3$ with $x^* y^* < \bar{w}$. 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 $xy = \bar{w}$ that has the minimum distance to (w^*, x^*, y^*, Z^*) by considering the following optimization problem:

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

which can be viewed as minimizing the weighted 1-norm distance between (x^*, y^*) and (x, y) . Note that points on the curve $xy = \bar{w}$ can be represented using $(\bar{w}/\rho, \rho)$ with ρ being a variable; furthermore, consider bounds on x and y when $xy = \bar{w}$, we have $y \in [\bar{w}/\bar{x}, \bar{w}]$, thus $\rho \in [\bar{w}/\bar{x}, \bar{y}]$ and the above optimization problem can be re-written as:

$$\min_{\rho} \left\{ \frac{1}{\bar{x} - \underline{x}} \left| \frac{\bar{w}}{\rho} - x^* \right| + \frac{1}{\bar{y} - \underline{y}} |\rho - y^*| : (\bar{w}/\bar{x}) \leq \rho \leq \bar{y} \right\} \quad (4-36)$$

We claim that the solution to the above problem is the following (see proof in Appendix A2.1):

(1) If $y^* \leq \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} \leq \bar{w}/x^*$, then $\rho = \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})}$.

(2) If $\sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} < y^*$, then $\rho = y^*$.

(3) If $\sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} > \bar{w}/x^*$, then $\rho = \bar{w}/x^*$.

After obtaining ρ , we have the point $(\bar{w}/\rho, \rho)$ on the curve $xy = \bar{w}$ that has the minimum distance to (w^*, x^*, y^*, Z^*) . We then generate Eqn. (4-32) with such ρ . Recall that when the nontrivial upper bound $\bar{w}$ is active, Eqn. (4-32) is tangent to the curve $xy = \bar{w}$ at point $(\bar{w}/\rho, \rho)$.

4.4 Solution methods

In this section we present different methods for generating the proposed constraints for model with only continuous variables, M^{CON} , and model with semi-continuous variables, M^{SC} .

4.4.1 Methods for model with only continuous variables

For model M^{CON} , we consider the following constraint obtained from summing over index $i \in \mathbf{I}^* \subseteq \mathbf{I}$ for the constraints in Eqn. (4-5):

$$\sum_{i \in \mathbf{I}^*} \hat{F}_{ijk} = R_{jk} \sum_{i \in \mathbf{I}^*} F_{ij}, \quad j, k \quad (4-37)$$

We note that $\sum_{i \in \mathbf{I}^*} \hat{F}_{ijk} \leq v_{jk}^U$ and $\sum_{i \in \mathbf{I}^*} F_{ij} \leq \gamma_j$, thus from Remark 4 we have the following valid constraint for M^{CON} :

$$\rho^2 \sum_{i \in \mathbf{I}^*} F_{ij} + v_{jk}^U R_{jk} - 2\rho \sum_{i \in \mathbf{I}^*} \hat{F}_{ijk} \geq 0 \quad (4-38)$$

We next present two constraint generation methods for M^{CON} that determine the value of ρ and the selection of set $\mathbf{I}^*$ for Eqn. (4-38).

4.4.1.1 Generation at the root node

Eqn. (4-38) can be generated at the root node in multiple rounds. At each round, we solve a linear relaxation of M^{CON} , 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 $M_m^{\text{CON-L}}$ contains all constraints in M^{CON} , except that the nonlinear constraint Eqn. (4-5) is replaced by:

$$\hat{F}_{ijk} \geq \gamma_j R_{jk} + F_{ij} - \gamma_j, \quad i, j, k \quad (4-39)$$

$$\hat{F}_{ijk} \leq \gamma_j R_{jk}, \quad i, j, k \quad (4-40)$$

$$\hat{F}_{ijk} \leq F_{ij}, \quad i, j, k \quad (4-41)$$

$M_m^{\text{CON-L}}$ also contains the following constraint:

$$\rho_{jkm'}^2 \sum_{i \in \mathbf{I}_{jkm'}^*} F_{ij} + v_{jk}^U R_{jk} - 2\rho_{jkm'} \sum_{i \in \mathbf{I}_{jkm'}^*} \hat{F}_{ijk} \geq 0, \quad (j, k, m') \in \mathbf{C}_m \quad (4-42)$$

where $\mathbf{C}_m$ contains (j, k, m') combinations that lead to Eqn. (4-42) in all previous rounds, and the set $\mathbf{I}_{jkm'}^*$ is defined as follows: At each round, we solve $M_m^{\text{CON-L}}$ and, for each (j, k) pair,

define set $\mathbf{I}_{jkm}^* = \{i: \hat{F}_{ijk}^* > 0\}$. We check the sign of $[4 \left(\sum_{i \in \mathbf{I}_{jkm}^*} \hat{F}_{ijk}^* \right)^2 - 4v_{jk}^U R_{jk}^* \sum_{i \in \mathbf{I}_{jkm}^*} F_{ij}^*]$; if positive, that means there exist a parameter ρ that leads to a constraint in Eqn. (4-42) violated by the current solution to $M_m^{\text{CON-L}}$. We then calculate $\rho = \sum_{i \in \mathbf{I}_{jkm}^*} \hat{F}_{ijk}^* / \sum_{i \in \mathbf{I}_{jkm}^*} F_{ij}^*$ (such ρ will lead to a constraint that is violated by the current solution to $M_m^{\text{CON-L}}$ by the greatest margin). We also update set $\mathbf{C}_{m+1}$, which contains index for Eqn. (4-42). We then solve $M_{m+1}^{\text{CON-L}}$, which contains Eqn. (4-42) that are generated in previous rounds. We repeat until no new constraints are generated or we reach the maximum number of constraint generation rounds (σ^U). The pseudocode of the aforementioned method is given in Algorithm 1.

Algorithm 1. Constraint generation at root node

Inputs: $c = \text{True}$, $m = 0$, σ^U , $\mathbf{I}_{jkm}^* = \emptyset$, and $\mathbf{C}_m = \emptyset$

While $c = \text{True}$ **AND** $m < \sigma^U$ **do**

$c = \text{False}$

 Solve $M_m^{\text{CON-L}}$.

 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 $4 \left(\sum_{i \in \mathbf{I}_{jkm}^*} \hat{F}_{ijk}^* \right)^2 - 4v_{jk}^U R_{jk}^* \sum_{i \in \mathbf{I}_{jkm}^*} F_{ij}^* > 0$ **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 4-1

Table 4-1. Parameters for the illustrative example

	α_i	β_k	$v_{J1,k}^U$	$\pi_{i,L1}$	$\pi_{k,L1}^U$	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^{CON} leads to a solution with optimal objective function value of 20.5. Solving the illustrative example with $M_0^{\text{CON-L}}$ leads to a solution that has an objective function value of 21. The optimal solution to $M_0^{\text{CON-L}}$ is shown in Figure 4-3 (where we drop index j for simplicity).

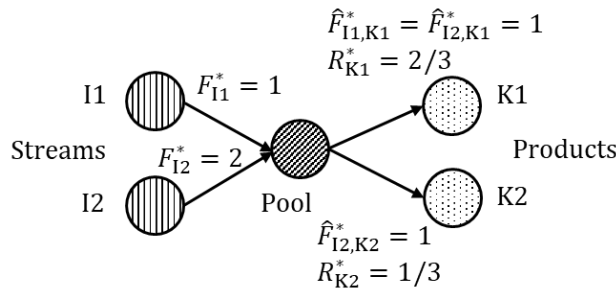


Figure 4-3. The optimal solution to the illustrative example from solving the first relaxation

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

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

In the next round, we solve M_1^{C-L} again after adding Eqn. (4-43). The optimal objective function value now becomes 20.78, which is closer to the objective function value obtained from solving the nonlinear model M^{CON} (which is 20.5). We show the feasible space for F_{I2} and R_{K2} intersects with the plane $\hat{F}_{I2,K2} = 1$ for this example, together with the constraint added, in Figure 4-4 where solid curve represents $F_{I2}R_{K2} = \hat{F}_{I2,K2}$ in M^{CON} intersects with $\hat{F}_{I2,K2} = 1$; dashed lines represent the intersection of Eqn. (4-39) - (4-41) in M_m^{CON-L} with $\hat{F}_{I2,K2} = 1$; point A represents (F_{I2}^*, R_{K2}^*) obtained from solving M_0^{CON-L} ; dot-dashed line represents Eqn. (4-43) intersects with $\hat{F}_{I2,K2} = 1$.

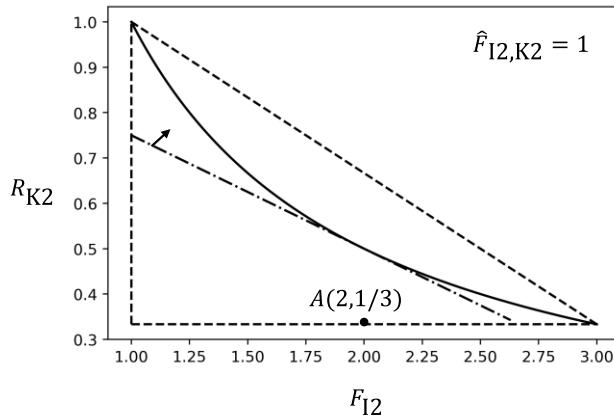


Figure 4-4. Tightening constraint for the illustrative example.

4.4.1.2 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^{CON} , except that the nonlinear constraint Eqn. (4-5) is replaced by:

$$\hat{F}_{ijk} \geq F_{ijn}^L R_{jk} + R_{jkn}^L F_{ij} - F_{ijn}^L R_{jkn}^L \quad (4-44)$$

$$\hat{F}_{ijk} \geq F_{ijn}^U R_{jk} + R_{jkn}^U F_{ij} - F_{ijn}^U R_{jkn}^U \quad (4-45)$$

$$\hat{F}_{ijk} \leq F_{ijn}^U R_{jk} + R_{jkn}^L F_{ij} - F_{ijn}^U R_{jkn}^L \quad (4-46)$$

$$\hat{F}_{ijk} \leq F_{ijn}^L R_{jk} + R_{jkn}^U F_{ij} - F_{ijn}^L R_{jkn}^U \quad (4-47)$$

where F_{ijn}^L/F_{ijn}^U and R_{jkn}^L/R_{jkn}^U are lower/upper bound on F_{ij} and R_{jk} for 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, $M_n^{\text{CON-L}}$ also contains the following constraint:

$$\rho_{jkn'}^2 \sum_{i \in \mathbf{I}_{jkn'}^*} F_{ij} + v_{jk}^U R_{jk} - 2\rho_{jkn'} \sum_{i \in \mathbf{I}_{jkn'}^*} \hat{F}_{ijk} \geq 0, \quad (j, k, n') \in \hat{\mathbf{C}}_n \quad (4-48)$$

where $\hat{\mathbf{C}}_n$ contains (j, k, n') combinations that lead to Eqn. (4-48) in all previous nodes, and subsets $\mathbf{I}_{jkn}^*$ is defined as follows: At each node, we solve $M_n^{\text{CON-L}}$ and, for each (j, k) pair, define $\mathbf{I}_{jkn}^* = \{i: \hat{F}_{ijk}^* > 0\}$.

We generate Eqn. (4-48) using Algorithm 2, with the optimal solution to $M_n^{\text{CON-L}}$ used as inputs. In Algorithm 2 we check if the nonlinear constraints in M^{CON} are satisfied; if not, we aim to generate constraints that cut off the current optimal solution to $M_n^{\text{CON-L}}$. For constraint generation, for each (j, k) pair we define the set $\mathbf{I}_{jkn}^*$ and check the sign of

$[4 \left(\sum_{i \in \mathbf{I}_{jkn}^*} \hat{F}_{ijk}^* \right)^2 - 4v_{jk}^U R_{jk}^* \sum_{i \in \mathbf{I}_{jkn}^*} F_{ij}^*]$; if positive, that means there exist a parameter ρ that

leads to a constraint in Eqn. (4-48) violated by the current solution to $M_n^{\text{CON-L}}$, and 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. (4-48) 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. (4-48) 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}^U 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** $4 \left(\sum_{i \in \mathbf{I}_{jkn}^*} \hat{F}_{ijk}^* \right)^2 - 4v_{jk}^U R_{jk}^* \sum_{i \in \mathbf{I}_{jkn}^*} F_{ij}^* > 0$ **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 4-5. We start with the solution to the relaxation at the

root node ($M_0^{\text{CON-L}}$) 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^{CON} are satisfied (i.e., $c_1 = \text{True}$), then such solution is a feasible solution to M^{CON} ; if new constraints are generated in Algorithm 2 (i.e., $c_2 = \text{True}$), then we add them to $M_n^{\text{CON-L}}$ and solve it again. Note that set $\hat{\mathbf{C}}_n$ for Eqn. (4-48) is updated in Algorithm 2, and constraints in Eqn. (4-48), once generated, will be included in all later nodes. After updating the solution to $M_n^{\text{CON-L}}$, we perform local search to find a feasible solution by solving M^{CON} using a local solver with the solution to $M_n^{\text{CON-L}}$ 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. (4-48)). The details for the implemented node selection rule, local search, prune rule, and branching strategy can be found in Appendix A2.2. 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. (4-48) 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.

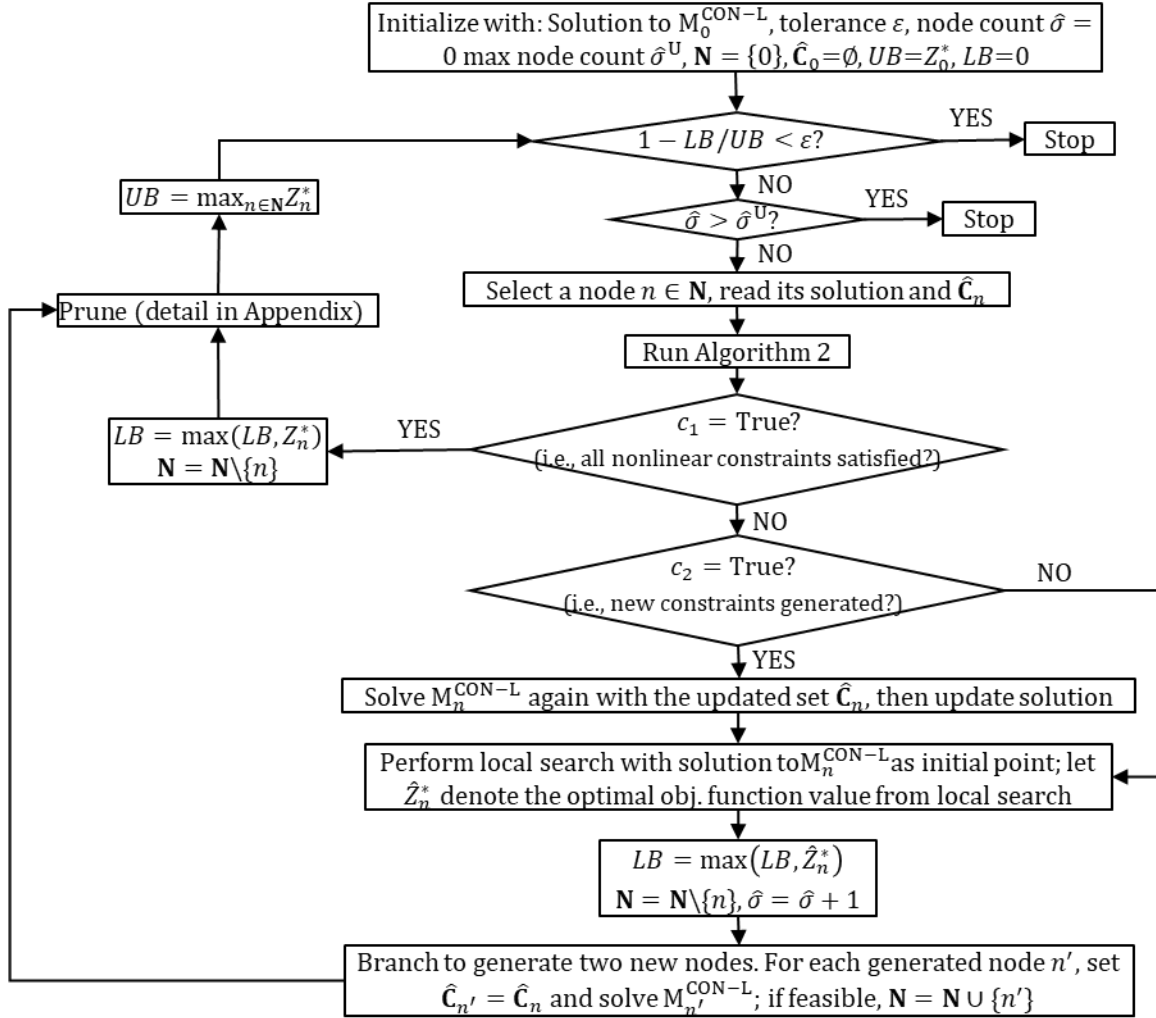


Figure 4-5. Flowchart of the customized B&C algorithm.

4.4.1.3 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:

$$\rho_{jko}^2 \tilde{F}_j + v_{jk}^U R_{jk} - 2\rho_{jko} \bar{F}_{jk} \geq 0, \quad j, k, o \in \mathbf{O} \quad (4-49)$$

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. (4-49) is generated without solving any optimization problem. Recall that

for a given (j, k) pair, when $\bar{F}_{jk} = v_{jk}^U$, Eqn. (4-49) is tangent to $\tilde{F}_j R_{jk} = v_{jk}^U$ at the point corresponding to $\tilde{F}_j = v_{jk}^U / \rho, R_{jk} = \rho$. Note that when $\tilde{F}_j R_{jk} = v_{jk}^U$, we have $R_{jk} \in [v_{jk}^U / \gamma_j, 1]$, thus one straightforward way to define ρ_{jko} is the following:

$$\rho_{jko} = v_{jk}^U / \gamma_j + o(1 - v_{jk}^U / \gamma_j) / |\mathbf{O}|, \quad j, k \quad (4-50)$$

Eqn. (4-50) generates a set of ρ_{jko} whose values are evenly distributed in $[v_{jk}^U / \gamma_j, 1]$. We show an illustrative graph for the points of tangency on bilinear curve generated from such ρ_{jko} in Figure 4-6.

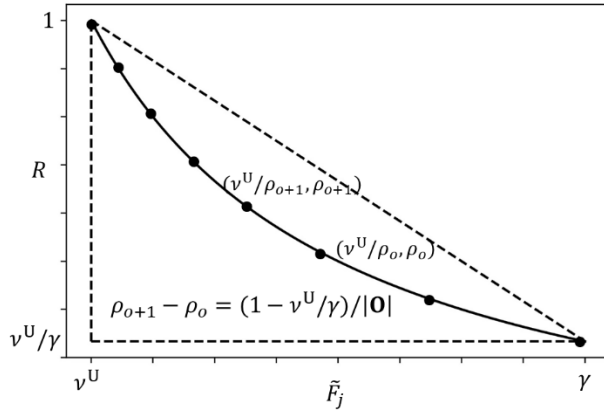


Figure 4-6. Illustrative graph for points of tangency on bilinear curve generated from Eqn. (4-50) with seven intervals (indices j and k are dropped in the graph for simplicity).

4.4.2 Methods for model with semi-continuous variables

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

$$\rho_{jkm'}^2 \tilde{F}_j + v_{jk}^U R_{jk} + 2\rho_{jkm'} (\sigma_{jk,1} \bar{F}_{jk} + \sigma_{jk,2} Z_{jk}) \geq 0, \quad (j, k, m') \in \mathbf{C}_m \quad (4-51)$$

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

We consider model M_m^{SC-L} which contains all constraints in model M^{SC} , except that the nonlinear constraint Eqn. (4-5) is replaced by Eqn. (4-39) - (4-41). M_m^{SC-L} also contains Eqn. (4-51).

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 $[4(\sigma_{jk,1}\bar{F}_{jk}^* + \sigma_{jk,2}Z_{jk}^*)^2 - 4v_{jk}^U \tilde{F}_j^* R_{jk}^*]$; if positive, we calculate parameter $\rho_{jkm} = -(\sigma_{jk,1}\bar{F}_{jk}^* + \sigma_{jk,2}Z_{jk}^*)/\tilde{F}_j^*$ and generate Eqn. (4-51). 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: $c = \text{True}$, $m = 0$, σ^U , and $C_m = \emptyset$

While $c = \text{True}$ **AND** $m < \sigma^U$ **do**

$c = \text{False}$

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

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

$C_{m+1} = C_m$

For $j \in J$ **do**

For $k \in K$ **do**

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

$C_{m+1} = C_{m+1} \cup \{(j, k, m)\}$

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

$c = \text{True}$

End

End

End

$m = m + 1$

End

Outputs: ρ_{jkm} and C_{m+1}

Similarly, Algorithm 4 generates constraints by solving the continuous relaxation of M_m^{SC-L} iteratively but based on solving the minimum distance problem. After solving the continuous

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

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

$$(2) \text{ If } R_{jk}^* > \sqrt{v_{jk}^U/\gamma_j}, \text{ then } \rho_{jkm} = R_{jk}^*.$$

$$(3) \text{ If } \sqrt{v_{jk}^U/\gamma_j} > v_{jk}^U/\tilde{F}_j^*, \text{ then } \rho_{jkm} = v_{jk}^U/\tilde{F}_j^*.$$

Algorithm 4. Constraint generation from minimizing distance

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

While $c = \text{True}$ **AND** $m < \sigma^U$ **do**

$c = \text{False}$

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

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

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

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

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

Else

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

End

End

End

End

$m = m + 1$

End

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

We can also generate the proposed constraint for M^{SC} using predefined parameters.

Specifically, we have:

$$\rho_{jko}^2 \tilde{F}_j + v_{jk}^U R_{jk} + 2\rho_{jko} (\sigma_{jk,1} \bar{F}_{jk} + \sigma_{jk,2} Z_{jk}) \geq 0, \quad j, k, o \in \mathbf{O} \quad (4-52)$$

with ρ_{jko} calculated from Eqn. (4-50).

4.5 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. 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 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).

4.5.1 Models with only continuous variables

4.5.1.1 Model M^{CON}

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

- 1) M_R^{CON} : model M^{CON} with Eqn. (4-42) generated iteratively at the root node; $\mathbf{C}_m$ in Eqn. (4-42) is obtained by running Algorithm 1 with $\sigma^U = 10$.
- 2) $M_{\text{B\&C}}^{\text{CON}}$: model M^{CON} with Eqn. (4-48) generated using the B\&C algorithm; $\hat{\mathbf{C}}_n$ in Eqn. (4-48) is obtained by running the algorithm shown in Figure 4-5 with $\hat{\sigma}^U = 10$.
- 3) M_H^{CON} : model M^{CON} with Eqn. (4-42) and Eqn. (4-48) generated from a hybrid approach; we first obtain $\mathbf{C}_m$ generating Eqn. (4-42) by running Algorithm 1 with $\sigma^U = 5$ and then obtain $\hat{\mathbf{C}}_n$ for generating Eqn. (4-48) by running the B\&C algorithm with $\hat{\sigma}^U = 5$ (all $M_n^{\text{CON-L}}$ include previously generated Eqn. (4-42)).

- 4) M_P^{CON} : M^{CON} with Eqn. (4-49) expressed for predefined parameter ρ_{jko} generated from Eqn. (4-50) with $|\mathbf{O}| = 10$.

We show the performance profiles for BARON 19.12.7 and SCIP 6.0 in Figure 4-7. Both profiles are generated over 60 instances, and for each profile we exclude instances that can be solved by all models within 10 seconds with the corresponding solver. We note that certain proposed methods, notably M_R^{CON} , bring computational improvements to both solvers. As a side note, the two solvers have similar performance for solving M^{CON} for the 60 instances mentioned above.

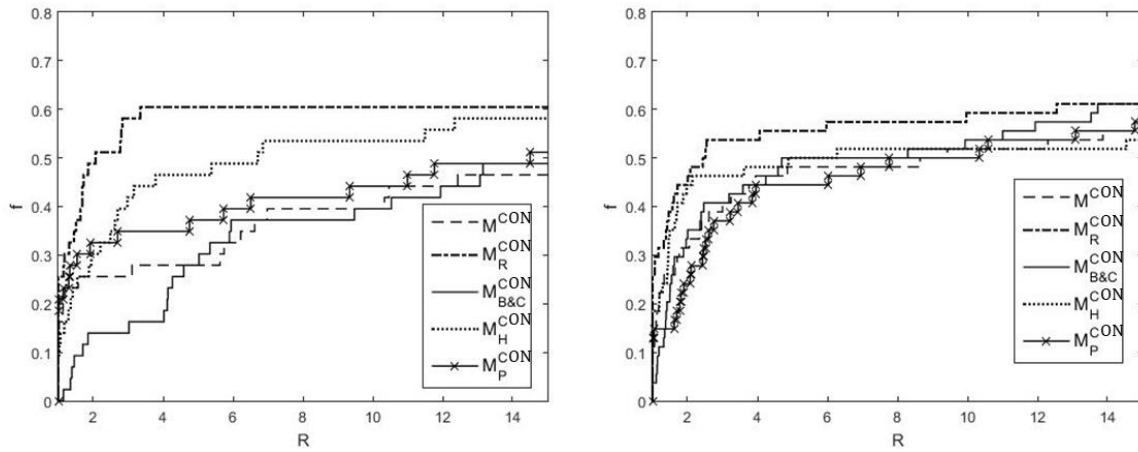


Figure 4-7. Performance profile for model with only continuous variable and its variants solved with BARON (left) and SCIP (right)

There are 21 instances that are not solved by BARON in 300 seconds with the original model M^{CON} . We solve the first seven instances within those with the B&C algorithm shown in Figure 4-5 with $\hat{\sigma}^U = 100$. 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 A2.3). Table 4-2 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 4-2. Solution statistics for B&B and B&C algorithms over select instances

Inst.	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.

4.5.1.2 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:

$$\hat{F}_{ijk} = q_{ij}\bar{F}_{jk}, \quad i, j, k \quad (4-53)$$

Summing over index k for Eqn. (4-53), we obtain:

$$\sum_k \hat{F}_{ijk} = q_{ij} \sum_k \bar{F}_{jk}, \quad i, j, k \quad (4-54)$$

Note that the LHS of Eqn. (4-54) 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. (4-54) 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.(4-42), 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. (4-49) generated using pre-determined parameter ρ values calculated from an equation similar to Eqn. (4-50) with $|\mathbf{O}| = 10$.

We show a performance profile containing 33 instances in Figure 4-8. 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.

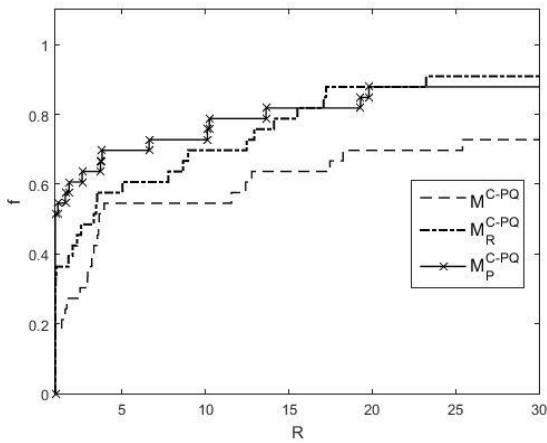


Figure 4-8. Performance profile for model based on pq-formulation and its variants solved with BARON with maximizing profit objective.

4.5.2 Model with semi-continuous variables

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

- 1) M_V^{SC} : model M^{SC} with Eqn. (4-51) using Algorithm 3 with $\sigma^U = 10$.
- 2) M_D^{SC} : model M^{SC} with Eqn. (4-51) using Algorithm 4 with $\sigma^U = 10$.
- 3) M_P^{SC} : M^{SC} with Eqn. (4-52) expressed for predefined parameter ρ_{jko} generated from Eqn. (4-50) with $|\mathbf{O}| = 10$.

We show performance profile for another set of 30 instances in Figure 4-9 with maximizing profit objective. Models are solved with BARON. Similarly, M_V^{SC} and M_D^{SC} typically contain additional constraints in the order of hundreds generated in around 10 seconds. Overall, our methods improve the performance of BARON.

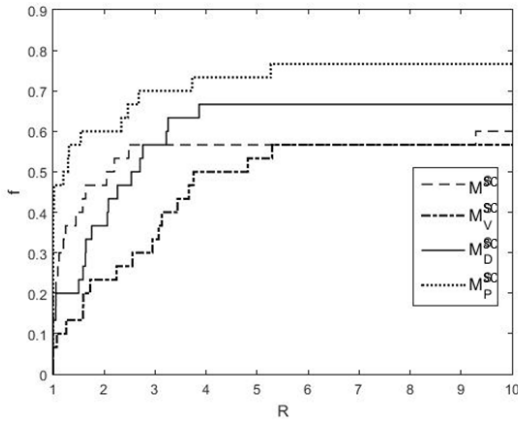


Figure 4-9. Performance profile for model with semi-continuous variables and its variants solved with BARON with maximizing profit objective.

We also test our methods for M^{SC} using an objective where we minimize cost and penalty for unmet demand, defined in (4-15). We show the performance profile in Figure 4-10 over

another set of 30 instances. Similarly, our methods reduce the computational requirement and optimality gap.

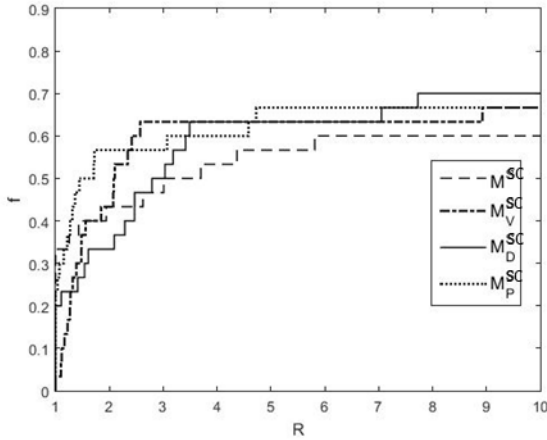


Figure 4-10. Performance profile for model with semi-continuous variables and its variants solved with BARON with minimizing cost objective.

4.6 Conclusion

We derived a family of strong valid 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 maximizing constraint violation and solving the minimum distance problem. We tested the generated constraints on the pooling problem. Computational results demonstrate the effectiveness of our methods in terms of reducing the optimality gap and computational time.

Chapter 5

Summary

In this thesis we presented solution methods for pooling and multiperiod blending problems. We first consider the multiperiod blending problem with minimizing cost objective. We develop a novel preprocessing algorithm to calculate lower bounds on stream flows. We define product dedicated flow variables to address product specific features involved in multiperiod blending problem. The bounds on stream flows and new product dedicated flow variables are then used to generate tightening constraints which significantly improve the solution time of the mixed-integer nonlinear programming models as well as models based on linear approximations.

For multiperiod blending problem with maximizing profit objective, we first propose a reformulation of the constraints involving bilinear terms using lifting. We introduce a procedure to calculate tight bounds on the lifted variables calculated by aggregating multiple constraints. We propose valid constraints derived from Reformulation-Linearization Technique that utilize the bounds on the lifted variables to further tighten the formulation. Computational results indicate our method can substantially reduce the solution time and optimality gap.

Finally, we develop tightening and solution methods based on nontrivial bounds on bilinear terms. We derive a family of valid linear constraints and further show that, when one of the nontrivial bounds is active, such constraints are tangent to one branch of hyperbola that represents the bilinear term. We propose different preprocessing methods for generating

strong constraints from the family and test them on the pooling problem. Computational results demonstrate the effectiveness of our methods in terms of reducing optimality gap and computational time.

Future research directions on related topics include: (1) Strong valid constraints for the multiperiod blending problem that exploit the combinatorial structure in it. (2). Implementation of the variable bound tightening methods introduced in Chapter 3 in B&B algorithm, with automatic detection for the specific structure we study in general problem. (3). Implementation of the valid constraints for bilinear terms with nontrivial bounds in B&C algorithm that can be applied to general nonlinear program with such bounds.

Appendix

A1 Explanations to Chapter 3

A1.1 Solving LP3

After introducing slack variables S_1, S_2 , and S_3 , LP3 is written as follows:

$$\begin{aligned}
 \min \quad & \sum_{i' \neq i} \mu_{i'l} V_{i'} \\
 \text{s. t.} \quad & \sum_{i' \neq i} V_{i'} + S_1 = \gamma - \hat{\gamma}_{il} \\
 & - \sum_{i' \neq i} V_{i'} + S_2 = \hat{\gamma}_{il} - \gamma \\
 & V_{b(l)} + S_3 = \bar{\gamma}_{b(l)} \\
 & V_{i'} \geq 0, S_1, S_2, S_3 \geq 0
 \end{aligned}$$

By inspection, we have $V_{b^+(l)} = \gamma - \hat{\gamma}_{il}$, $V_{i'} = 0 \forall i' \notin \{i, b^+(l)\}$, $S_1 = S_2 = 0$, and $S_3 = \bar{\gamma}_{b(l)}$ as initial feasible solution. Let $S_1, V_{b^+(l)}$, and S_3 be basic variables, we have the following tableau:

Basic var.	$V_{b(l)}$	$V_{b^+(l)}$	$[V_{i'}, \forall i' \notin \{i, b(l), b^+(l)\}]$	S_1	S_2	S_3	
S_1	0	0	$[0, \dots, 0]$	1	1	0	0
$V_{b^+(l)}$	1	1	$[1, \dots, 1]$	0	-1	0	$\gamma - \hat{\gamma}_{il}$
S_3	1	0	$[0, \dots, 0]$	0	0	1	$\bar{\gamma}_{b(l)}$
z	$\mu_l^+ - \mu_l^*$	0	$[\mu_l^+ - \mu_{i'l}, \forall i' \notin \{i, b(l), b^+(l)\}]$	0	μ_l^+	0	$\mu_l^+ (\hat{\gamma}_{il} - \gamma)$

where $[.]$ denotes a row vector of dimension $(|\mathbf{I}| - 3)$.

When $\mu_l^+ \leq 0$, we have the following optimal tableau:

Basic var.	$V_{b(l)}$	$V_{b^+(l)}$	$[V_{i'}, \forall i' \notin \{i, b(l), b^+(l)\}]$	S_1	S_2	S_3	
S_1	0	0	$[0, \dots, 0]$	1	1	0	0
$V_{b^+(l)}$	0	1	$[1, \dots, 1]$	0	-1	0	$\gamma - \hat{\gamma}_{il} - \bar{\gamma}_{b(l)}$
$V_{b(l)}$	1	0	$[0, \dots, 0]$	0	0	1	$\bar{\gamma}_{b(l)}$
z	0	0	$[\mu_l^+ - \mu_{i'l}, \forall i' \notin \{i, b(l), b^+(l)\}]$	0	μ_l^+	$\mu_l^* - \mu_l^+$	$\mu_l^+ (\hat{\gamma}_{il} - \gamma) + (\mu_l^* - \mu_l^+) \bar{\gamma}_{b(l)}$

When $\mu_l^+ > 0$, we have the following optimal tableau:

Basic var.	$V_{b(l)}$	$V_{b^+(l)}$	$[V_{i'}, \forall i' \notin \{i, b(l), b^+(l)\}]$	S_1	S_2	S_3	
S_1	0	0	$[0, \dots, 0]$	1	1	0	0
$V_{b^+(l)}$	1	1	$[1, \dots, 1]$	0	-1	0	$\gamma - \hat{\gamma}_{il} - \bar{\gamma}_{b(l)}$
$V_{b(l)}$	1	0	$[0, \dots, 0]$	0	0	1	$\bar{\gamma}_{b(l)}$
z	0	0	$[\mu_l^+ - \mu_{i'l}, \forall i' \notin \{i, b(l), b^+(l)\}]$	$-\mu_l^+$	0	$\mu_l^* - \mu_l^+$	$\mu_l^+ (\hat{\gamma}_{il} - \gamma) + (\mu_l^* - \mu_l^+) \bar{\gamma}_{b(l)}$

A1.2 Illustrative example

A1.2.1. Feasibility Based Bound Tightening

Recall that for the illustrative example we have:

$$V_1 + V_2 + V_3 \leq 1$$

$$-V_1 + 2V_2 + V_3 \leq 0$$

$$V_1 - 3V_2 + 2V_3 \leq 0$$

Assume we use 0 and 1 as the initial lower and upper bound, that is, $V_1, V_2, V_3 \in [0, 1]$. FBBT uses the following inequality to find tighter upper bounds (note that 0 is the tightest lower bound on V_i):

$$V_i \leq \frac{1}{\alpha_{m^*, i}} \left[\beta_{m^*} - \sum_{i' \neq i} \min(a_{m^*, i'} \bar{\gamma}_{i'}, 0) \right] \quad a_{m^*, i} > 0 \quad (\text{A-1})$$

where $\alpha_{m^*, i}$ is the coefficient of V_i for inequality m^* , β_{m^*} is the RHS of inequality m^* , and $\bar{\gamma}_i$ is the upper bound on V_i . In FBBT we choose an inequality with positive coefficient for V_i , to evaluate the RHS of Eqn. (A-1) to find its upper bound:

$$V_1 \leq \frac{1}{1} [1 - \min(1,0) - \min(1,0)] = 1$$

$$V_1 \leq \frac{1}{1} [0 - \min(-3,0) - \min(2,0)] = 3$$

$$V_2 \leq \frac{1}{1} [1 - \min(1,0) - \min(1,0)] = 1$$

$$V_2 \leq \frac{1}{2} [0 - \min(-1,0) - \min(1,0)] = 1/2$$

Note that we now have a tighter upper bound on V_2 , so we update $\bar{\gamma}_2$: $\bar{\gamma}_2 = 1/2$.

$$V_3 \leq \frac{1}{1} [1 - \min(1,0) - \min(1/2,0)] = 1$$

$$V_3 \leq \frac{1}{1} [0 - \min(-1,0) - \min(1,0)] = 1$$

$$V_3 \leq \frac{1}{2} [0 - \min(1,0) - \min(-3/2,0)] = 3/4$$

Note that we now have a tighter upper bound on V_3 , so we update $\bar{\gamma}_3$: $\bar{\gamma}_3 = 3/4$.

In FBBT we typically start another round of evaluation using the tightened bounds. For the illustrative example, no further improvement can be obtained. FBBT thus returns: $\bar{\gamma}_1 = 1, \bar{\gamma}_2 = 1/2, \bar{\gamma}_3 = 3/4$.

A1.2.2. OBBT for the illustrative example

OBBT is based on the solution of the following LP:

$$\begin{array}{ll} \max & V_i \ (i = 1,2,3) \\ & V_1 + V_2 + V_3 \leq 1 \\ \text{s. t} & -V_1 + 2V_2 + V_3 \leq 0 \\ & V_1 - 3V_2 + 2V_3 \leq 0 \end{array}$$

The value of $\bar{\gamma}_i$ is equal to the objective function value of the i -th LP. After solving three LPs, OBBT returns: $\bar{\gamma}_1 = 3/4, \bar{\gamma}_2 = 1/3, \bar{\gamma}_3 = 1/11$.

A1.2.3. Illustrative graph for our tightening methods

Consider the following nonlinear set:

$$\mathbf{S}_1 = \left\{ (\hat{F}_1, \hat{F}_2, \hat{F}_3, R, V_1, V_2, V_3) \in \mathbb{R}^+ : \begin{array}{l} V_1 + V_2 + V_3 \leq 1 \\ -V_1 + 2V_2 + V_3 \leq 0 \\ V_1 - 3V_2 + 2V_3 \leq 0 \\ \hat{F}_1 = V_1 R \\ \hat{F}_2 = V_2 R \\ \hat{F}_3 = V_3 R \end{array} \right\}$$

which contains three linear constraints that are identical to the constraints in the illustrative example in section 3, along with three nonlinear equality constraints to model the flows.

We introduce a hyperplane:

$$\mathbf{S}_2 = \left\{ (R, V_1, V_2, V_3) \in \mathbb{R}^+ : \begin{array}{l} R = 1/2 \\ V_1 = 2/3 \\ V_2 = 1/3 \\ V_3 = 0 \end{array} \right\}$$

The intersection of $\mathbf{S}_1$ and $\mathbf{S}_2$ is shown in Figure A1-1. It is point A on the $(\hat{F}_1, \hat{F}_2)$ plane.

We consider a linear relaxation of $\mathbf{S}_1$, denoted as $\mathbf{S}_1^{\text{MC}}$, using McCormick envelopes without bound tightening. Since $V_1, V_2, V_3 \in [0,1]$, we have:

$$\hat{F}_i \leq R, \quad i = \{1,2,3\} \tag{A-2}$$

$$\hat{F}_i \leq V_i, \quad i = \{1,2,3\} \tag{A-3}$$

$$\hat{F}_i \geq R + V_i - 1, \quad i = \{1,2,3\} \tag{A-4}$$

$$\hat{F}_i \geq 0, \quad i = \{1,2,3\} \tag{A-5}$$

We also have the following RLT constraints:

$$\hat{F}_1 + \hat{F}_2 + \hat{F}_3 \leq R \quad (\text{A-6})$$

$$-\hat{F}_1 + 2\hat{F}_2 + \hat{F}_3 \leq 0 \quad (\text{A-7})$$

$$\hat{F}_1 - 3\hat{F}_2 + 2\hat{F}_3 \leq 0 \quad (\text{A-8})$$

The set $\mathbf{S}_1^{\text{MC}}$ is thus defined as:

$$\mathbf{S}_1^{\text{MC}} = \left\{ (\hat{F}_1, \hat{F}_2, \hat{F}_3, R, V_1, V_2, V_3) \in \mathbb{R}^+ : \begin{array}{l} V_1 + V_2 + V_3 \leq 1 \\ -V_1 + 2V_2 + V_3 \leq 0 \\ V_1 - 3V_2 + 2V_3 \leq 0 \\ \text{Eqns. (A-2) - (A-5)} \\ \text{Eqns. (A-6) - (A-8)} \end{array} \right\}$$

The intersection of $\mathbf{S}_1^{\text{MC}}$ and $\mathbf{S}_2$ is the quadrilateral $ABCD$.

We consider a linear relaxation of $\mathbf{S}_1$, denoted as $\mathbf{S}_1^{\text{T}}$, using McCormick envelopes with tightened bounds. Our methods lead to: $V_1 \in [0, 3/4]$, $V_2 \in [0, 1/3]$, $V_3 \in [0, 1/3]$. McCormick envelopes constructed using such bounds are:

$$\hat{F}_1 \leq \frac{3}{4}R \quad (\text{A-9})$$

$$\hat{F}_2 \leq \frac{1}{3}R \quad (\text{A-10})$$

$$\hat{F}_3 \leq \frac{1}{3}R \quad (\text{A-11})$$

$$\hat{F}_1 \geq \frac{3}{4}R + V_1 - \frac{3}{4} \quad (\text{A-12})$$

$$\hat{F}_2 \geq \frac{1}{3}R + V_2 - \frac{1}{3} \quad (\text{A-13})$$

$$\hat{F}_3 \geq \frac{1}{3}R + V_3 - \frac{1}{3} \quad (\text{A-14})$$

together with Eqn. (A-3) and Eqn.(A-5). Note that Eqn. (A-12)– (A-14) are identical to Eqn. (3-27) for the illustrative example.

The set $\mathbf{S}_1^T$ is thus defined as:

$$\mathbf{S}_1^T = \left\{ (\hat{F}_1, \hat{F}_2, \hat{F}_3, R, V_1, V_2, V_3) \in \mathbb{R}^+ : \begin{array}{l} V_1 + V_2 + V_3 \leq 1 \\ -V_1 + 2V_2 + V_3 \leq 0 \\ V_1 - 3V_2 + 2V_3 \leq 0 \\ \text{Eqns. (A-3), (A-5), (A-9) – (A-14)} \\ \text{Eqns. (A-6) – (A-8)} \end{array} \right\}$$

The intersection of $\mathbf{S}_1^T$ and $\mathbf{S}_2$ is also point A, which coincides with the intersection of the nonlinear set $\mathbf{S}_1$ and $\mathbf{S}_2$.

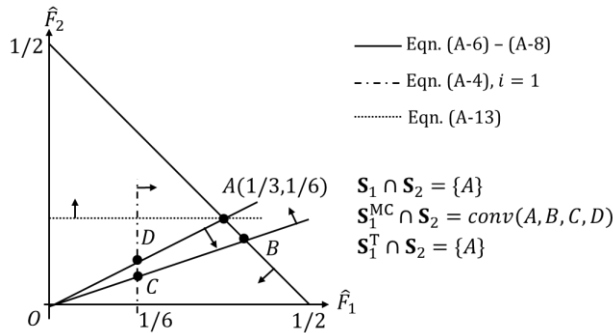


Figure A1-1. Illustrative graph for tightening constraints

A2 Explanations to Chapter 4

A2.1 Solving the minimum distance problem

Consider the following optimization problem:

$$\min_{\rho} \left\{ \frac{1}{\bar{x} - \underline{x}} \left| \frac{\bar{w}}{\rho} - x^* \right| + \frac{1}{\bar{y} - \underline{y}} |\rho - y^*| : (\bar{w}/\bar{x}) \leq \rho \leq \bar{y} \right\} \quad (4-36)$$

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

(1) If $y^* \leq \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} \leq \bar{w}/x^*$, then $\rho = \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})}$.

(2) If $\sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} < y^*$, then $\rho = y^*$.

(3) If $\sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} > \bar{w}/x^*$, then $\rho = \bar{w}/x^*$.

Proof We discuss the above three cases separately.

(1) When $y^* \leq \rho = \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} \leq \bar{w}/x^*$, it follows that $\rho - y^* \geq 0$ and $\bar{w}/\rho \geq \bar{w}/(\bar{w}/x^*) \geq x^*$, thus the optimization problem in (4-36) is equivalent to:

$$\min_{\rho} \left\{ \frac{1}{\bar{x} - \underline{x}} \left(\frac{\bar{w}}{\rho} - x^* \right) + \frac{1}{\bar{y} - \underline{y}} (\rho - y^*) : (\bar{w}/\bar{x}) \leq \rho \leq \bar{y} \right\}$$

Dropping the constant terms, we have:

$$\min_{\rho} \left\{ \frac{1}{\bar{x} - \underline{x}} \cdot \frac{\bar{w}}{\rho} + \frac{1}{\bar{y} - \underline{y}} \cdot \rho : (\bar{w}/\bar{x}) \leq \rho \leq \bar{y} \right\} \quad (\text{A-15})$$

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

$$\frac{1}{\bar{x} - \underline{x}} \cdot \frac{\bar{w}}{\rho} + \frac{1}{\bar{y} - \underline{y}} \cdot \rho \geq 2 \sqrt{\frac{\bar{w}}{(\bar{x} - \underline{x})(\bar{y} - \underline{y})}} \quad (\text{A-16})$$

The equal sign in (A-16) holds when $\rho = \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})}$ and by construction such ρ is in the range of $[\bar{w}/\bar{x}, \bar{y}]$ (since $\bar{w}/x^U \leq y^* \leq \rho$, and $\rho \leq \bar{w}/x^* \leq \bar{w}/(\bar{w}/\bar{y}) \leq \bar{y}$). Thus, $\rho = \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})}$ is the solution to (4-36) when $y^* \leq \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} \leq \bar{w}/x^*$.

(2) We first assume the optimal solution $\rho < y^*$. If that is the case, we also have $\bar{w}/\rho > \bar{w}/y^* > x^*$ since $x^*y^* < \bar{w}$. It follows that such ρ is not an optimal solution to (4-36) since there exists $\varepsilon > 0$ such that $(\rho + \varepsilon)$ leads to smaller value for both $|\bar{w}/\rho - x^*|$ and $|\rho - y^*|$.

We next assume the optimal solution $\rho > y^*$. If that is the case, we first note that $\bar{w}/\rho > x^*$ should hold since otherwise there exists $\varepsilon > 0$ such that $(\rho - \varepsilon)$ leads to the objective function value $\frac{1}{\bar{x} - \underline{x}} \left(x^* - \frac{\bar{w}}{\rho - \varepsilon} \right) + \frac{1}{\bar{x} - \underline{x}} (\rho - y^* - \varepsilon) < \frac{1}{\bar{x} - \underline{x}} \left(x^* - \frac{\bar{w}}{\rho} \right) + \frac{1}{\bar{x} - \underline{x}} (\rho - y^*)$. Now, since $\rho > y^*$ and $\bar{w}/\rho > x^*$, we again have the optimization problem defined in (A-15) with $\rho = \sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} < y^*$, which contradicts with $\rho > y^*$. Thus, the optimal solution can only be $\rho = y^*$.

(3) We first assume the optimal solution $\rho > \bar{w}/x^*$. If that is the case, we also have $\rho > y^*$ since $x^*y^* < \bar{w}$. It follows that such ρ is not an optimal solution to (4-36) since there exists $\varepsilon > 0$ such that $(\rho + \varepsilon)$ leads to smaller value for both $|\bar{w}/\rho - x^*|$ and $|\rho - y^*|$.

We next assume the optimal solution $\rho < \bar{w}/x^*$. 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 $\frac{1}{\bar{x} - \underline{x}} \left(\frac{\bar{w}}{\rho + \varepsilon} - x^* \right) + \frac{1}{\bar{x} - \underline{x}} (y^* - \rho - \varepsilon) < \frac{1}{\bar{x} - \underline{x}} \left(\frac{\bar{w}}{\rho} - x^* \right) + \frac{1}{\bar{x} - \underline{x}} (y^* - \rho)$. Now, since $\rho < \bar{w}/x^*$ and $\rho > y^*$, we again have the optimization problem defined in (A-15) with $\rho =$

$\sqrt{\bar{w}(\bar{y} - \underline{y})/(\bar{x} - \underline{x})} > \bar{w}/x^*$, which contradicts with $\rho < \bar{w}/x^*$. Thus, the optimal solution can only be $\rho = \bar{w}/x^*$. ■

A2.2 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^{CON} using CONOPT, with the initial point being the solution to $M_n^{\text{C-L}}$.

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^{CON} , 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: $(i, j, k) = \arg \max_{i'j'k'} |\hat{F}_{i'j'k'}^* - F_{i'j'k'}^*|$, where $\hat{F}_{i'j'k'}^*$, $F_{i'j'k'}^*$, and $R_{j'k'}^*$ are obtained from solving $M_n^{\text{CON-L}}$. Once the specific (i, j, k) is identified, we evaluate the following equation: $\hat{\delta}_{jk} = \left| R_{jk}^* - \frac{R_{jkn}^U - R_{jkn}^L}{2} \right|$. Parameter $\hat{\delta}_{jk}$ aims to quantify the distance between R_{jk}^* and the midpoint for its range. We branch on R_{jk} corresponds to the smallest $\hat{\delta}_{jk}$. For branching, the break point is at the variable value in the solution to $M_n^{\text{CON-L}}$; 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.

A2.3 B&B algorithm

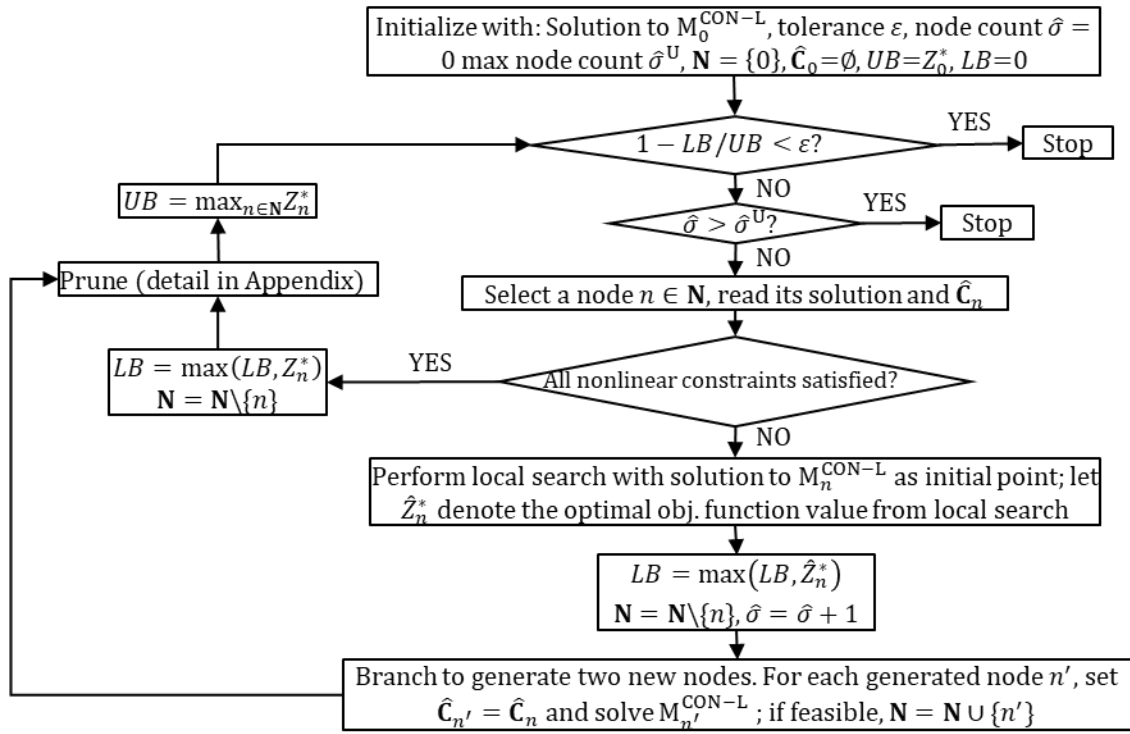


Figure A2-1. Flowchart for a customized B&B algorithm.

Bibliography

- Achterberg, Tobias. 2007. "Constraint Integer Programming." Technische Universität Berlin, Fakultät II - Mathematik und Naturwissenschaften. <https://depositonce.tu-berlin.de/handle/11303/1931>.
- Achterberg, Tobias, Robert E. Bixby, Zonghao Gu, Edward Rothberg, and Dieter Weninger. 2020. "Presolve Reductions in Mixed Integer Programming." *INFORMS Journal on Computing* 32 (2): 473–506. <https://doi.org/10.1287/ijoc.2018.0857>.
- Adhya, Nilanjan, Mohit Tawarmalani, and Nikolaos V. Sahinidis. 1999. "A Lagrangian Approach to the Pooling Problem." *Industrial and Engineering Chemistry Research* 38 (5): 1956–72. <https://doi.org/10.1021/ie980666q>.
- 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).
- Baker, Thomas E., and Leon S. Lasdon. 1985. "Successive Linear Programming at Exxon." *Management Science* 31 (3): 264–74. <https://doi.org/10.1287/mnsc.31.3.264>.
- Baltea-Lugojan, Radu, and Ruth Misener. 2017. "Piecewise Parametric Structure in the Pooling Problem: From Sparse Strongly-Polynomial Solutions to NP-Hardness." *Journal of Global Optimization*, October, 1–36. <https://doi.org/10.1007/s10898-017-0577-y>.
- Belotti, Pietro. 2013. "Bound Reduction Using Pairs of Linear Inequalities." *Journal of Global Optimization* 56 (3): 787–819. <https://doi.org/10.1007/s10898-012-9848-9>.
- Belotti, Pietro, Jon Lee, Leo Liberti, François Margot, and Andreas Wächter. 2009. "Branching and Bounds Tightening techniques for Non-Convex MINLP." *Optimization Methods and Software* 24 (4–5): 597–634. <https://doi.org/10.1080/10556780903087124>.
- 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.
- Burkard, R.E., and J. Hatzl. 2005. "Review, Extensions and Computational Comparison of MILP Formulations for Scheduling of Batch Processes." *Computers & Chemical Engineering* 29 (8): 1752–69. <https://doi.org/10.1016/J.COMPCHEMENG.2005.02.037>.
- Castillo, Pedro A Castillo, and Vladimir Mahalec. 2014a. "Inventory Pinch Based, Multiscale Models for Integrated Planning and Scheduling-Part I: Gasoline Blend Planning." *AIChE Journal*. <https://doi.org/10.1002/aic.14423>.
- . 2014b. "Inventory Pinch Based, Multiscale Models for Integrated Planning and Scheduling-Part II: Gasoline Blend Scheduling." *AIChE Journal*. <https://doi.org/10.1002/aic.14444>.
- Castillo, Pedro A Castillo, Vladimir Mahalec, and Jeffrey D. Kelly. 2013. "Inventory Pinch Algorithm for Gasoline Blend Planning." *AIChE Journal* 59 (10): 3748–66. <https://doi.org/10.1002/aic.14113>.
- Castro, Pedro M. 2015a. "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>.
- . 2015b. "New MINLP Formulation for the Multiperiod Pooling Problem." *AIChE Journal* 61 (11): 3728–38. <https://doi.org/10.1002/aic.15018>.
- Ceccon, Francesco, Georgia Kouyialis, and Ruth Misener. 2016. "Using Functional Programming to Recognize Named Structure in an Optimization Problem: Application to Pooling." *AIChE Journal* 62 (9): 3085–95. <https://doi.org/10.1002/aic.15308>.
- Cerdá, Jaime, Pedro C. Pautasso, and Diego C. Cafaro. 2016. "A Cost-Effective Model for the Gasoline Blend Optimization Problem." *AIChE Journal* 62 (9): 3002–19. <https://doi.org/10.1002/aic.15208>.

- 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>.
- . 2021a. "Tightening Methods Based on Nontrivial Bounds on Bilinear Terms." *Manuscript under Review*.
- . 2021b. "Variable Bound Tightening and Valid Constraints for Multiperiod Blending." *Manuscript under Revision*.
- 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.
- DeWitt, Calvin W., Leon S. Lasdon, Allan D. Waren, Donald A. Brenner, and Simon A. Melhem. 1989. "OMEGA: An Improved Gasoline Blending System for Texaco." *Interfaces. INFORMS*. <https://doi.org/10.2307/25061187>.
- 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>.
- Domes, Ferenc, and Arnold Neumaier. 2016. "Constraint Aggregation for Rigorous Global Optimization." *Mathematical Programming* 155 (1–2): 375–401. <https://doi.org/10.1007/s10107-014-0851-4>.
- 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>.
- Gleixner, Ambros M., Timo Berthold, Benjamin Müller, and Stefan Weltge. 2017. "Three Enhancements for Optimization-Based Bound Tightening." *Journal of Global Optimization* 67 (4): 731–57. <https://doi.org/10.1007/s10898-016-0450-4>.
- 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>.
- Greenberg, Harvey J. 1995. "Analyzing the Pooling Problem." *ORSA Journal on Computing* 7 (2): 205–17. <https://doi.org/10.1287/ijoc.7.2.205>.
- Gupte, Akshay, Shabbir Ahmed, Myun Seok Cheon, and Santanu Dey. 2013. "Solving Mixed Integer Bilinear Problems Using MILP Formulations." *SIAM Journal on Optimization* 23 (2): 721–44. <https://doi.org/10.1137/110836183>.
- 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>.
- Janak, Stacy L., and Christodoulos A. Floudas. 2008. "Improving Unit-Specific Event Based Continuous-Time Approaches for Batch Processes: Integrality Gap and Task Splitting." *Computers & Chemical Engineering* 32 (4–5): 913–55. <https://doi.org/10.1016/J.COMPCHENG.2007.03.019>.
- 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>.
- Kelly, J. D., and J. L. Mann. 2003. "Crude Oil Blend Scheduling Optimization: An Application with Multimillion Dollar Benefits - Part 2." *Hydrocarbon Processing*.
- Kelly, Jeffrey D., Brenno C. Menezes, and Ignacio E. Grossmann. 2018. "Successive LP Approximation for Nonconvex Blending in MILP Scheduling Optimization Using Factors for Qualities in the Process Industry." *Industrial & Engineering Chemistry Research* 57 (32): 11076–93. <https://doi.org/10.1021/acs.iecr.8b01093>.
- 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, Pedro M. Castro, and Ignacio E. Grossmann. 2013. "Global Optimization of Bilinear Programs with a Multiparametric Disaggregation Technique." *Journal of Global Optimization*. <https://doi.org/10.1007/s10898-012-0022-1>.
- 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>.
- Kolodziej, Scott P., Ignacio E. Grossmann, Kevin C. Furman, and Nicolas W. Sawaya. 2013. "A Discretization-Based Approach for the Optimization of the Multiperiod Blend Scheduling Problem." *Computers and Chemical Engineering* 53: 122–42. <https://doi.org/10.1016/j.compchemeng.2013.01.016>.
- Li, Jie, Ruth Misener, and Christodoulos A. Floudas. 2012. "Continuous-Time Modeling and Global Optimization Approach for Scheduling of Crude Oil Operations." *AIChE Journal* 58 (1): 205–26. <https://doi.org/10.1002/aic.12623>.
- Lotero, Irene, Francisco Trespacios, Ignacio E. Grossmann, Dimitri J. Papageorgiou, and Myun Seok Cheon. 2016. "An MILP-MINLP Decomposition Method for the Global Optimization of a Source Based Model of the Multiperiod Blending Problem." *Computers and Chemical Engineering*. <https://doi.org/10.1016/j.compchemeng.2015.12.017>.
- 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>.
- Maranas, Costas D., and Christodoulos A. Floudas. 1997. "Global Optimization in Generalized Geometric Programming." *Computers & Chemical Engineering* 21 (4): 351–69. [https://doi.org/10.1016/S0098-1354\(96\)00282-7](https://doi.org/10.1016/S0098-1354(96)00282-7).
- 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>.
- Merchan, Andres F., Hojae Lee, and Christos T. Maravelias. 2016. "Discrete-Time Mixed-Integer Programming Models and Solution Methods for Production Scheduling in Multistage Facilities." *Computers & Chemical Engineering* 94 (November): 387–410. <https://doi.org/10.1016/J.COMPCHENG.2016.04.034>.
- Merchan, Andres F., Sara Velez, and Christos T. Maravelias. 2013. "Tightening Methods for Continuous-Time Mixed-Integer Programming Models for Chemical Production Scheduling." *AIChE Journal* 59 (12): 4461–67. <https://doi.org/10.1002/aic.14249>.
- 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/0046352e7d1dfcb40f000000/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.COMPCHENG.2010.02.014>.
- Misener, Ruth, Jeffrey P. Thompson, and Christodoulos A. Floudas. 2011. "APOGEE: Global Optimization of Standard, Generalized, and Extended Pooling Problems via Linear and Logarithmic Partitioning Schemes." *Computers & Chemical Engineering* 35 (5): 876–92. <https://doi.org/10.1016/J.COMPCHENG.2011.01.026>.
- 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>.
- Puranik, Yash, and Nikolaos V. Sahinidis. 2017. "Domain Reduction Techniques for Global NLP and MINLP Optimization." *Constraints* 22 (3): 338–76.

<https://doi.org/10.1007/s10601-016-9267-5>.

- Quesada, I., and I.E. Grossmann. 1995. "Global Optimization of Bilinear Process Networks with Multicomponent Flows." *Computers & Chemical Engineering* 19 (12): 1219–42. [https://doi.org/10.1016/0098-1354\(94\)00123-5](https://doi.org/10.1016/0098-1354(94)00123-5).
- Ryoo, Hong S., and Nikolaos V. Sahinidis. 1996. "A Branch-and-Reduce Approach to Global Optimization." *Journal of Global Optimization* 8 (2): 107–38. <https://doi.org/10.1007/BF00138689>.
- Savelsbergh, M. W. P. 1994. "Preprocessing and Probing Techniques for Mixed Integer Programming Problems." *ORSA Journal on Computing* 6 (4): 445–54. <https://doi.org/10.1287/ijoc.6.4.445>.
- 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>.
- Shectman, J. Parker, and Nikolaos V. Sahinidis. 1998. "A Finite Algorithm for Global Minimization of Separable Concave Programs." *Journal of Global Optimization* 12 (1): 1–36. <https://doi.org/10.1023/A:1008241411395>.
- Smith, E.M.B., and C.C. Pantelides. 1999. "A Symbolic Reformulation/Spatial Branch-and-Bound Algorithm for the Global Optimisation of Nonconvex MINLPs." *Computers & Chemical Engineering* 23 (4–5): 457–78. [https://doi.org/10.1016/S0098-1354\(98\)00286-5](https://doi.org/10.1016/S0098-1354(98)00286-5).
- Street, Larimer. 1989. "Constraint Propagation, Relational Arithmetic in AI Systems and Mathematical Programs." *Annals of Operations Research* 21: 143–48.
- 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.
- Velez, Sara, and Christos T. Maravelias. 2013a. "Mixed-Integer Programming Model and Tightening Methods for Scheduling in General Chemical Production Environments." *Industrial & Engineering Chemistry Research* 52 (9): 3407–23. <https://doi.org/10.1021/ie302741b>.
- . 2013b. "Reformulations and Branching Methods for Mixed-Integer Programming Chemical Production Scheduling Models." *Industrial & Engineering Chemistry Research* 52 (10): 3832–41. <https://doi.org/10.1021/ie303421h>.
- Velez, Sara, Arul Sundaramoorthy, and Christos T. Maravelias. 2013. "Valid Inequalities Based on Demand Propagation for Chemical Production Scheduling MIP Models." *AIChE Journal* 59 (3): 872–87. <https://doi.org/10.1002/aic.14021>.
- 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>.