An MILP-MINLP decomposition method for the global optimization of a source based model of the multiperiod blending problem

Irene Lotero[a], Francisco Trespalacios[a],
Myun-Seok Cheon[b], Dimitri J. Papageorgiou[b] and Ignacio E. Grossmann[a,*]

[a] Department of Chemical Engineering, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[b] ExxonMobil, Corporate Strategic Research
1545 Route 22 East, Annandale, NJ 08801, USA
* *Author to whom correspondence should be addressed: grossmann@cmu.edu*

Abstract

The multiperiod blending problem involves binary variables and bilinear terms, yielding a nonconvex MINLP. In this work we present two major contributions for the global solution of the problem. The first one is an alternative formulation of the problem. This formulation makes use of redundant constraints that improve the MILP relaxation of the MINLP. The second contribution is an algorithm that decomposes the MINLP model into two levels. The first level, or master problem, is an MILP relaxation of the original MINLP. The second level, or subproblem, is a smaller MINLP in which some of the binary variables of the original problem are fixed. The results show that the new formulation can be solved faster than alternative models, and that the decomposition method can solve the problems faster than state of the art general purpose solvers.

1 Introduction

Many processes in the petrochemical industry involve the blending of intermediate and final products. Large cost savings can be achieved by efficient blending schemes that satisfy the technical and regulatory specifications of products. For example, the economic and operability benefits from optimal crude-oil blend scheduling can reach multimillion dollars per year[1].

One of the first mathematical programming models to represent the scheduling of blending operations is the pooling problem[2]. The pooling problem seeks to find the optimal blend of materials available from a set of supply streams, while satisfying the demand of a set of products. The model enforces that the end products satisfy a specified minimum and maximum level for each specification. The objective is to minimize the total cost (or maximize the profit) of the operation. Several optimization models for the pooling problem have been reported in the literature. The p-formulation[2], based on total flows and component compositions, is commonly used in chemical process industries. The q-formulation[3] uses variables based on the fraction that each input stream contributes to the total input to each pool, and does not explicitly use the pool specifications as variables. The pq-formulation[4] is obtained by including valid redundant inequalities in the q-formulation. Tawarmalani and Sahinidis[4] prove that the redundant constraints help to obtain a stronger polyhedral relaxation of the pooling problem. Lastly, Audet et al.[5] propose a hybrid formulation by combining the p and q models to avoid additional bilinear terms that arise when generalized pooling problems are modeled using the q-formulation.

The multiperiod blending problem can be regarded as an extension of the pooling problem. In addition to the pooling problem restrictions, it considers inventory and time variations of supply and demand. The multiperiod blending problem can be formulated as a mixed-integer nonlinear programming (MINLP) problem[6]. Binary variables are required to model the movements of materials in and out of the tanks and to account for fixed costs. Even in the absence of binary variables, bilinear terms (which are necessary to model the mixing of various streams) make the problem nonconvex. Due to this highly combinatorial and nonconvex nature, the blend scheduling problem is very challenging. General purpose global optimization solvers fail to solve even small instances.

To the best of our knowledge, Foulds et al.[7] were the first to propose a global optimization algorithm to solve a single-component pooling problem. They use McCormick envelopes[8] to relax the bilinear terms. Androulakis et al.[9] propose a convex quadratic NLP relaxation, known as αBB underestimator. However, due to its generality, the NLP relaxation is weaker than its LP counterpart. Ben-Tal et al.[3] and Adhya et al.[10] present different Lagrangean relaxation approaches for developing lower bounds for the pooling problem. These bounds are tighter than standard LP relaxations used in global optimization algorithms.

In the context of processing network problems, Quesada and Grossmann[11] apply the reformulation-linearization technique (RLT)[12], together with McCormick envelopes, to improve the relaxation of a bilinear program by creating redundant constraints. These authors combine concentration and flow based models in order to obtain a relaxed LP formulation that provides a valid and strong lower bound to the global optimum. Similar results are obtained by Tawarmalani and Sahinidis[4] for the multicomponent pooling problem. The idea of using redundant constraints to strengthen the relaxation of the original problem is also used by Karuppiah et al.[13] in the context of water networks. These constraints correspond to total mass balance of contaminants and serve as deep cuts in the McCormick relaxation.

Piecewise MILP relaxations are an alternative relaxation of MINLPs that provide stronger bounds than traditional MILP relaxations. The first references to the use of piecewise MILP relaxation are by Bergamini et al.[14] and Karuppiah et al.[13]. Following this idea, Wicaksono and Karimi[15] propose several novel formulations for piecewise MILP under and overestimators for bilinear programs. Gounaris et al.[16] present a comprehensive computational comparison study of a collection of fifteen piecewise linear relaxations over a collection of benchmark pooling problems. Misener et al.[17], building on the ideas from Vielma and Nemhauser[18], introduce a formulation for the piecewise linear relaxation of bilinear functions with a logarithmic number of binary variables. Another alternative to piecewise relaxations are discretization techniques, such as multiparametric disaggregation[19, 20]. The number of additional binary variables increases linearly with each increment in the precision of the discretization.

As an alternative to branch-and-bound solution procedures, Kolodziej et al.[19] propose a heuristic as well as a rigorous two-stage MILP-NLP and MILP-MILP global optimization algorithms. Approximate and relaxed MILPs are obtained through the multiparametric disaggregation technique. Kesavan et al.[21] propose two approaches to generalize the outer approximation algorithm to separable nonconvex MINLP. Similarly, Bergamini et al.[14], based on the work from Turkay and Grossmann[22], present a deterministic algorithm based on logic-based outer approximation that can guarantee global optimality in the solution of an optimal the synthesis of process network problem.

Although the multiperiod blending problem arises in several applications, crude-oil blending is of great importance due to the potential increase in profit derived from optimal operation. In fact, crude-oil costs account for about 80% of the refinery turnover[23]. As a scheduling extension of the blending problem, crude-oil scheduling involves the unloading of crude marine vessels into storage tanks, followed by the transfer of crude from storage to charging tanks and finally, to the crude-oil distillation units (CDUs)[24, 25]. Lately, crude-oil scheduling models incorporate more quantity, quality, and logistics decisions related to real-life refinery operations, such as minimum run-length requirements, one-flow out of blender or sequence-dependent switchovers[26].

Several authors have proposed different algorithms relying on mixed-integer linear formulations to avoid solving the full nonconvex MINLP. These models can be seen as relaxations of the original MINLP. Mendez et al.[27] present a novel MILP-based method where a very complex MINLP formulation is replaced by a sequential MILP approximation that can deal with non-linear gasoline properties and variable recipes for different product grades. Similarly, a two-stage MILP-NLP solution procedure is employed by Jia et al.[28] and Mouret et al.[29], featuring in the first stage a relaxed MILP model without the bilinear blending constraints followed by the solution of the original MINLP after fixing all binary variables. The same two-stage algorithm is studied by Castro and Grossmann[30] together with several global optimization methods. However, instead of dropping the bilinear constraints in the two-stage algorithm, they use multiparametric disaggregation to relax the bilinear terms. Moro and Pinto[31] and Karuppiah et al.[13] tackle the problem with the augmented penalty version and a specialized version of the outer-approximation method, respectively. Reddy et al.[32] propose an MILP relaxation combined with a rolling-horizon algorithm to eliminate the composition discrepancy. Finally, Li et al.[23] use a spatial branch-and-bound global optimization algorithm, that at each node uses the MILP-NLP two-stage strategy previously mentioned, to solve the MINLP problems.

Even though, continuous-time models seem to be preferred for crude-oil scheduling, the demand-driven nature of the multiperiod blending problem makes a simple discrete-time framework a better choice for our problem. Despite the latest modeling and algorithmic advances for this class of problems, large instances are still intractable. Improvements or even new problem formulations and solution approaches must be proposed.

Generalized Disjunctive Programming (GDP) is a higher level representation of discrete/continuous optimization problems[33]. GDP problems can be reformulated and solved as MILP/MINLP problems. In this work, problems are presented as GDP and solved as MILP/MINLP. Section 2.2 presents more details on this reformulation.

In this work, we make two primary contributions for solving multiperiod blending problems. The first is an alternative formulation of the problem, in terms of generalized disjunctive programming (GDP), that makes use of redundant constraints. These constraints considerably improve the linear GDP relaxation of the nonlinear GDP. Based on the observation that one can reduce the complexity of a problem by fixing values of certain variables, a decomposition method is proposed next. The algorithm decomposes the GDP model into two levels. The first level, or master problem, is a linear GDP relaxation of the original GDP that provides rigorous upper bounds. The second level, or subproblem, is a smaller GDP in which some of the Boolean variables of the original problem are fixed. The subproblem, when a feasible solution is found, provides a feasible solution to the original GDP and a rigorous lower bound. These problems are solved successively until the gap between the upper and lower bound is closed. We illustrate the new formulation and decomposition method with several test problems. The results show that the new formulation can be solved faster than the alternatives, and that the decomposition method can solve the problems faster than state-of-the-art general purpose solvers.

Table 1: Nomenclature of sets and variables

Sets	Symbols	Element
Total number of tanks	$\mathcal{N} = \mathcal{S} \cup \mathcal{B} \cup \mathcal{D}$	n
Blending tanks	$\mathcal{B}$	b
Supply tanks	$\mathcal{S}$	s
Demand tanks	$\mathcal{D}$	d
Specifications	$\mathcal{Q}$	q
Sources	$\mathcal{R}$	r
Time periods	$\mathcal{T}$	t
Variables	Symbols	Sets
Continuous Variables		
Flow between tanks n and n' at the end of time t	$F_{nn't}$	$(n, n') \in \mathcal{A}, t \in \mathcal{T}$
Demand flow from tanks d at time t	FD_{dt}	$d \in \mathcal{D}$, $t \in \mathcal{T}$
Inventory in tank n at the end of time t	I_{nt}	$n \in \mathcal{N}$, $t \in \mathcal{T}$
Specification q in tank b at the end of time t	C_{qbt}	$q \in \mathcal{Q}$, $b \in \mathcal{B}$, $t \in \mathcal{T}$
Flow of specification q between tanks n and n' at time t	$\bar{F}_{qnn't}$	$q \in \mathcal{Q}$, $(n, n') \in \mathcal{A}$, $t \in \mathcal{T}$
Inventory of specification q in blending tank b at time t	$\bar{I}_{qbt}$	$q \in \mathcal{Q}$, $n \in \mathcal{N}$, $t \in \mathcal{T}$
Flow of source r between tanks n and n' at time t	$\tilde{F}_{rnn't}$	$r \in \mathcal{R}$, $(n, n') \in \mathcal{A}$, $t \in \mathcal{T}$
Inventory of source r in blending tank b at time t	$\tilde{I}_{rbt}$	$r \in \mathcal{R}$, $n \in \mathcal{N}$, $t \in \mathcal{T}$
Fraction of inventory in blending tank b sent to tank n at the end of time t	ξ_{bnt}	$(b, n) \in \mathcal{A}$, $t \in \mathcal{T}$
Boolean Variables		
Variable that indicates the existence of flow between tanks n and n' at the end of time t	$X_{nn't}$	$(n, n') \in \mathcal{A}$, $t \in \mathcal{T}$
Variable that indicates the operating mode of blending tank b at time t. $YB_{bt} = True$ indicates the tank is charging. $YB_{bt} = False$ indicates the tank is discharging	YB_{bt}	$b \in \mathcal{B}$, $t \in \mathcal{T}$
Binary Variables		
Binary variable that corresponds to Boolean variable Y_{bt}. $yb_{bt} = 1$ indicates the tank is charging. $yb_{bt} = 0$ indicates the tank is discharging	yb_{bt}	$b \in \mathcal{B}$, $t \in \mathcal{T}$

Table 2: Nomenclature of parameters

Parameters	Symbols	Sets
Initial inventory for tank n	I_n^0	$n \in \mathcal{N}$
Initial values for the specifications q in tank b	C_{qb}^0	$q \in \mathcal{Q}$, $b \in \mathcal{B}$
Incoming supply flows enter tank s at time t	F_{st}^{IN}	$s \in \mathcal{S}$, $t \in \mathcal{T}$
Specification q in supply flow to tank s	C_{qs}^{IN}	$q \in \mathcal{Q}$, $s \in \mathcal{S}$
Specification q in source r	$\hat{C}_{qr}^0$	$q \in \mathcal{Q}$, $r \in \mathcal{R}$
Bounds on demand flow from tanks d at time t	$[FD_{dt}^{\mathrm{L}}, FD_{dt}^{\mathrm{U}}]$	$d \in \mathcal{D}$, $t \in \mathcal{T}$
Bound on specification q in demand tank d	$[C_{qd}^{\mathrm{L}}, C_{qd}^{\mathrm{U}}]$	$q \in \mathcal{Q}$, $d \in \mathcal{D}$
Bounds on inventory for tank n	$[I_n^{\mathrm{L}}, I_n^{\mathrm{U}}]$	$n \in \mathcal{N}$
Bounds on flow between tank n and n'	$[F_{nn'}^{\mathrm{L}}, F_{nn'}^{\mathrm{U}}]$	$(n, n') \in \mathcal{A}$
Costs for the supply flow for tank s	β_s^T	$s \in \mathcal{S}$
Prices for demand flow for tank d	β_d^T	$d \in \mathcal{D}$
Fixed costs for flow from tank n to tank n'	$\alpha_{nn'}^N$	$(n, n') \in \mathcal{A}$
Variable costs for flow from tank n to tank n'	$\beta_{nn'}^N$	$(n, n') \in \mathcal{A}$

Table 3: GDP models

Model	**Description**
($\mathbb{C}$)	Concentration of individual specifications is a variable. Bilinear terms appear when blending tank is in “charging” mode.
($\mathbb{SF}$)	Flow and inventory of specifications and split fraction are variables. Bilinear terms appear when blending tank is in “discharging” mode.
($\mathbb{SB}$)	Flow and inventory of sources and split fraction are variables. Bilinear terms appear when blending tank is in “discharging” mode.
($\mathbb{CSB}$)	Same as ($\mathbb{C}$), but including redundant constraints from ($\mathbb{SB}$).
($\mathbb{MP}$)	Master problem. Linear relaxation of ($\mathbb{CSB}$), including enumeration cuts.
($\mathbb{SP}$)	Subproblem using ($\mathbb{CSB}$) model with Y_{bt} fixed.

2 The Multiperiod Blending Problem

The multiperiod blending problem is defined on a network $(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of arcs connecting these nodes. The set of nodes is partitioned into three subsets corresponding to the types of tanks: supply nodes $s \in \mathcal{S}$, blending nodes $b \in \mathcal{B}$, and demand nodes $d \in \mathcal{D}$. Directed arcs $(n, n') \in \mathcal{A}$ between nodes correspond to streams from tank n to tank n'. In general, interconnections between the supply and demand tanks, as well as between blending tanks, are allowed by the model. The streams and inventories in the system possess different specifications $q \in \mathcal{Q}$, such as concentration of chemical compounds or physical properties. The network operates over a time horizon composed of multiple time periods, $\mathcal{T} = \{0, 1, \ldots, T\}$, over which demand within concentration specifications has to be satisfied at the end of each time period. Figure 1 shows a schematic representation of the blending system.

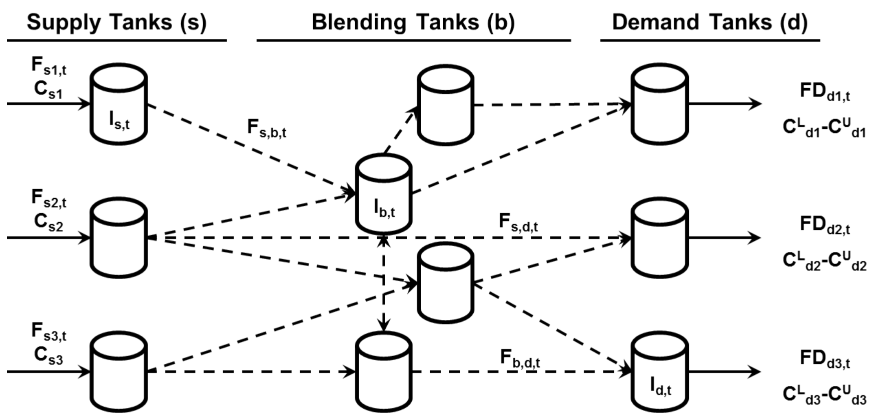

Figure 1: Sketch of the multiperiod blending problem

Given fixed feed compositions C^{IN}_{qs} and incoming flows F^{IN}_{st} to the supply tanks, as well as initial conditions in each tank, the problem consists of determining the flows $F_{nn't}$, FD_{dt}, inventories I_{nt} and compositions C_{qbt} in the network in each time period so as to maximize the profit (or minimize the total cost) of the blending schedule, while meeting the demand limits $[FD^{\mathrm{L}}_{dt}, FD^{\mathrm{U}}_{dt}]$ within specified limits of composition $[C^{\mathrm{L}}_{qd}, C^{\mathrm{U}}_{qd}]$.

Note that each time period $t \in \mathcal{T}$ is not independent of the others due to the coupling created by the inventories [6]. For instance, the composition and flow of an outgoing stream from a blending tank at time t depends on the inventory in that tank at the end of the previous time period, $t-1$. As a consequence, the optimization must be performed simultaneously over all time periods.

For simplicity, the composition of the incoming flow to the supply tanks C^{IN}_{qs} and the bounds on the concentration of flows leaving the demand tanks $[C^{\mathrm{L}}_{qd}, C^{\mathrm{U}}_{qd}]$ are assumed to be constant over the time horizon. As a result, the compositions C_{qbt} in the blending tanks in each time period are the only ones that are unknown in the system (hence the subscript b instead of n in C_{qbt}). On the other hand, the supply and demand flows can vary in amount over time (hence the subscript t in F^{IN}_{st} and $[FD^{\mathrm{L}}_{dt}, FD^{\mathrm{U}}_{dt}]$).

The system operates within bounds on the inventories $[I^{\mathrm{L}}_{n}, I^{\mathrm{U}}_{n}]$, and on the flows $[F^{\mathrm{L}}_{nn'}, F^{\mathrm{U}}_{nn'}]$

between each pair of tanks $(n, n') \in \mathcal{A}$.

In order to quantify the profit of the blending process, costs for the supply flows β_s^T, prices for the demand flows β_d^T, and fixed and variable costs $[\alpha_{nn'}^N, \beta_{nn'}^N]$ for the flows within the network are taken into account.

An important assumption is that, due to operational and safety considerations, simultaneous input/output streams to blending tanks is not allowed, i.e. flow cannot enter and exit a blending tank in the same time period. Boolean variables $X_{nn't}$, which represent existence ($X_{nn't} = True$) or absence ($X_{nn't} = False$) of flow between tanks n and n', are required to model this assumption, as well as to represent fixed costs for using the pipelines in the objective function. Additional Boolean variables (YB_{bt}) are used to represent the operating mode of a blending tank ($YB_{bt} = True$ if a tank is "charging" and $YB_{bt} = False$ if a tank is "discharging"). Finally, the multiple liquid streams that enter the blending tanks are assumed to be perfectly mixed at the end of the time period.

Tables 1 and 2 contain a detailed explanation of the nomenclature used for sets, variables and data in the problem. It can be noted from these tables that parameters contain a superscript and variables do not. Table 3 contains the different models presented in this work, as well as a brief description with the main difference among them.

2.1 Motivating Example

In this section we present a small illustrative example to provide some insight on the complexity of multiperiod blending problems. It should be noted that the instance is significantly simple so that the solution can in fact be obtained by inspection. The instance consists of 2 supply tanks, 8 blending tanks, 2 demand tanks, 6 time periods and 1 specification. The topology of the network is shown in Figure 2.

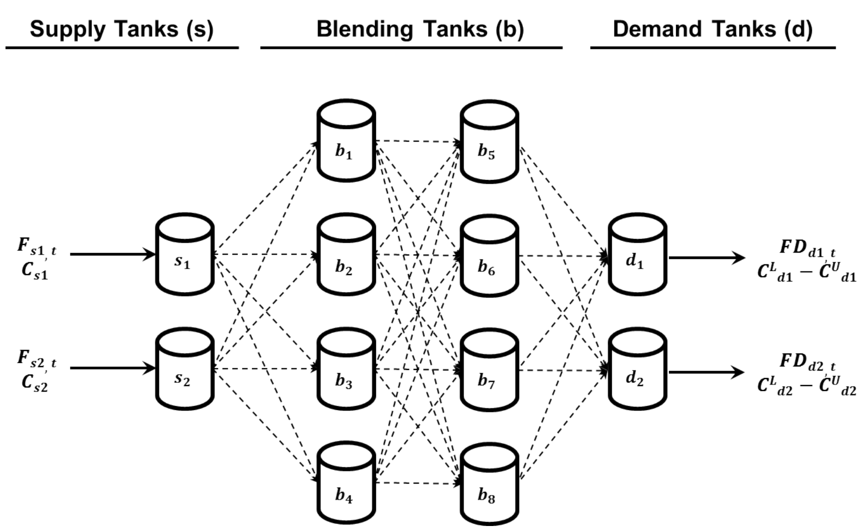

Figure 2: Topology of the motivating example

Tables 4 and 5 contain the parameters of the supply and demand streams. The initial inventory and concentration are zero for all blending tanks, $I_b^0 = C_{qn}^0 = 0 \ \forall \ b \in \mathcal{B}, \ q \in \mathcal{Q}$.

There is no inventory capacity in supply and demand tanks ($I^{\mathrm{U}}_s = I^{\mathrm{U}}_d = 0\ \forall\ s \in \mathcal{S},\ d \in \mathcal{D}$). The maximum inventory in the blending tanks is 30 for the first row of tanks ($I^{\mathrm{U}}_b = 30\ \ \forall\, b \in \{1,2,3,4\}$) and 20 for the second row of tanks ($I^{\mathrm{U}}_b = 20\ \ \forall\, b \in \{5,6,7,8\}$). The maximum flow between tanks is 30 ($F^{\mathrm{U}}_{nn'} = 30\ \ \forall\,(n,n') \in \mathcal{A}$). The fixed cost for using the pipelines of 0.1 ($\alpha^{N}_{nn'} = 0.1\ \ \forall\,(n,n') \in \mathcal{A}$).

Table 4: Supply tank specifications

	C^{IN}_s	F^{IN}_{st}						
Supply tank	Qual.A	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	β^T_s
s_1	0.06	10	10	10	0	0	0	0
s_2	0.26	30	30	30	0	0	0	0

Table 5: Demand tank specifications

	$[C^L_{qd}, C^U_{qd}]$	FD^L_{dt}						
Demand tank	Qual.A	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	β^T_d
d_1	$[0, 0.16]$	0	0	15	15	15	15	2
d_2	$[0, 1]$	0	0	15	15	15	15	1

Note the rigid structure of the instance. The sum of the supply flow over the time horizon equals the demand. Since the initial inventory is zero, all the supply should be used to satisfy the demand, thus all blending tanks will be empty at the end of the time horizon. Besides, the supply with low concentration of specification A should be equally mixed with flow from supply s_2 in order to satisfy the specifications of demand tank d_1. The rest of supply s_2 can be sent directly to demand tank d_2 because there is no upper limit for specification A. The uneven inventory upper bounds on the tanks and the high symmetry derived from an empty initial inventory, increases the complexity of a seemingly simple instance. The maximum profit of this problem is 177.5 and an optimal flow schedule is shown in Figure 3. Table 6 contains the dimensions of the problem in terms of number of variables, constraints and bilinear terms.

Table 6: Size of the ($\mathbb{C}$) formulation (explained below) for the motivating example

Continuous Variables	Binary variables	Constraints	Bilinear terms
584	240	1178	128

Even though it is a relative trivial instance, global optimization solvers, such as BARON 14.0[34], ANTIGONE 1.1[35] or SCIP 3.1[36], have difficulty even finding a feasible solution

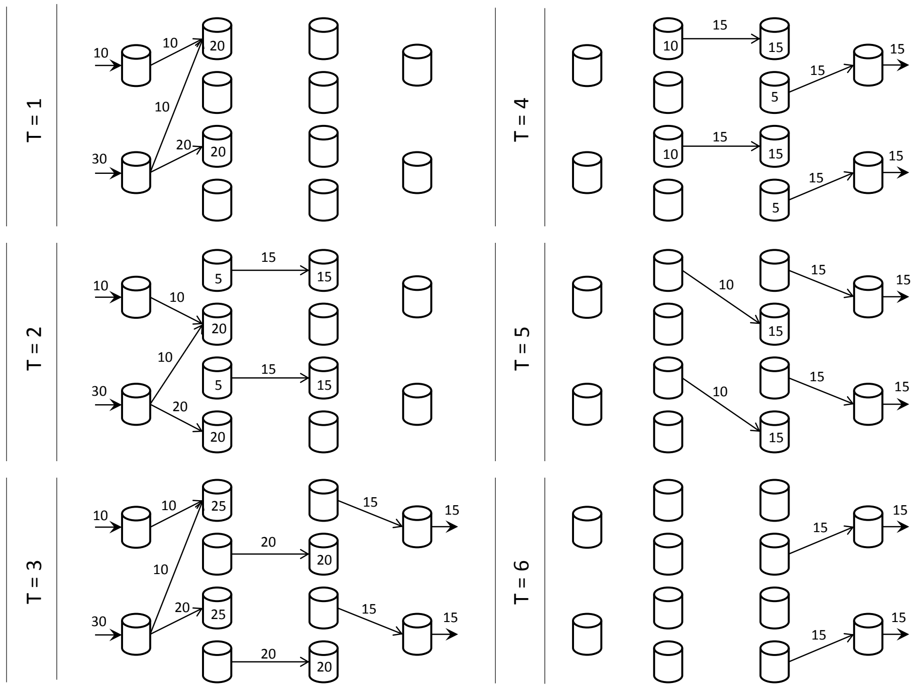

Figure 3: An optimal flow schedule for the motivating example

to this problem when using the original MINLP formulation of Kolodziej et al.[6]. In fact, after 30 minutes of computational time, none of them reported a feasible solution. As mentioned before, this example motivates the need for alternative formulations and customized techniques that can handle even larger instances.

2.2 Generalized Disjunctive Programming (GDP) Formulations

In this section we present two alternative formulations for the multiperiod blending problem: a concentration model ($\mathbb{C}$) and a split fraction model ($\mathbb{SF}$). The concentration model ($\mathbb{C}$) includes the concentration of individual specifications as variables. As such, the bilinear terms of this formulation appear when a tank is "blending". The split fraction model ($\mathbb{SF}$) includes as variables the flow and inventory of individual specifications, and the split fraction of discharge. As such, the bilinear terms appear when a tank is "discharging". Both formulations are presented as Generalized Disjunctive Programming (GDP) models.

GDP is an alternative method for formulating discrete/continuous optimization problems. GDP models can be reformulated as MINLP problems using either the Big-M (BM) or hull-reformulation (HR). For more details on the reformulations, we refer the reader to the review work on modeling optimization problems through generalized disjunctive programming[33]. In this work, problems are formulated as GDP models and solved as MINLPs by using the (BM) reformulation.

To illustrate the (BM) reformulation consider the following simple disjunction:

$$
\begin{aligned}
&\begin{bmatrix} X \\ F^{\mathrm{L1}} \le F \le F^{\mathrm{U1}} \end{bmatrix} \vee \begin{bmatrix} \neg X \\ F^{\mathrm{L2}} \le F \le F^{\mathrm{U2}} \end{bmatrix} \\
&F \in \mathbb{R} \\
&X \in \{True, False\}
\end{aligned} \tag{1}
$$

where F is a continuous variable, $[F^{\mathrm{L1}}, F^{\mathrm{U1}}, F^{\mathrm{L2}}, F^{\mathrm{U2}}]$ are parameters, and X is a Boolean variable. The GDP (1) can be reformulated as an MILP using (BM) as follows:

$$
\begin{aligned}
&F^{\mathrm{L1}} - F \le M_1(1-x) \\
&F - F^{\mathrm{U1}} \le M_2(1-x) \\
&F^{\mathrm{L2}} - F \le M_3 x \\
&F - F^{\mathrm{U2}} \le M_4 x \\
&F \in \mathbb{R} \\
&x \in \{0,1\}
\end{aligned} \tag{2}
$$

Note that in (2) the Boolean variable X is replaced by the binary variable x. M_1, M_2, M_3, M_4 are large enough parameters, so when $x = 1$ the first two constraints in (2) are enforced, while the third and fourth constraints become redundant. When $x = 0$ the third and fourth constraints are enforced, and the first two become redundant. In this example, $M_1 = F^{\mathrm{L1}} - F^{\mathrm{L2}}$, $M_2 = F^{\mathrm{U2}} - F^{\mathrm{U1}}$, $M_3 = F^{\mathrm{L2}} - F^{\mathrm{L1}}$, $M_4 = F^{\mathrm{U1}} - F^{\mathrm{U2}}$ are valid M-parameters.

Kolodziej et al.[6] presented an MINLP model for the multiperiod blending problem, in terms of total flow and concentration. The GDP formulation of this concentration model ($\mathbb{C}$) is as follows:

($\mathbb{C}$):

$$
\max \quad \sum_{t \in \mathcal{T}} \Big[\sum_{(n,d) \in \mathcal{A}} \beta_d^T F_{ndt} - \sum_{(s,n) \in \mathcal{A}} \beta_s^T F_{snt} - \sum_{(n,n') \in \mathcal{A}} (\alpha_{nn'}^N x_{nn't} + \beta_{nn'}^N F_{nn't}) \Big] \tag{3}
$$

s.t.

$$
I_{st} = I_{st-1} + F_{st}^{\mathrm{IN}} - \sum_{(s,n) \in \mathcal{A}} F_{snt} \qquad \forall\, s \in \mathcal{S},\, t \in \mathcal{T} \tag{4a}
$$

$$
I_{dt} = I_{dt-1} + \sum_{(n,d) \in \mathcal{A}} F_{ndt} - FD_{dt} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{4b}
$$

$$
\begin{bmatrix} X_{nbt} \\ F_{nb}^{\mathrm{L}} \le F_{nbt} \le F_{nb}^{\mathrm{U}} \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \forall\, (n,b) \in \mathcal{A},\, t \in \mathcal{T} \tag{5}
$$

$$
\begin{bmatrix} X_{sdt} \\ F_{sd}^{\mathrm{L}} \le F_{sdt} \le F_{sd}^{\mathrm{U}} \\ C_{qd}^{\mathrm{L}} \le C_{qs}^{\mathrm{IN}} \le C_{qd}^{\mathrm{U}} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \end{bmatrix} \qquad \forall\, (s,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{6}
$$

$$\begin{bmatrix} X_{bdt} \\ F^{\mathrm{L}}_{bd} \leq F_{bdt} \leq F^{\mathrm{U}}_{bd} \\ C^{\mathrm{L}}_{qd} \leq C_{qbt-1} \leq C^{\mathrm{U}}_{qd} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \forall\, (b,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{7}$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_{bt-1} + \sum_{(n,b)\in\mathcal{A}} F_{nbt} \\ I_{bt}C_{qbt} = I_{bt-1}C_{qbt-1} + \sum_{(s,b)\in\mathcal{A}} F_{sbt}C^{\mathrm{IN}}_{qs} \\ + \sum_{(b',b)\in\mathcal{A}} F_{b'bt}C_{qb't-1} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg YB_{bt} \\ I_{bt} = I_{bt-1} - \sum_{(b,n)\in\mathcal{A}} F_{bnt} \\ C_{qbt} = C_{qbt-1} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \; \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{8}$$

$$X_{nbt} \Rightarrow YB_{bt} \qquad \forall\, (n,b) \in \mathcal{A},\, t \in \mathcal{T} \tag{9a}$$

$$X_{bnt} \Rightarrow \neg YB_{bt} \qquad \forall\, (b,n) \in \mathcal{A},\, t \in \mathcal{T} \tag{9b}$$

$$I^{\mathrm{L}}_{n} \leq I_{nt} \leq I^{\mathrm{U}}_{n} \qquad \forall\, n \in \mathcal{N},\, t \in \mathcal{T} \tag{10a}$$

$$F^{\mathrm{L}}_{nn'} \leq F_{nn't} \leq F^{\mathrm{U}}_{nn'} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{10b}$$

$$FD^{\mathrm{L}}_{dt} \leq FD_{dt} \leq FD^{\mathrm{U}}_{dt} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{10c}$$

$$C^{\mathrm{L}}_{q} \leq C_{qbt} \leq C^{\mathrm{U}}_{q} \qquad \forall\, q \in \mathcal{Q},\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{10d}$$

$$X_{nn't} \in \{True, False\} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{11a}$$

$$YB_{bt} \in \{True, False\} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{11b}$$

In ($\mathbb{C}$), the objective function (3) maximizes the profit that results from delivering products to the demand tanks, minus the costs associated with supply flows as well as fixed and variable costs of transferring the liquids between tanks. Note that costs and revenues are accounted through flows leaving the supply tanks and entering the demand tanks. Equations (4) are total mass balances over the supply and demand tanks.

Disjunctions (5) to (7) represent the set of constraints regarding the existence of flow between nodes. If the flow between nodes exists ($X_{nn't} = True$), then upper and lower bounds on flow and concentration are enforced. If the flow does not exist ($X_{nn't} = False$), then the flow is zero and no concentration constraints are enforced. Note that in disjunction (6) C^{IN}_{qs} is a parameter. However, this disjunction enforces that there can only exist flow between supply and demand when the supply specifications lie within the demand bounds ($C^{\mathrm{L}}_{qd} \leq C^{\mathrm{IN}}_{qs} \leq C^{\mathrm{U}}_{qd}$).

Disjunction (8) models the operation of the blending tanks. Since there cannot be simultaneous input/output streams to blending tanks, they can be either charging or discharging but not both. The total mass balance of the inventory is calculated if a tank is either charging or discharging. However, the individual specification inventory balance is only calculated

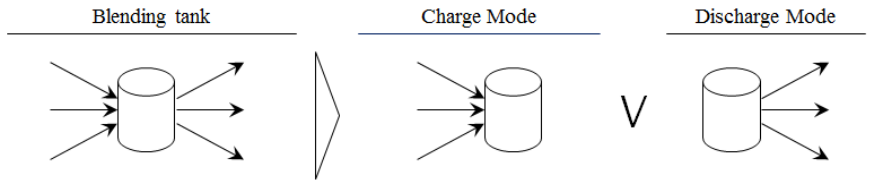

Figure 4: Illustration of no simultaneous input/output streams in a blending tank

when a tank is charging ($YB_{bt} = True$). When it is discharging ($YB_{bt} = False$), the required constraint specifies that there is no change in the concentration from the previous time period. Figure 4 illustrates the disjunction used to model the blending tanks.

Constraints (9) state the logic relationship between the binary variables. If there is flow coming into a blending tank ($X_{nbt} = True$), then YB_{bt} must be active ($YB_{bt} = True$) to indicate that it is in charging mode; the opposite if flow is leaving the tank. The last set of constraints (10) impose upper and lower bounds on the variables.

It is important to note that the original MINLP formulation of Kolodziej et al.[6] does not make use of disjunction (8). Instead, the mass balance of the blending tanks is described through global constraints. Introducing (8) in the formulation not only makes the "no simultaneous charge/discharge" condition more explicit, but it also reduces the number of bilinear terms. The reason for this reduction is that the mass balance individual specifications is defined as a global constraint in the model by Kolodziej et al. ($I_{bt}C_{qbt} = I_{bt-1}C_{qbt-1} + \sum_{(s,b)\in\mathcal{A}} F_{sbt}C_{qs}^{\mathrm{IN}} + \sum_{(b',b)\in\mathcal{A}} F_{b'bt}C_{qb't-1} - \sum_{(b,n)\in\mathcal{A}} F_{bnt}C_{qbt-1}$). As such, bilinear terms appear in the constraint regardless if the tank is charging or discharging. Furthermore, the bilinear terms not only involve flow and concentration of blending tanks as in ($\mathbb{C}$), but also flow and concentration of the nodes connected to the blending tanks ($F_{bnt}C_{qbt-1}; (b,n) \in \mathcal{A}$). When compared with the original MINLP formulation of Kolodziej et al. for the motivating example presented before, the number of bilinear terms decreases 50%, from 248 to 128 . This formulation requires more binary variables but, due to the logic implications (9), it does not increase the combinatorial complexity of the problem.

Model ($\mathbb{C}$) uses total flows, inventories and concentration of specifications as variables ($F_{nn't}$, I_{nt} and C_{qbt}). In this sense, formulation ($\mathbb{C}$) is akin to the p-formulation of the pooling problem. An alternative formulation for the multiperiod blending problem is the split fraction model ($\mathbb{SF}$). ($\mathbb{SF}$) includes as variables the flow and inventory of individual specifications, and the split fraction of discharge ($\bar{F}_{qnn't}$, $\bar{I}_{qbt}$ and ξ_{bnt}). This type of model was first proposed by Quesada and Grossmann[11] in their work on general process networks.

The split fraction model ($\mathbb{SF}$) is as follows:

($\mathbb{SF}$):

$$\max \quad \sum_{t\in\mathcal{T}} \Big[\sum_{(n,d)\in\mathcal{A}} \beta_d^T F_{ndt} - \sum_{(s,n)\in\mathcal{A}} \beta_s^T F_{snt} - \sum_{(n,n')\in\mathcal{A}} (\alpha_{nn'}^N y_{nn't} + \beta_{nn'}^N F_{nn't}) \Big] \tag{12}$$

s.t.

$$I_{st} = I_{st-1} + F^{\mathrm{IN}}_{st} - \sum_{(s,n)\in\mathcal{A}} F_{snt} \qquad \forall\, s \in \mathcal{S},\, t \in \mathcal{T} \tag{13a}$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d)\in\mathcal{A}} F_{ndt} - FD_{dt} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{13b}$$

$$\begin{bmatrix} X_{nbt} \\ F^{\mathrm{L}}_{nb} \le F_{nbt} \le F^{\mathrm{U}}_{nb} \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \\ \bar{F}_{qnbt} = 0 \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \qquad \forall (n,b) \in \mathcal{A},\, t \in \mathcal{T} \tag{14}$$

$$\begin{bmatrix} X_{sdt} \\ F^{\mathrm{L}}_{sd} \le F_{sdt} \le F^{\mathrm{U}}_{sd} \\ C^{\mathrm{L}}_{qd} \le C^{\mathrm{IN}}_{qs} \le C^{\mathrm{U}}_{qd} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \\ \bar{F}_{qbdt} = 0 \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \qquad \forall\, (s,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{15}$$

$$\begin{bmatrix} X_{bdt} \\ F^{\mathrm{L}}_{bd} \le F_{bdt} \le F^{\mathrm{U}}_{bd} \\ F_{bdt} C^{\mathrm{L}}_{qd} \le \bar{F}_{qbdt} \le F_{bdt} C^{\mathrm{U}}_{qd} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \\ \bar{F}_{qbdt} = 0 \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \qquad \forall\, (b,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{16}$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_{bt-1} + \sum\limits_{n\in\check{\mathcal{N}}_b} F_{nbt} \\ \bar{I}_{qbt} = \bar{I}_{qbt-1} + \sum\limits_{(s,b)\in\mathcal{A}} F_{sbt} C^{\mathrm{IN}}_{qs} \\ \qquad + \sum\limits_{(b',b)\in\mathcal{A}} \bar{F}_{qb'bt} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg YB_{bt} \\ I_{bt} = I_{bt-1} - \sum\limits_{(b,n)\in\mathcal{A}} F_{bnt} \\ \bar{I}_{qbt} = \bar{I}_{qbt-1} - \sum\limits_{(b,n)\in\mathcal{A}} \bar{F}_{qbnt} \quad \forall\, q \in \mathcal{Q} \\ F_{bnt} = \xi_{bnt} I_{bt-1} \qquad \forall\, (b,n) \in \mathcal{A} \\ \bar{F}_{qbnt} = \xi_{bnt} \bar{I}_{qbt-1} \quad \forall\, q \in \mathcal{Q},\, (b,n) \in \mathcal{A} \end{bmatrix} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{17}$$

$$X_{nbt} \Rightarrow YB_{bt} \qquad \forall\, (n,b) \in \mathcal{A},\, t \in \mathcal{T} \tag{18a}$$

$$X_{bnt} \Rightarrow \neg YB_{bt} \qquad \forall\, (b,n) \in \mathcal{A},\, t \in \mathcal{T} \tag{18b}$$

$$I^{\mathrm{L}}_{n} \le I_{nt} \le I^{\mathrm{U}}_{n} \qquad \forall\, n \in \mathcal{N},\, t \in \mathcal{T} \tag{19a}$$

$$F^{\mathrm{L}}_{nn'} \le F_{nn't} \le F^{\mathrm{U}}_{nn'} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{19b}$$

$$FD^{\mathrm{L}}_{dt} \le FD_{dt} \le FD^{\mathrm{U}}_{dt} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{19c}$$

$$I^{\mathrm{L}}_{b} C^{\mathrm{L}}_{q} \le \bar{I}_{qbt} \le I^{\mathrm{U}}_{b} C^{\mathrm{U}}_{q} \qquad \forall\, q \in \mathcal{Q},\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{19d}$$

$$F^{\mathrm{L}}_{nn'} C^{\mathrm{L}}_{q} \le \bar{F}_{qnn't} \le F^{\mathrm{U}}_{nn'} C^{\mathrm{U}}_{q} \qquad \forall\, q \in \mathcal{Q},\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{19e}$$

$$0 \le \xi_{bnt} \le 1 \qquad \forall\, (b,n) \in \mathcal{A},\, t \in \mathcal{T} \tag{19f}$$

$$X_{nn't} \in \{True, False\} \qquad \forall\, (n, n') \in \mathcal{A},\, t \in \mathcal{T} \tag{20a}$$

$$YB_{bt} \in \{True, False\} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{20b}$$

The main difference between ($\mathbb{C}$) and ($\mathbb{SF}$) is that in the former the concentration of individual specifications is a variable (C_{qbt}), while in the latter the flow and inventory of individual specifications, and the split fraction of discharge are the variables ($\bar{F}_{qnn't}$, $\bar{I}_{qbt}$ and ξ_{bnt}). In ($\mathbb{SF}$), constraints (12) and (13) are the same as constraints (3) and (4) in ($\mathbb{C}$). Constraints (14), (15) and (16) enforce flow and concentration bounds when there exists flow between two nodes. Disjunction (17) models the charging and discharging constraints of blending tanks. In order to enforce the same specification concentrations in the outflows and inventory of a tank, it is necessary to introduce a new variable ξ_{bnt}. When discharging, ξ_{bnt} represents the proportion of the inventory that flows to a tank ($F_{bnt} = \xi_{bnt} I_{bt-1}$). This proportion needs to be the same for the total flow and the flow of the individual specifications ($\bar{F}_{qbnt} = \xi_{bnt} \bar{I}_{qbt-1}$). Note that in formulation ($\mathbb{SF}$) the bilinear terms appear in the formulation every time a blending tank operates in discharge mode ($YB_{bt} = False$). Constraints (18) state the logic relationship between the binary variables. Finally, (19) impose the bounds on the variables.

Note that model ($\mathbb{SF}$) is not equivalent to the q-formulation of the generalized pooling problem[3]. The proportion variables in the q-formulation denote the fraction of incoming flow to the blending tank that is contributed by input n, which implies that the sum over all n add to 1. In other words, the variables model the incoming streams to the tank and not what is being withdrawn, which does not have to sum up to one if the tank is not being emptied completely. In addition, the variables in the q-formulation represent the fraction of raw materials that are supplied to the system, whereas the split fraction model tracks the specifications just as the concentration model. In other words, instead of fractions of raw materials, the variables of the split fraction model represent the actual amount of flow of each specification q in each and every stream. The q-formulation is discussed in more detail in Section 3.1. Nevertheless, Alfaki and Haugland[37] use proportions for flows transported from pools to demand tanks for the standard pooling problem and named it the TP formulation.

Table 7: Number of bilinear terms of GDP formulations. $\hat{B} = (b, b') \in \mathcal{A}$, $\hat{N}_b = (b, n) \in \mathcal{A}$

Model	Bilinear terms	Motivating Example $\lvert\mathcal{Q}\rvert = 5$, $\lvert\mathcal{T}\rvert = 6$
($\mathbb{C}$)	$\lvert\mathcal{Q}\rvert \left[\lvert\mathcal{B}\rvert\lvert\mathcal{T}\rvert + \lvert\hat{\mathcal{B}}\rvert(\lvert\mathcal{T}\rvert - 1)\right]$	640
($\mathbb{SF}$)	$\lvert\hat{\mathcal{N}}_b\rvert(\mathcal{T} - 1)(1 + \lvert\mathcal{Q}\rvert)$	720

Table 7 compares the number of bilinear terms of the two GDP models. Table 7 also shows that, for the motivating example, ($\mathbb{C}$) has fewer bilinear terms than ($\mathbb{SF}$) (640 vs. 720). However, depending on the structure of the network, number of tanks, time periods and specifications, one formulation can have more bilinear terms than the other. Note that the number of continuous variables is larger when the system is modeled using individual flows and inventories ($\mathbb{SF}$).

The proposed formulations imply that a decomposition approach can be used to exploit the operational constraint on the blending tanks. By deciding whether the tank is in charge or discharge mode, the number of binary variables representing the connection between blending tanks and the rest of the network is reduced. Moreover, if the operating mode of the tanks is fixed at each period of time, the number of bilinear terms can be further reduced, thus yielding smaller and easier problems to solve. Before presenting the details of the decomposition algorithm, an alternative formulation is proposed.

3 Improving Formulation with Redundant Constraints

A crucial feature for solving a nonconvex MINLP is the tightness of the formulation when the non-convex constraints are relaxed (i.e. the MILP relaxation of a nonconvex MINLP). Note that performing the linear relaxation on the GDP and then using the (BM) reformulation yields the exact same MILP as first using the (BM) reformulation and then performing the linear relaxation (assuming the same big-M parameters are used). Therefore, the MILP relaxation of the (BM) reformulation of a GDP is the same as the (BM) reformulation of the linear GDP (LGDP) relaxation of the original GDP. A tighter LGDP relaxation of a GDP means a tighter MILP relaxation of the MINLP reformulation of the GDP. Therefore, tighter LGDP relaxations of a GDP typically translate into improved solution times in the MINLP reformulation of the GDP.

In this section we present two new models for the multiperiod blending problem: a source based model ($\mathbb{SB}$) and a hybrid model between the concentration and source based models ($\mathbb{CSB}$). We first describe the new source based model ($\mathbb{SB}$). We prove that the LGDP relaxation of the source based model ($\mathbb{SB}$) is tighter than the LGDP relaxation of the split fraction model ($\mathbb{SF}$). We present computational experiments that show that it is also tighter than the LGDP relaxation of the concentration model ($\mathbb{C}$) in all tested cases. Using the key idea behind the source based model ($\mathbb{SB}$), we then present an improvement to the concentration model ($\mathbb{C}$) using redundant constraints. The resulting model ($\mathbb{CSB}$) is a hybrid between the source based model ($\mathbb{SB}$) and the concentration model ($\mathbb{C}$). We prove that the model ($\mathbb{CSB}$) has the tightest LGDP relaxation of all the models presented in this work.

3.1 Alternative Problem Formulation

If the blending network is modeled using concentrations, as in the ($\mathbb{C}$) model, or using individual flows and inventories, as in the ($\mathbb{SF}$) model, the physical insight behind the equations is to track the specifications from supply to demand. The disadvantage of these models is that, when the non-convex constraints are dropped entirely, the composition limits of the demand can be violated. For instance, if model ($\mathbb{C}$) is relaxed, total mass balances are the only equations that restrict flows and inventories. As a consequence, the streams entering the demand tanks can have any composition. Similarly, when the bilinear terms are dropped from model ($\mathbb{SF}$), the individual flows and inventories are allowed to take any value between the bounds. The drawback is that any configuration that satisfies the total mass balance in the tanks is a feasible solution for the relaxed problem, whereas most of them will be infeasible to the original problem.

Alternatively, there is the option of tracking the "sources" or "commodities" in the system, which is the insight behind the q-formulation of the pooling problem. This type of model has also been used in crude-oil scheduling problems. The idea of "following" the crudes along the network seems reasonable at the front-end of a refinery due to the specifications in the feed to the distillation columns [24, 29].

Each supply and initial inventory in the blending tanks can be considered as a different "source". For instance, if crudes A and B are being unloaded and supplied to the system, in which tanks 1 and 3 contain an initial inventory of crude C and D respectively, the blending network has a total of four different types of crudes (or sources). Following with the crude-oil scheduling example, once the crudes are mixed and right before the mixture is discharged to the distillation columns, it is possible to calculate the relative amount of each specification in the blend since the composition of the sources is known. It is not until the final mixture of crudes is fed to the distillation columns that the composition specifications are checked.

Sources are defined as the supply and the blending tanks that have initial inventory greater than zero. The new index $r \in \mathcal{R}$ denotes the set of sources in the blending network. It is defined as $\mathcal{R} = \mathcal{S} \cup \breve{\mathcal{B}}$ where $\breve{\mathcal{B}} = \{b \in \mathcal{B} : I_b^0 > 0\}$. The variables in the model resemble the ones in the ($\mathbb{SF}$) model, but note that now $\tilde{F}_{rnn't}$ and $\tilde{I}_{rbt}$ are individual flows and inventories per source r instead of per specification q. Also, the model involves new parameters $\hat{C}_{qr}^0$ that represent the amount of specification q in source r and are defined as follows:

$$\hat{C}_{qs}^0 = C_{qs}^{\mathrm{IN}} \qquad \forall\, s \in \mathcal{S} \tag{21a}$$

$$\hat{C}_{qb}^0 = C_{qb}^0 \qquad \forall\, b \in \breve{\mathcal{B}} \tag{21b}$$

The source based model ($\mathbb{SB}$) is similar to the split fraction model ($\mathbb{SF}$), but the sources are tracked instead of the specifications. In the ($\mathbb{SF}$) model, the fraction of specification q in a stream is defined as the amount of flow of specification q in the stream, divided by the total flow between tanks, see (22a). In model ($\mathbb{SB}$), the composition of a stream is determined from the compositions of each of the sources present in the stream. The sum of the amount of specification q in each source corresponds to the total amount of specification q in the stream, i.e. $\bar{F}_{qbdt} = \sum_{r \in \mathcal{R}} \tilde{F}_{rbdt} \hat{C}_{qr}^0$. If divided by the total flow, the composition can be calculated as in equation (22b).

$$C_{qbt} = \frac{\bar{F}_{qbdt}}{F_{bdt}} \qquad \forall\, q \in \mathcal{Q},\ (b,d) \in \mathcal{A},\ t \in \mathcal{T} \tag{22a}$$

$$C_{qbt} = \frac{\sum_{r \in \mathcal{R}} \tilde{F}_{rbdt} \hat{C}_{qr}^0}{F_{bdt}} \qquad \forall\, q \in \mathcal{Q}, (b,d) \in \mathcal{A},\ t \in \mathcal{T} \tag{22b}$$

The source-based model ($\mathbb{SB}$), where sources $r \in \mathcal{R}$ correspond to the supply and blending tanks with initial inventory, is as follows:

($\mathbb{SB}$):

$$\max \quad \sum_{t\in\mathcal{T}} \Big[\sum_{(n,d)\in\mathcal{A}} \beta_d^T F_{ndt} - \sum_{(s,n)\in\mathcal{A}} \beta_s^T F_{snt} - \sum_{(n,n')\in\mathcal{N}} (\alpha_{nn'}^N y_{nn't} + \beta_{nn'}^N F_{nn't}) \Big] \tag{23}$$

s.t.

$$I_{st} = I_{st-1} + F_{st}^{\mathrm{IN}} - \sum_{(s,n)\in\mathcal{A}} F_{snt} \qquad \forall\, s \in \mathcal{S},\, t \in \mathcal{T} \tag{24a}$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d)\in\mathcal{A}} F_{ndt} - FD_{dt} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{24b}$$

$$F_{nn't} = \sum_{r\in\mathcal{R}} \tilde{F}_{rnn't} \qquad \forall\, n \in \mathcal{N},\, n' \in \hat{\mathcal{N}}_n,\, t \in \mathcal{T} \tag{25a}$$

$$I_{bt} = \sum_{r\in\mathcal{R}} \tilde{I}_{rbt} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{25b}$$

$$\begin{bmatrix} X_{nbt} \\ F_{nb}^{\mathrm{L}} \le F_{nbt} \le F_{nb}^{\mathrm{U}} \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \forall\, (n,b) \in \mathcal{A},\, t \in \mathcal{T} \tag{26}$$

$$\begin{bmatrix} X_{sdt} \\ F_{sd}^{\mathrm{L}} \le F_{sdt} \le F_{sd}^{\mathrm{U}} \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \\ \tilde{F}_{rsdt} = 0 \quad \forall r \in \mathcal{R} \end{bmatrix} \qquad \forall\, (s,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{27}$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^{\mathrm{L}} \le F_{bdt} \le F_{bd}^{\mathrm{U}} \\ C_{qd}^{\mathrm{L}} F_{bdt} \le \sum_{r\in R} \tilde{F}_{rbdt} \hat{C}_{qr}^0 \le C_{qd}^U F_{bdt} \quad \forall\, q \in Q \\ C_{qd}^{\mathrm{L}} I_{bt-1} \le \sum_{r\in\mathcal{R}} \tilde{I}_{rbt-1} \hat{C}_{qr}^0 \le C_{qd}^{\mathrm{U}} I_{bt-1} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \quad \forall\, (b,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{28}$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_{bt-1} + \sum_{(n,b)\in\mathcal{A}} F_{nbt} \\ \tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum_{(n,b)\in\mathcal{A}} \tilde{F}_{rnbt} \quad \forall\, r \in \mathcal{R} \end{bmatrix} \vee \begin{bmatrix} \neg YB_{bt} \\ I_{bt} = I_{bt-1} - \sum_{(b,n)\in\mathcal{A}} F_{bnt} \\ \tilde{I}_{rbt} = \tilde{I}_{rbt-1} - \sum_{(b,n)\in\mathcal{A}} \tilde{F}_{rbnt} \quad \forall\, r \in \mathcal{R} \\ F_{bnt} = \xi_{bnt} I_{bt-1} \\ \tilde{F}_{rbnt} = \xi_{bnt} \tilde{I}_{rbt-1} \qquad \forall\, r \in \mathcal{R} \end{bmatrix} \forall\, b \in \mathcal{B}, t \in \mathcal{T}$$

$$(29)$$

$$X_{nbt} \Rightarrow YB_{bt} \qquad \forall\,(n,b) \in \mathcal{A},\; t \in \mathcal{T} \tag{30a}$$

$$X_{bnt} \Rightarrow \neg YB_{bt} \qquad \forall\,(b,n) \in \mathcal{A},\; t \in \mathcal{T} \tag{30b}$$

$$I_n^{\mathrm{L}} \le I_{nt} \le I_n^{\mathrm{U}} \qquad \forall\, n \in \mathcal{N},\; t \in \mathcal{T} \tag{31a}$$

$$F_{nn'}^{\mathrm{L}} \le F_{nn't} \le F_{nn'}^{\mathrm{U}} \qquad \forall\,(n,n') \in \mathcal{A},\; t \in \mathcal{T} \tag{31b}$$

$$FD_{dt}^{\mathrm{L}} \le FD_{dt} \le FD_{dt}^{\mathrm{U}} \qquad \forall\, d \in \mathcal{D},\; t \in \mathcal{T} \tag{31c}$$

$$I_b^{\mathrm{L}} \le \tilde{I}_{rbt} \le I_b^{\mathrm{U}} \qquad \forall\, r \in \mathcal{R},\; b \in \mathcal{B},\; t \in \mathcal{T} \tag{31d}$$

$$F_{nn'}^{\mathrm{L}} \le \tilde{F}_{rnn't} \le F_{nn'}^{\mathrm{U}} \qquad \forall\, r \in \mathcal{R},\; (n,n') \in \mathcal{A},\; t \in \mathcal{T} \tag{31e}$$

$$0 \le \xi_{bnt} \le 1 \qquad \forall\,(b,n) \in \mathcal{A},\; t \in \mathcal{T} \tag{31f}$$

$$\tilde{F}_{rsnt}|_{r=s} = F_{snt} \qquad \forall\,(s,n) \in \mathcal{A},\; t \in \mathcal{T} \tag{32a}$$

$$\tilde{F}_{rbnt}|_{r=b} = F_{bnt} \qquad \forall\,(b,n) \in \mathcal{A},\; t = 1 \tag{32b}$$

$$X_{nn't} \in \{True, False\} \qquad \forall\,(n,n') \in \mathcal{A},\; t \in \mathcal{T} \tag{33a}$$

$$YB_{bt} \in \{True, False\} \qquad \forall\, b \in \mathcal{B},\; t \in \mathcal{T} \tag{33b}$$

The source based model ($\mathbb{SB}$) follows the same general idea as the split fraction model ($\mathbb{SF}$). However, there are four main differences. The first one is that the individual flows and inventories are based on sources $r \in \mathcal{R}$ instead of specifications $q \in \mathcal{Q}$. The second difference are the constraints (25). These constraints relate the source flows and inventories to the total flows and inventories, and they assume linear blending. Note that (25) is redundant for the GDP, however, it is not redundant for its LGDP relaxation. Also note that similar constraints cannot be included in the split fraction model ($\mathbb{SF}$), since the specifications can represent completely different properties (e.g. density and concentration of sulfur). The third difference is disjunction (28). In this disjunction, the bounds on the different specifications $q \in \mathcal{Q}$ for the demand, are transformed into restrictions for the sources $r \in \mathcal{R}$. This transformation is easily performed using the equations presented in (22). The last difference lies in equations (32). These equations link the supply and initial inventories with the corresponding individual flow per source. For instance, supply tank 1 holds source 1 and nothing else.

The LGDP relaxation of the source based model ($\mathbb{SB}$) is tighter than the LGDP relaxation of the split fraction model ($\mathbb{SF}$), as shown in the following theorem.

Theorem 3.1 *Let ($\mathbb{R}-\mathbb{SF}$) and ($\mathbb{R}-\mathbb{SB}$) be, respectively, an LGDP relaxation of ($\mathbb{SF}$) and ($\mathbb{SB}$) in which the nonlinear constraints are removed from the problem formulation. Then ($\mathbb{R}-\mathbb{SB}$) $\subseteq$ ($\mathbb{R}-\mathbb{SF}$).*

Proof. Let $(I_{nt}, F_{nn't}, FD_{dt}, \tilde{I}_{rbt}, \tilde{F}_{rnn't}, X_{nn't}, YB_{bt})$ be a feasible point in $(\mathbb{R}{-}\mathbb{SB})$. Let $\bar{I}_{qbt} = \sum_{r\in\mathcal{R}} \tilde{I}_{rbt}\hat{C}^0_{qr}$ and $\bar{F}_{qnn't} = \sum_{r\in\mathcal{R}} \tilde{F}_{rnn't}\hat{C}^0_{qr}$.

If $X_{nbt} = False$, then $\tilde{F}_{rnbt} = 0 \quad \forall r \in \mathcal{R}$. Then, for every $q \in \mathcal{Q}$ it is possible to multiply both sides of the equation by $\hat{C}^0_{qr}$:

$$\hat{C}^0_{qr}\tilde{F}_{rnbt} = 0 \quad \forall r \in \mathcal{R}, q \in \mathcal{Q} \tag{34}$$

By summing over all sources:

$$\sum_{r\in\mathcal{R}} \hat{C}^0_{qr}\tilde{F}_{rnbt} = 0 \quad \forall q \in \mathcal{Q} \tag{35}$$

$$\bar{F}_{qnbt} = 0 \quad \forall q \in \mathcal{Q} \tag{36}$$

The same scheme can be used when $X_{sdt} = False$, $X_{bdt} = False$ to obtain $\bar{F}_{qndt} = 0 \quad \forall q \in \mathcal{Q}$. For the source inventory balance constraint (associated with the Boolean variable YB_{bt}) the same two steps can be applied.

If $YB_{bt} = True$, then $\tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum_{(n,b)\in\mathcal{A}} \tilde{F}_{rnbt} \; \forall\, r \in \mathcal{R}$, which implies:

$$\bar{I}_{qbt} = \bar{I}_{qbt-1} + \sum_{(n,b)\in\mathcal{A}} \bar{F}_{qnbt} \; \forall\, q \in \mathcal{Q} \tag{37}$$

where $\bar{F}_{qsbt} = F_{sbt}C^{IN}_{qs} \; (s, b) \in \mathcal{A}$

If $YB_{bt} = False$, then $\tilde{I}_{rbt} = \tilde{I}_{rbt-1} - \sum_{(b,n)\in\mathcal{A}} \tilde{F}_{rbnt} \; \forall\, r \in \mathcal{R}$, and then:

$$\bar{I}_{qbt} = \bar{I}_{qbt-1} - \sum_{(b,n)\in\mathcal{A}} \bar{F}_{qbnt} \; \forall\, q \in \mathcal{Q} \tag{38}$$

When $X_{bdt} = True$, then $C^{\mathrm{L}}_{qd}F_{bdt} \leq \sum_{r\in\mathcal{R}} \tilde{F}_{rbdt}\hat{C}^0_{qr} \leq C^U_{qd}F_{bdt} \quad \forall\, q \in Q$, so:

$$C^{\mathrm{L}}_{qd}F_{bdt} \leq \bar{F}_{qbdt} \leq C^U_{qd}F_{bdt} \quad \forall\, q \in Q \tag{39}$$

The same procedure can be applied to obtain valid upper and lower bounds for the variables.

It is clear then, considering constraints (36), (37), (38) and (39), that for a feasible point $(I_{nt}, F_{nn't}, FD_{dt}, \tilde{I}_{rbt}, \tilde{F}_{rnn't}, X_{nn't}, YB_{bt})$ in $(\mathbb{R}{-}\mathbb{SB})$ it is possible to set $\bar{I}_{qbt} = \sum_{r\in\mathcal{R}} \tilde{I}_{rbt}\hat{C}^0_{qr}$ and $\bar{F}_{qnn't} = \sum_{r\in\mathcal{R}} \tilde{F}_{rnn't}\hat{C}^0_{qr}$ and obtain the point $(I_{nt}, F_{nn't}, FD_{dt}, \bar{I}_{rbt}, \bar{F}_{rnn't}, X_{nn't}, YB_{bt})$ that is feasible for $(\mathbb{R}{-}\mathbb{SF})$. This means that any $(I_{nt}, F_{nn't}, FD_{dt}, X_{nn't}, YB_{bt})$ that is feasible for $(\mathbb{R}{-}\mathbb{SB})$ is also feasible for $(\mathbb{R}{-}\mathbb{SF})$ $\square$.

Table 8: Comparison between the LGDP relaxation of different formulations.

			# Variables			**# Constraints**			**Normalized relaxation**		
Ex.	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{Q}\rvert$	($\mathbb{C}$)	($\mathbb{SF}$)	($\mathbb{SB}$)	($\mathbb{C}$)	($\mathbb{SF}$)	($\mathbb{SB}$)	($\mathbb{C}$)	($\mathbb{SF}$)	($\mathbb{SB}$)
1	2	1	681	889	793	1558	2582	2054	1.007	1.020^i	1.000^*
2	2	5	1385	1849	793	4342	7158	2822	1.012	1.026^i	1.000^*
3	2	10	2265	3049	793	7846	12902	3782	1.012^i	1.026^i	1.000^*
4	5	1	681	889	2713	1558	2582	5894	1.006^i	1.006	1.000^*
5	5	5	1385	1849	2713	4342	7158	6662	1.022	1.022	1.011
6	5	10	2265	3049	2713	7846	12902	7622	1.022^i	1.022^i	1.006
7	10	1	681	889	1513	1558	2582	3494	1.000^*	1.000^*	1.000^*
8	10	5	1385	1849	1513	4342	7158	4262	1.005	1.005	1.005
9	10	10	2265	3049	1513	7846	12902	5222	1.005	1.005	1.005

Note that Theorem 3.1 considers the linear relaxations ($\mathbb{R}-\mathbb{SB}$) and ($\mathbb{R}-\mathbb{SF}$) without the McCormick envelopes. The relaxations using the McCormick envelopes depend on the bounds of $\tilde{I}_{rbt}$ and $\hat{I}_{rbt}$.

Table 8 compares the value of the objective function of the LGDP relaxation of ($\mathbb{C}$), ($\mathbb{SF}$) and ($\mathbb{SB}$) for 9 instances. The LGDP relaxation was obtained using McCormick envelopes. All instances have 240 binary variables. The solutions were obtained using CPLEX 12.6. All values reported were below 1% gap after 1800 seconds of computational time. Values are normalized to the best known feasible solution. An asterisk $*$ marks the instances in which the value of the Boolean variables in the LGDP relaxation is the same as their value in the optimal solution to the GDP. i indicates those relaxed solution that will lead to an infeasible subproblem when the set of Boolean variables YB_{bt} is fixed accordingly. Instances with 1, 5 and 10 specifications and 1, 5 and 10 sources are used for the comparison. All the instances have 6 time periods and same network topology as the motivating example. Three conclusions can be inferred from the results:

1. For the examples tested, ($\mathbb{SB}$) is stronger than ($\mathbb{C}$) and ($\mathbb{SF}$) when the bilinear terms in the source-based model are relaxed using McCormick envelopes.

2. The difference between relaxations is larger when the number of specifications is high and the number of sources is low.

3. In general, the size of the relaxed ($\mathbb{C}$) formulation strongly depends on the number of specifications, whereas the size of the relaxed ($\mathbb{SB}$) model depends on the number of sources.

The difference between the values of the normalized relaxations may not seem that significant at first. However, the feasibility of the subproblem when the set of discrete variables YB_{bt} is fixed according to the solution of the relaxed LGDP is crucial for the decomposition algorithm. In 6 of the 9 examples, the upper bound provided by the relaxation of the source based model ($\mathbb{SB}$) is the same as the best known solution. Furthermore, when fixing the Boolean variables YB_{bt} form the LGDP relaxation of ($\mathbb{SB}$), all solutions are feasible to the original problem. In the concentration ($\mathbb{C}$) and split fraction ($\mathbb{SF}$) models 3 and 4 instances become infeasible, when fixing the value of YB_{bt} from the LGDP relaxation.

Note that the q and pq-formulations are also exploiting the idea of sources or commodities to model the blending process. Even though the ideas are similar, these formulation have clear differences with the source-based model. The proportion variables in the q-formulation denote the fraction that each source contributes to the total incoming flow to the blending tank, which implies that the sum of the fractions over all sources add to 1. Instead of following the fraction of the total flow that corresponds to each source, the source-based model tracks the actual amount of source in each and every stream in the system. Also, the source-based model uses splits fractions in order to ensure consistency in the discharge. This is not necessary in the traditional q-formulation, since the pooling problem does not consider inventories. Gupte et al. [38] proposed an extension of the q-formulation to handle inventories and semi-continuous flows. However, their model requires the introduction of more bilinear terms with up to five indexes per term. This implies that the number of bilinear terms will increase drastically even with small instances. Finally, the classical Haverly pooling problem is used to illustrate the difference between the traditional formulations in the pooling community and the new formulation presented in this report. See Appendix A for details.

3.2 Using redundant constraints in the ($\mathbb{C}$) model

The linear constraints of the source based model ($\mathbb{SB}$) can be used as redundant constraints in the concentration model ($\mathbb{C}$). This allows to obtain stronger LGDP relaxations. The new model ($\mathbb{CSB}$) (hybrid of the ($\mathbb{C}$) and ($\mathbb{SB}$) models) will increase in size but will have a stronger LGDP relaxation. The model is as follows:

($\mathbb{CSB}$):

$$\max \quad \sum_{t\in\mathcal{T}} \Big[\sum_{(n,d)\in\mathcal{A}} \beta_d^T F_{ndt} - \sum_{(s,n)\in\mathcal{A}} \beta_s^T F_{snt} - \sum_{(n,n')\in\mathcal{A}} (\alpha_{nn'}^N x_{nn't} + \beta_{nn'}^N F_{nn't}) \Big] \tag{40}$$

s.t.

$$I_{st} = I_{st-1} + F_{st}^{\mathrm{IN}} - \sum_{(s,n)\in\mathcal{A}} F_{snt} \qquad \forall\, s\in\mathcal{S},\, t\in\mathcal{T} \tag{41a}$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d)\in\mathcal{A}} F_{ndt} - FD_{dt} \qquad \forall\, d\in\mathcal{D},\, t\in\mathcal{T} \tag{41b}$$

$$F_{nn't} = \sum_{r\in\mathcal{R}} \tilde{F}_{rnn't} \qquad \forall\, (n,n')\in\mathcal{A},\, t\in\mathcal{T} \tag{42a}$$

$$I_{bt} = \sum_{r\in\mathcal{R}} \tilde{I}_{rbt} \qquad \forall\, b\in\mathcal{B},\, t\in\mathcal{T} \tag{42b}$$

$$\begin{bmatrix} X_{nbt} \\ F_{nb}^{\mathrm{L}} \le F_{nbt} \le F_{nb}^{\mathrm{U}} \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \forall\, (n,b)\in\mathcal{A},\, t\in\mathcal{T} \tag{43}$$

$$\begin{bmatrix} X_{sdt} \\ F^{\mathrm{L}}_{sd} \leq F_{sdt} \leq F^{\mathrm{U}}_{sd} \\ C^{\mathrm{L}}_{qd} \leq C^{\mathrm{IN}}_{qs} \leq C^{\mathrm{U}}_{qd} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \end{bmatrix} \qquad \forall\,(s,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{44}$$

$$\begin{bmatrix} X_{bdt} & \\ F^{\mathrm{L}}_{bd} \leq F_{bdt} \leq F^{\mathrm{U}}_{bd} & \\ C^{\mathrm{L}}_{qd} \leq C_{qbt-1} \leq C^{\mathrm{U}}_{qd} & \forall\, q \in \mathcal{Q} \\ C^{\mathrm{L}}_{qd} F_{bdt} \leq \sum_{r \in \mathcal{R}} \tilde{F}_{rbdt} \hat{C}^0_{qr} \leq C^{\mathrm{U}}_{qd} F_{bdt} & \forall\, q \in \mathcal{Q} \\ C^{\mathrm{L}}_{qd} I_{bt-1} \leq \sum_{r \in \mathcal{R}} \tilde{I}_{rbt-1} \hat{C}^0_{qr} \leq C^{\mathrm{U}}_{qd} I_{bt-1} & \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \quad \forall\,(b,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{45}$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_{bt-1} + \sum_{(n,b) \in \mathcal{A}} F_{nbt} \\ I_{bt} C_{qbt} = I_{bt-1} C_{qbt-1} + \sum_{(s,b) \in \mathcal{A}} F_{sbt} C^{\mathrm{IN}}_{qs} \\ \quad + \sum_{(b',b) \in \mathcal{A}} F_{b'bt} C_{qb't-1} \quad \forall\, q \in \mathcal{Q} \\ \tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum_{(n,b) \in \mathcal{A}} \tilde{F}_{rnbt} \quad \forall\, r \in \mathcal{R} \end{bmatrix} \vee \begin{bmatrix} \neg YB_{bt} & \\ I_{bt} = I_{bt-1} - \sum_{(b,n) \in \mathcal{A}} F_{bnt} & \\ C_{qbt} = C_{qbt-1} & \forall\, q \in \mathcal{Q} \\ \tilde{I}_{rbt} = \tilde{I}_{rbt-1} - \sum_{(b,n) \in \mathcal{A}} \tilde{F}_{rbnt} & \forall\, r \in \mathcal{R} \end{bmatrix} \quad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{46}$$

$$X_{nbt} \Rightarrow YB_{bt} \qquad \forall\, b \in \mathcal{B},\, n \in \check{\mathcal{N}}_b,\, t \in \mathcal{T} \tag{47a}$$

$$X_{bnt} \Rightarrow \neg YB_{bt} \qquad \forall\, b \in \mathcal{B},\, n \in \hat{\mathcal{N}}_b,\, t \in \mathcal{T} \tag{47b}$$

$$I_n^{\mathrm{L}} \leq I_{nt} \leq I_n^{\mathrm{U}} \qquad \forall\, n \in \mathcal{N},\, t \in \mathcal{T} \tag{48a}$$

$$F_{nn'}^{\mathrm{L}} \leq F_{nn't} \leq F_{nn'}^{\mathrm{U}} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{48b}$$

$$FD_{dt}^{\mathrm{L}} \leq FD_{dt} \leq FD_{dt}^{\mathrm{U}} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{48c}$$

$$C_q^{\mathrm{L}} \leq C_{qbt} \leq C_q^{\mathrm{U}} \qquad \forall\, q \in \mathcal{Q},\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{48d}$$

$$I_b^{\mathrm{L}} \leq \tilde{I}_{rbt} \leq I_b^{\mathrm{U}} \qquad \forall\, r \in \mathcal{R},\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{48e}$$

$$F_{nn'}^{\mathrm{L}} \leq \tilde{F}_{rnn't} \leq F_{nn'}^{\mathrm{U}} \qquad \forall\, r \in \mathcal{R},\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{48f}$$

$$\tilde{F}_{rsnt}|_{r=s} = F_{snt} \qquad \forall\, (s,n) \in \mathcal{A},\, t \in \mathcal{T} \tag{49a}$$

$$\tilde{F}_{rbnt}|_{r=b} = F_{bnt} \qquad \forall\, (b,n) \in \mathcal{A},\, t = 1 \tag{49b}$$

$$X_{nn't} \in \{True, False\} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{50a}$$

$$YB_{bt} \in \{True, False\} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{50b}$$

In addition to the constraints in the concentration model ($\mathbb{C}$), ($\mathbb{CSB}$) includes the last two inequalities in the first term of the disjunction (45), the last equations in disjunction (46), and equalities (42) and (49). Note that all of these inequalities are linear.

Consider the concentration model ($\mathbb{C}$), the source based model ($\mathbb{SB}$), and the hybrid model ($\mathbb{CSB}$). The LGDP relaxation of ($\mathbb{CSB}$) is tighter than the LGDP relaxation of the other two, as stated in the following theorem.

Theorem 3.2 *Let ($\mathbb{R}-\mathbb{C}$), ($\mathbb{R}-\mathbb{SB}$) and ($\mathbb{R}-\mathbb{CSB}$) be, respectively, a linear relaxation of ($\mathbb{C}$), ($\mathbb{SB}$) and ($\mathbb{CSB}$) in which the nonlinear constraints are removed from the problem formulation. Then ($\mathbb{R}-\mathbb{CSB}$) $\subseteq$ ($\mathbb{R}-\mathbb{SB}$) and ($\mathbb{R}-\mathbb{CSB}$) $\subseteq$ ($\mathbb{R}-\mathbb{C}$).*

The proof of Theorem 3.2 is trivial, since ($\mathbb{R}-\mathbb{CSB}$) includes all of the constraints of ($\mathbb{R}-\mathbb{C}$) and ($\mathbb{R}-\mathbb{SB}$).

In summary, we have presented four formulations in this work: the concentration model ($\mathbb{C}$), the split fraction model ($\mathbb{SF}$), the source based model ($\mathbb{SB}$), and the hybrid model ($\mathbb{CSB}$). We can stablish the following relations between the LGDP relaxation of these models: ($\mathbb{R}-\mathbb{CSB}$) $\subseteq$ ($\mathbb{R}-\mathbb{SB}$) $\subseteq$ ($\mathbb{R}-\mathbb{SF}$), and ($\mathbb{R}-\mathbb{CSB}$) $\subseteq$ ($\mathbb{R}-\mathbb{C}$). Therefore, when removing the nonlinear constraints from the formulations, ($\mathbb{CSB}$) is stronger than the other formulations. Note that a linear relaxation of the different formulations can be achieved by using McCormick[8] envelopes of the bilinear terms. In such a case, the strength of the linear relaxation also depends on the bounds of the variables involved in the bilinear terms. In real applications, it is likely that the bounds for total flow and concentration are stronger than the bounds for individual specification inventories and split fractions. In such cases, the advantage of ($\mathbb{CSB}$) over ($\mathbb{R}-\mathbb{SB}$), and ($\mathbb{R}-\mathbb{SF}$) is further increased.

The number of bilinear terms in ($\mathbb{CSB}$) is the same as in ($\mathbb{C}$). The number of bilinear terms in ($\mathbb{SB}$) depends not only on $\mathcal{Q}$, $\mathcal{T}$ and $\mathcal{B}$, but also on $\mathcal{S}$ and the number of blending tanks with $I_b^0 > 0$. Table 9 presents the number of bilinear terms for ($\mathbb{CSB}$) and for ($\mathbb{SB}$) for two instances. Both instances have the same topology and $|\mathcal{Q}| = 5$ and $|\mathcal{T}| = 6$. However, the initial inventory of all the blending tanks in the first instance is zero. The initial inventory of all blending tanks in the second instance is greater than zero. It is clear from Table 9 that the number of bilinear terms for the ($\mathbb{SB}$) can change drastically for "similar" instances (480 vs.1760).

Table 9: Number of bilinear terms of GDP formulations. $\hat{B} = (b, b') \in \mathcal{A}$, $\hat{N}_b = (b, n) \in \mathcal{A}$

Model	Bilinear terms	Motivating Example $\lvert\mathcal{Q}\rvert = 5$, $\lvert\mathcal{T}\rvert = 6$ $I_b^0 = 0$, $\lvert\mathcal{R}\rvert = 2$	$\lvert\mathcal{Q}\rvert = 5$, $\lvert\mathcal{T}\rvert = 6$ $I_b^0 > 0$, $\lvert\mathcal{R}\rvert = 10$
($\mathbb{CSB}$)	$\lvert\mathcal{Q}\rvert \left[\lvert\mathcal{B}\rvert\lvert\mathcal{T}\rvert + \lvert\hat{\mathcal{B}}\rvert(\lvert\mathcal{T}\rvert - 1) \right]$	640	640
($\mathbb{SB}$)	$\lvert\hat{\mathcal{N}}_b\rvert(\mathcal{T} - 1)(1 + \lvert\mathcal{R}\rvert)$	480	1760

The MINLP reformulation of the concentration model, with and without redundant constraints (($\mathbb{C}$), ($\mathbb{CSB}$)) was tested in 48 instances. Half of these instances include initial inventory and the other half do not (See Section 5 for more details on the instances). Table 10 shows the fraction of the instances for which the solver could find at least one feasible solution. The global solvers BARON 14.0, ANTIGONE 1.1, and SCIP 3.1 were used.

Table 10: Fraction of instances for which a feasible solution was found in less than 30 minutes.

Solver	($\mathbb{CSB}$)	($\mathbb{C}$)
SCIP	**0.42**	0.31
BARON	0.29	0.21
ANTIGONE	0.31	0.29

In general, a feasible solution is obtained for a larger number of instances if the problem is modeled using the redundant constraints. For instance, in the case of SCIP, the number of instances for which SCIP can find a solution increase from 15 to 20 out of 48 instances. In addition, SCIP performs better than its competitors, since it can find a feasible solution in 42% of the instances against the 29% and 31% of BARON and ANTIGONE, respectively. Due to this superior performance, SCIP 3.1 is used as a reference for comparison in the computational results.

It can be seen that the performance of the solver is better when the redundant constraints are added to the concentration model. Nevertheless, the number of instances for which a feasible solution was found is still small. This motivates the need to develop a specialized algorithm that can better exploit the structure of the problem.

4 A Two-Stage MILP-MINLP Decomposition Algorithm

Considering the performance of commercial solvers and the potential advantages of the ($\mathbb{CSB}$) formulation, a decomposition algorithm is proposed next. As mentioned before, if the operating mode of the blending tanks is fixed, the resulting GDP becomes easier to solve, due to a reduction in size and complexity. In fact, all variables related to incoming arcs to a blending tank (i.e. F_{nbt} and x_{nbt}) will be removed from the model when the tank is discharging. Similarly, if the tank is being charged, all outgoing flows from the blending tank (i.e. F_{bnt} and x_{bnt}) will be set to zero. Furthermore, the number of bilinear terms will decrease compared to the original GDP in the following circumstances:

1. If the blending tank is discharging at time t, the equations that describe the operation are linear for that period (i.e., the second disjunct of disjunction (46) is True). Thus, all bilinearities related to that blending tank and time period are eliminated from the model. In addition, if a blending tank is in idle mode, it can be set to discharge mode in order to avoid considering unnecessary bilinear terms.

2. The bilinear term $F_{b'bt}C_{qb't-1}$ has to be included if and only if tank b' is discharging ($\neg YB_{b't}$) and tank b is charging (YB_{bt}) at time t. Therefore, if blending tank b has an incoming stream from a supply tank, i.e. it is in charge mode, but there is no other blending tank (b'), connected to tank b, that is discharging at that time t, bilinear terms of the form $F_{b'bt}C_{qb't-1}$ are unnecessary and can be eliminated.

To exploit these ideas, the proposed algorithm decomposes the GDP model into two levels. The first level, or master problem, is a linear relaxation of the original GDP that provides rigorous upper bounds for the profit. The second level, or subproblem, is a smaller GDP in which the set of discrete variables YB_{bt} is fixed. The subproblem, when a feasible solution is found, provides a feasible solution to the original GDP and a rigorous lower bound. These problems are solved successively until the gap between the upper and lower bounds is within a tolerance. Figure 5 presents the flow diagram of the algorithm.

The solution of the master problem is used to define the subproblem, which is more tractable than the original problem. A master problem with a tight relaxation is crucial for the success of the algorithm, since its solution will be used to fixed the operating mode of the tanks. The feasibility of the subproblem will strictly depend on the solution of the master problem.

As mentioned in the introduction, there are many relaxation techniques that can be used to construct the master problem. In the algorithm, the master problem is a linear relaxation of ($\mathbb{CSB}$) in which the non-convex constraints are dropped. Optimality and/or feasibility cuts are added in the form of integer cuts, eliminating regions already evaluated in previous iterations. Note that McCormick envelopes could be used for linearly relaxing ($\mathbb{CSB}$). However, from computational experiments we observed that dropping the nonlinearities improved the performance of the algorithm. In particular, the master problem solves faster, and we did not observe a significant difference in the number of iterations of the algorithm. We acknowledge that for other instances the use of McCormick envelopes could help the algorithm to perform better.

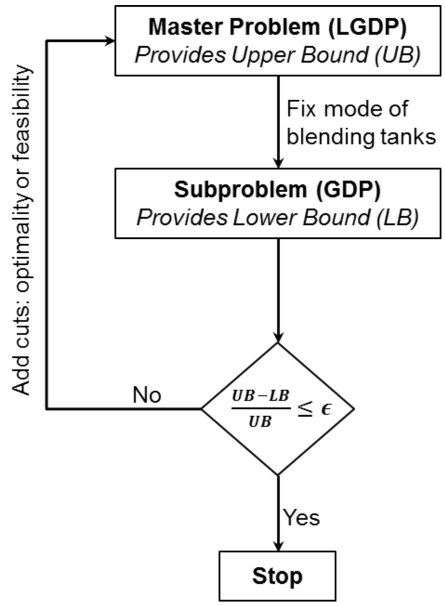

Figure 5: Decomposition Algorithm

The subproblem can be solved using a global optimization solver or through a specialized technique that ensures global optimality. The concentration model plus the source-based redundant constraints ($\mathbb{CSB}$) is used in the subproblem.

4.1 Description of the algorithm

The following master problem is a linear relaxation of the ($\mathbb{CSB}$) in which the nonlinear constraints were dropped. Also, optimality and/or feasibility cuts are added in the form of integer cuts:

($\mathbb{MP}$):

$$\max \quad Z \tag{51}$$

s.t.

$$Z \leq \sum_{t \in \mathcal{T}} \Big[\sum_{(n,d) \in \mathcal{A}} \beta_d^T F_{ndt} - \sum_{(s,n) \in \mathcal{A}} \beta_s^T F_{snt} - \sum_{(n,n') \in \mathcal{N}} (\alpha_{nn'}^N y_{nn't} + \beta_{nn'}^N F_{nn't}) \Big] \tag{52}$$

$$I_{st} = I_{st-1} + F_{st}^{\mathrm{IN}} - \sum_{(s,n) \in \mathcal{A}} F_{snt} \qquad \forall\, s \in \mathcal{S},\, t \in \mathcal{T} \tag{53a}$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d) \in \mathcal{A}} F_{ndt} - FD_{dt} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{53b}$$

$$F_{nn't} = \sum_{r\in\mathcal{R}} \tilde{F}_{rnn't} \qquad \forall\, n \in \mathcal{N},\, n' \in \hat{\mathcal{N}}_n,\, t \in \mathcal{T} \tag{54a}$$

$$I_{bt} = \sum_{r\in\mathcal{R}} \tilde{I}_{rbt} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{54b}$$

$$\begin{bmatrix} X_{nbt} \\ F^{\mathrm{L}}_{nb} \leq F_{nbt} \leq F^{\mathrm{U}}_{nb} \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \forall\,(n,b) \in \mathcal{A},\, t \in \mathcal{T} \tag{55}$$

$$\begin{bmatrix} X_{sdt} \\ F^{\mathrm{L}}_{sd} \leq F_{sdt} \leq F^{\mathrm{U}}_{sd} \\ C^{\mathrm{L}}_{qd} \leq C^{\mathrm{IN}}_{qs} \leq C^{\mathrm{U}}_{qd} \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \end{bmatrix} \qquad \forall\,(s,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{56}$$

$$\begin{bmatrix} X_{bdt} & \\ F^{\mathrm{L}}_{bd} \leq F_{bdt} \leq F^{\mathrm{U}}_{bd} & \\ C^{\mathrm{L}}_{qd} \leq C_{qbt-1} \leq C^{\mathrm{U}}_{qd} & \forall\, q \in \mathcal{Q} \\ C^{\mathrm{L}}_{qd} F_{bdt} \leq \sum\limits_{r\in\mathcal{R}} \tilde{F}_{rbdt} \hat{C}^{0}_{qr} \leq C^{\mathrm{U}}_{qd} F_{bdt} & \forall\, q \in \mathcal{Q} \\ C^{\mathrm{L}}_{qd} I_{bt-1} \leq \sum\limits_{r\in\mathcal{R}} \tilde{I}_{rbt-1} \hat{C}^{0}_{qr} \leq C^{\mathrm{U}}_{qd} I_{bt-1} & \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \forall\,(b,d) \in \mathcal{A},\, t \in \mathcal{T} \tag{57}$$

$$\begin{bmatrix} YB_{bt} & \\ yb_{bt} = 1 & \\ I_{bt} = I_{bt-1} + \sum\limits_{(n,b)\in\mathcal{A}} F_{nbt} & \\ \tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum\limits_{(n,b)\in\mathcal{A}} \tilde{F}_{rnbt} & \forall\, r \in \mathcal{R} \end{bmatrix} \vee \begin{bmatrix} \neg YB_{bt} & \\ yb_{bt} = 0 & \\ I_{bt} = I_{bt-1} - \sum\limits_{(b,n)\in\mathcal{A}} F_{bnt} & \\ C_{qbt} = C_{qbt-1} & \forall\, q \in \mathcal{Q} \\ \tilde{I}_{rbt} = \tilde{I}_{rbt-1} - \sum\limits_{(b,n)\in\mathcal{A}} \tilde{F}_{rbnt} & \forall\, r \in \mathcal{R} \end{bmatrix} \forall\, b \in \mathcal{B}, t \in \mathcal{T} \tag{58}$$

$$\begin{aligned} Z \leq -(UB - Z^i)\Big(\sum_{\substack{b\in B,\, t\in T: \\ \hat{yb}^{i}_{bt}=1}} yb_{bt} - \sum_{\substack{b\in B,\, t\in T: \\ \hat{yb}^{i}_{bt}=0}} yb_{bt} \Big) & \\ +(UB - Z^i)\Big(\sum_{b\in B,\, t\in T} (\hat{yb}^{i}_{bt}) - 1 \Big) + UB & \qquad \forall\, i \in \mathcal{I}_O \end{aligned} \tag{59a}$$

$$\sum_{\substack{b\in B,\, t\in T: \\ \hat{yb}^{i}_{bt}=1}} (1 - yb_{bt}) + \sum_{\substack{b\in B,\, t\in T: \\ \hat{yb}^{i}_{bt}=0}} yb_{bt} \geq 1 \qquad \forall\, i \in \mathcal{I}_F \tag{59b}$$

$$X_{nbt} \Rightarrow YB_{bt} \qquad \forall\,(n,b) \in \mathcal{A},\, t \in \mathcal{T} \tag{60a}$$

$$X_{bnt} \Rightarrow \neg YB_{bt} \qquad \forall\,(b,n) \in \mathcal{A},\, t \in \mathcal{T} \tag{60b}$$

$$I_n^{\mathrm{L}} \leq I_{nt} \leq I_n^{\mathrm{U}} \qquad \forall\, n \in \mathcal{N},\, t \in \mathcal{T} \tag{61a}$$

$$F_{nn'}^{\mathrm{L}} \leq F_{nn't} \leq F_{nn'}^{\mathrm{U}} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{61b}$$

$$FD_{dt}^{\mathrm{L}} \leq FD_{dt} \leq FD_{dt}^{\mathrm{U}} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{61c}$$

$$I_b^{\mathrm{L}} \leq \tilde{I}_{rbt} \leq I_b^{\mathrm{U}} \qquad \forall\, r \in \mathcal{R},\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{61d}$$

$$F_{nn'}^{\mathrm{L}} \leq \tilde{F}_{rnn't} \leq F_{nn'}^{\mathrm{U}} \qquad \forall\, r \in \mathcal{R},\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{61e}$$

$$0 \leq \xi_{bnt} \leq 1 \qquad \forall\, (b,n) \in \mathcal{A},\, t \in \mathcal{T} \tag{61f}$$

$$\tilde{F}_{rsnt}|_{r=s} = F_{snt} \qquad \forall\, (s,n) \in \mathcal{A},\, t \in \mathcal{T} \tag{62a}$$

$$\tilde{F}_{rbnt}|_{r=b} = F_{bnt} \qquad \forall\, (b,n) \in \mathcal{A},\, t = 1 \tag{62b}$$

$$X_{nn't} \in \{True, False\} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{63a}$$

$$YB_{bt} \in \{True, False\} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{63b}$$

Note that variable yb_{bt} is introduced in the formulation. This variable takes the value of the binary variable that corresponds to YB_{bt} in the (BM) reformulation of the GDP (i.e. $yb_{bt} = 1$, when $YB_{bt} = True$). It is necessary to introduce the variable to add the enumeration cuts (59), which are added in the form of integer cuts that eliminate regions already evaluated in previous iterations. $\mathcal{I}_F$ is the set of enumeration cuts that are added when a subproblem is infeasible[39]. $\mathcal{I}_O$ is the set of enumeration cuts that are added otherwise[40]. Z is the value of the objective function, UB a global upper bound for the GDP, and Z^i an upper bound for the objective function corresponding to the solution $\hat{yb}_{bt}^{i}$. (59b) will eliminate from the feasible space those solutions for the master problem that resulted in infeasible subproblems. When yb_{bt} is different from $\hat{yb}_{bt}^{i}$, then $\sum_{\substack{b \in B,\, t \in T:\\ \hat{yb}_{bt}^{i}=1}} yb_{bt} - \sum_{\substack{b \in B,\, t \in T:\\ \hat{yb}_{bt}^{i}=0}} yb_{bt}$ is smaller than $\sum_{b \in B,\, t \in T} (\hat{yb}_{bt}^{i})$ and (59a) becomes $Z \leq UB$ (or an even weaker cut). When $yb_{bt} = \hat{yb}_{bt}^{i}$, then $\sum_{\substack{b \in B,\, t \in T:\\ \hat{yb}_{bt}^{i}=1}} yb_{bt} - \sum_{\substack{b \in B,\, t \in T:\\ \hat{yb}_{bt}^{i}=0}} yb_{bt} = \sum_{b \in B,\, t \in T} (\hat{yb}_{bt}^{i})$. In such a case, (59a) becomes $Z \leq Z^i$ and the cut is valid (since Z^i is an upper bound of the objective function in the solution $\hat{yb}_{bt}^{i}$).

For a given $\tilde{YB}_{bt}^{\mathrm{fix}} \in \{True, False\} \forall\, b \in \mathcal{B},\, t \in \mathcal{T}$, consider the following subproblem (which is the ($\mathbb{CSB}$) model with tanks fixed in "charge" or "discharge" mode):

($\mathbb{SP}$):

$$\max \quad \sum_{t \in \mathcal{T}} \Big[\sum_{(n,d) \in \mathcal{A}} \beta_d^T F_{ndt} - \sum_{(s,n) \in \mathcal{A}} \beta_s^T F_{snt} - \sum_{(n,n') \in \mathcal{A}} (\alpha_{nn'}^N x_{nn't} + \beta_{nn'}^N F_{nn't}) \Big] \tag{64}$$

s.t.

$$I_{st} = I_{st-1} + F^{\mathrm{IN}}_{st} - \sum_{(s,n)\in\mathcal{A}} F_{snt} \qquad \forall\, s \in \mathcal{S},\, t \in \mathcal{T} \tag{65a}$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d)\in\mathcal{A}} F_{ndt} - FD_{dt} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{65b}$$

$$F_{nn't} = \sum_{r\in\mathcal{R}} \tilde{F}_{rnn't} \qquad \forall\, (n, n') \in \mathcal{A},\, t \in \mathcal{T} \tag{66a}$$

$$I_{bt} = \sum_{r\in\mathcal{R}} \tilde{I}_{rbt} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{66b}$$

$$\begin{bmatrix} X_{nbt} \\ F^{\mathrm{L}}_{nb} \le F_{nbt} \le F^{\mathrm{U}}_{nb} \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \forall\, (n, b) \in \mathcal{A},\, t \in \mathcal{T}, \tilde{YB}^{\mathrm{fix}}_{bt} = True \tag{67}$$

$$\begin{bmatrix} X_{sdt} \\ F^{\mathrm{L}}_{sd} \le F_{sdt} \le F^{\mathrm{U}}_{sd} \\ C^{\mathrm{L}}_{qd} \le C^{\mathrm{IN}}_{qs} \le C^{\mathrm{U}}_{qd} \quad \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \end{bmatrix} \qquad \forall\, (s, d) \in \mathcal{A},\, t \in \mathcal{T} \tag{68}$$

$$\begin{bmatrix} X_{bdt} & \\ F^{\mathrm{L}}_{bd} \le F_{bdt} \le F^{\mathrm{U}}_{bd} & \\ C^{\mathrm{L}}_{qd} \le C_{qbt-1} \le C^{\mathrm{U}}_{qd} & \forall\, q \in \mathcal{Q} \\ C^{\mathrm{L}}_{qd} F_{bdt} \le \sum_{r\in\mathcal{R}} \tilde{F}_{rbdt} \hat{C}^{0}_{qr} \le C^{\mathrm{U}}_{qd} F_{bdt} & \forall\, q \in \mathcal{Q} \\ C^{\mathrm{L}}_{qd} I_{bt-1} \le \sum_{r\in\mathcal{R}} \tilde{I}_{rbt-1} \hat{C}^{0}_{qr} \le C^{\mathrm{U}}_{qd} I_{bt-1} & \forall\, q \in \mathcal{Q} \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \forall\, (b, d) \in \mathcal{A},\, t \in \mathcal{T}, \tilde{YB}^{\mathrm{fix}}_{bt} = False \tag{69}$$

$$I_{bt} = I_{bt-1} + \sum_{(n,b)\in\mathcal{A}} F_{nbt} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T}, \tilde{YB}^{\mathrm{fix}}_{bt} = True \tag{70a}$$

$$\begin{aligned} I_{bt} C_{qbt} = {} & I_{bt-1} C_{qbt-1} + \sum_{(s,b)\in\mathcal{A}} F_{sbt} C^{\mathrm{IN}}_{qs} \\ & + \sum_{\substack{(b',b)\in\mathcal{A} \\ \tilde{YB}^{\mathrm{fix}}_{b't} = False}} F_{b'bt} C_{qb't-1} \qquad \forall\, q \in \mathcal{Q}, r \in \mathcal{R}, b \in \mathcal{B},\, t \in \mathcal{T}, \tilde{YB}^{\mathrm{fix}}_{bt} = True \end{aligned} \tag{70b}$$

$$\tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum_{(n,b)\in\mathcal{A}} \tilde{F}_{rnbt} \qquad \forall r \in \mathcal{R}, b \in \mathcal{B},\, t \in \mathcal{T}, \tilde{YB}^{\mathrm{fix}}_{bt} = True \tag{70c}$$

$$I_{bt} = I_{bt-1} - \sum_{(b,n)\in\mathcal{A}} F_{bnt} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T}, \tilde{YB}^{\text{fix}}_{bt} = False \tag{71a}$$

$$C_{qbt} = C_{qbt-1} \qquad q \in \mathcal{Q}, b \in \mathcal{B},\, t \in \mathcal{T}, \tilde{YB}^{\text{fix}}_{bt} = False \tag{71b}$$

$$\tilde{I}_{rbt} = \tilde{I}_{rbt-1} - \sum_{(b,n)\in\mathcal{A}} \tilde{F}_{rbnt} \qquad \forall\, r \in \mathcal{R}, b \in \mathcal{B},\, t \in \mathcal{T}, \tilde{YB}^{\text{fix}}_{bt} = False \tag{71c}$$

$$F_{bdt} = 0 \qquad \forall\, (b,d) \in \mathcal{A},\, t \in \mathcal{T}, \tilde{YB}^{\text{fix}}_{bt} = True \tag{72a}$$

$$F_{nbt} = 0 \qquad \forall\, (n,b) \in \mathcal{A},\, t \in \mathcal{T}, \tilde{YB}^{\text{fix}}_{bt} = False \tag{72b}$$

$$I^{\text{L}}_{n} \leq I_{nt} \leq I^{\text{U}}_{n} \qquad \forall\, n \in \mathcal{N},\, t \in \mathcal{T} \tag{73a}$$

$$F^{\text{L}}_{nn'} \leq F_{nn't} \leq F^{\text{U}}_{nn'} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{73b}$$

$$FD^{\text{L}}_{dt} \leq FD_{dt} \leq FD^{\text{U}}_{dt} \qquad \forall\, d \in \mathcal{D},\, t \in \mathcal{T} \tag{73c}$$

$$C^{\text{L}}_{q} \leq C_{qbt} \leq C^{\text{U}}_{q} \qquad \forall\, q \in \mathcal{Q},\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{73d}$$

$$I^{\text{L}}_{b} \leq \tilde{I}_{rbt} \leq I^{\text{U}}_{b} \qquad \forall\, r \in \mathcal{R},\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{73e}$$

$$F^{\text{L}}_{nn'} \leq \tilde{F}_{rnn't} \leq F^{\text{U}}_{nn'} \qquad \forall\, r \in \mathcal{R},\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{73f}$$

$$\tilde{F}_{rsnt}|_{r=s} = F_{snt} \qquad \forall\, (s,n) \in \mathcal{A},\, t \in \mathcal{T} \tag{74a}$$

$$\tilde{F}_{rbnt}|_{r=b} = F_{bnt} \qquad \forall\, (b,n) \in \mathcal{A},\, t = 1 \tag{74b}$$

$$X_{nn't} \in \{True, False\} \qquad \forall\, (n,n') \in \mathcal{A},\, t \in \mathcal{T} \tag{75a}$$

$$YB_{bt} \in \{True, False\} \qquad \forall\, b \in \mathcal{B},\, t \in \mathcal{T} \tag{75b}$$

Note that the summation of streams that contains the bilinear terms in (70b) only involves the blending tanks that are operating as "discharge" ($\tilde{YB}^{\text{fix}}_{b't} = False$) at a given time period.

The decomposition algorithm is as follows:

0. Specify gap $\epsilon > 0$. Set $UB = \inf$, $LB = -\inf$, $i = 1$, $\mathcal{I}_O = \{\emptyset\}$, and $\mathcal{I}_F = \{\emptyset\}$;

1. Solve ($\mathbb{MP}$). Let $\tilde{YB}^{\text{fix}}_{bt}$ be the value of YB_{bt} at the optimal solution. Let $\hat{yb}^{i}_{bt}$ be the binary representation of the Boolean parameter $\tilde{YB}^{\text{fix}}_{bt}$ (i.e. if $\tilde{YB}^{\text{fix}}_{bt} = True$ then $\hat{yb}^{i}_{bt} = 1$, and if $\tilde{YB}^{\text{fix}}_{bt} = False$ then $\hat{yb}^{i}_{bt} = 0$). Let UB be the value of the optimal objective function.

2. Solve ($\mathbb{SP}$) using $\tilde{YB}^{\text{fix}}_{bt}$ with optimality gap $\epsilon_{SP} \leq \epsilon$.

If ($\mathbb{SP}$) is infeasible, let $i \in I_F$, and go to 3.

If ($\mathbb{SP}$) is feasible, let $i \in I_O$. Let Z^{i*} be the value of the optimal objective function, and Z^{i} be the upper bound of the objective function. If $Z^{i*} > LB$ then set $LB = Z^{i*}$, let

$(I^*_{nt}, F^*_{nn't}, C^*_{qbt}, \tilde{I}^*_{rbt}, \tilde{F}^*_{rnn't}, X^*_{nn't}, YB^*_{bt})$ be the optimal values of the variables in $(\mathbb{SP})$ and go to 3. If $Z^{i*} \leq LB$ go to 3.

3. If $(UB-LB)/LB \leq \epsilon$, stop with optimal solution $(I^*_{nt}, F^*_{nn't}, C^*_{qbt}, \tilde{I}^*_{rbt}, \tilde{F}^*_{rnn't}, X^*_{nn't}, YB^*_{bt})$. Else, set $i = i + 1$ and go to 1.

Theorem 4.1 *The decomposition algorithm converges to the global optimal solution, within ϵ optimality gap, after a finite number of iterations.*

Proof. Enumeration cut (59b) guarantees that infeasible solutions are not revisited again by the master problem. Cut (59a) ensures that if a feasible solution is revisited, then the UB from $(\mathbb{MP})$ equals the upper bound of $(\mathbb{SP})$ for that solution (Z^i). Since $\epsilon_{SP} \leq \epsilon$, then $(UB - LB)/LB \leq \epsilon$. $\square$.

Two phases were implemented for the algorithm. Both phases follow the same steps, but different stopping criteria. In the first phase, the stopping criteria of the master and the subproblem are the maximum execution time and the optimality gap. In the second phase, the optimality gap is the only criteria. The objective of the first phase is to quickly find feasible solutions by limiting the time limit for solving the master and subproblem. Instead of focusing on a region, the algorithm is allowed to move to the next iteration and try a different configuration of tanks after a small amount of time. In the second phase, the objective is to find the optimal solution for the problem within a tolerance. In order to guarantee global optimality, the master and the subproblem have to be solved, at least, to the specified optimality gap of the algorithm.

4.2 Illustration of the algorithm

In order to illustrate the decomposition algorithm, consider the motivating example presented in section 2.1. It has an optimal solution of 177.5. The iterations of the algorithm for this example are explained below.

Step 0. The iteration counter is set to $i = 1$. The maximum execution time of the algorithm is set to 30 minutes and the optimality gap is set to 0.01%

Phase 1. The maximum execution time of the master problem is set to 30 seconds and the optimality gap is set to 0.5%. For the subproblem, the maximum execution time is 100 seconds and the optimality gap is 0.5%. The maximum duration of the first phase is 15 minutes and the optimality gap is set to 0.5%

Iteration 1.

Step 1.1: Master problem. The MILP reformulation of the LGDP relaxation of the hybrid formulation $(\mathbb{CSB})$ is solved using CPLEX 12.6. The optimality gap after 4 seconds is below 0.5%. The best possible objective value provided by the solver is an upper bound for the original GDP. The optimal solution is not a true upper bound because the MILP is not solved to the tolerance of the algorithm. The upper bound is set to $UB = 177.8$ and the solution of the master problem is stored for later use.

Step 1.2: Subproblem. The MINLP reformulation of $(\mathbb{CSB})$, in which the operating mode of blending tanks has been fixed according to the solution of the master problem

Table 11: Value of Boolean variables YB_{bt} at the relaxed solution

	YB^*_{1bt}					
Blending tank	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$
1	*True*		*True*			
2	*True*		*True*			
3		*True*				
4		*True*				
5		*True*		*True*		
6		*True*		*True*		
7			*True*		*True*	
8			*True*		*True*	

shown in Table 11, is solved using SCIP 3.1. The subproblem is feasible with a solution of 177.3. SCIP is able to close the gap to less than 0.5% in a second. The lower bound is set to $LB = 177.3$.

Step 1.3: Stopping criteria. Since the gap between the lower and upper bounds is less than the tolerance of the first phase, gap= $0.3\% \leq 0.5\%$, the algorithm proceeds to the second phase. The iteration counter is set to $i = 2$.

The gap between the upper bound 177.8 and the lower bound 177.3 is very small. However, for illustration purposes, the phase 2 of the algorithm is presented for the example.

Note that the algorithm only needs one iteration and less than 5 seconds to find a good feasible solution. Neither BARON 14.0, SCIP 3.1 or ANTIGONE 1.1 are able to find a solution in 30 minutes when solving the original MINLP formulation by Kolodziej et al.[6].

Phase 2. The optimality gap for the master and the subproblem is set equal to the tolerance of the algorithm, 0.01%. Time restrictions do not apply in the second phase. Since no cuts are added, the master and the subproblem in the first iteration of the second phase are the same as in the previous iteration, but the optimization has a different stopping criteria.

Iteration 2.

Step 2.1: Master problem. The optimal solution is found after 50 seconds. The new upper bound is $UB = 177.5$. At the solution, the operating modes of the blending tanks are the same as in step 1.2., which means that in the previous iteration CPLEX 12.6 found the optimal solution to the relaxed problem but it did not have time to prove global optimality.

Step 1.2: Subproblem. The GDP is the same as in the previous iteration, which means that, after finding the same solution of 177.3, SCIP continues the search until the gap is less than 0.01% and the new lower bound increases to $LB = 177.5$.

Step 1.3: Stopping criteria. Since the gap between the lower and upper bounds is less than the tolerance, gap $= 0\%$, the algorithm stops.

In summary, the decomposition algorithm only requires two iterations and less than two minutes to find the global optimum. In fact, a good feasible solution is found in only a few seconds. The correct combination of blending tanks obtained from a tight LGDP relaxation in the master problem and the critical reduction in the number of binary variables and bilinear terms leads to a feasible and more tractable GDP in the subproblem, as will be shown in the next section.

5 Computational Results

In this section, we present the computational results of applying the algorithm described in the previous section to several instances. The MINLP reformulation of the GDPs were also solved with the global optimization solver SCIP 3.1 for comparison. There are rules that could be used for deciding if the algorithm moves from one phase to another or from one iteration to the next. In this study, the stopping criteria used in the master problem, in the subproblem and in the first and second phases, are the same as the stopping criteria used to illustrate the algorithm in section 4.2.

48 instances were tested. All instances have eight blending tanks and the same topology. They can be divided in two groups: instances with initial inventory and instances without initial inventory. In each group, all combinations of instances with 1, 2, 5 and 10 specifications and 6 and 8 periods of time were generated. Table 12 shows the size of the instances in terms of the number of variables, constraints and bilinear terms. The values of the parameters were generated randomly.

Table 12: Size of the instances for the ($\mathbb{C}$) formulation. The number in parenthesis indicates the number of instances in each group.

Instances	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{Q}\rvert$	**Binary var.**	**Bilinear terms**	**Variables**	**Constraints**
A(6)	6	1	240	128	552	984
B(6)	6	2	240	256	600	1176
C(6)	6	5	240	640	772	1752
D(6)	6	10	240	1280	984	2712
E(6)	8	1	320	176	736	1312
F(6)	8	2	320	352	800	1568
G(6)	8	5	320	889	992	2336
H(6)	8	10	320	1760	1312	3616

The algorithm and models were implemented in GAMS[41]. All computations were performed on a Dell PowerEdge T410 computer with twelve Intel Xeon processors at 2.67 GHz each, 16 GB of RAM, and running Ubuntu Server 14.04 LTS (64-bit).

The decomposition algorithm is able to find at least a feasible solution for 45 out of the 48 instances generated. The three instances that were unsolved had 8 periods and 10

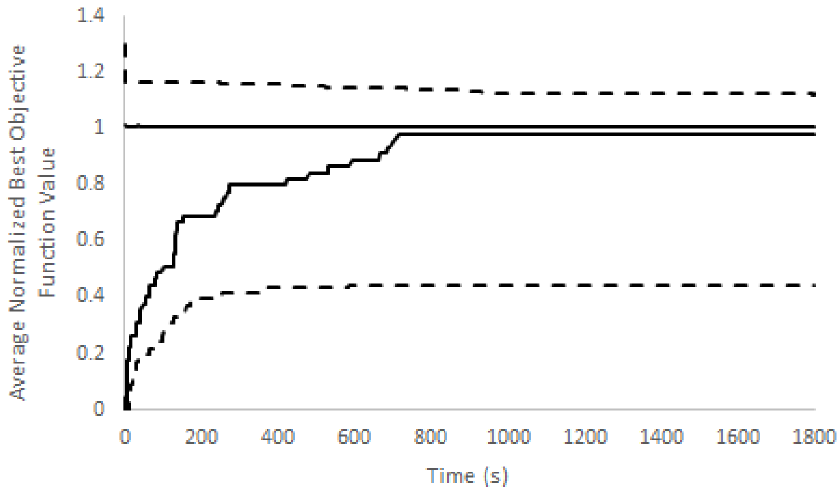

Figure 6: Evolution of the average normalized upper and lower bounds, for the decomposition algorithm (solid line) and SCIP 3.1 (dashed line) when tested in 48 instances. The graph contains the 45 instances for which a solution could be found.

specifications. SCIP 3.1 can only find solutions for 20 instances as was shown earlier in Table 10. Figure 6 shows the performance of the decomposition algorithm and the MINLP reformulation of ($\mathbb{CSB}$) using SCIP 3.1. The figure shows the average normalized upper and lower bounds. The average normalized lower bound provides the average best objective function value (ANBOFV)

The figure shows that the decomposition algorithm performs better than SCIP. After approximately two minutes, the ANBOFV of the instances solved with the algorithm is close to 0.5, whereas SCIP is below 0.3. As the execution continues, the normalized lower bound keeps increasing. After 600 seconds, the average solution is within 0.1 from the best known objective function value. After 1200 seconds, the average normalized lower and upper bounds are within 0.03 for the decomposition algorithm, while the solutions provided by SCIP are far from the best known solutions. Another important result is the value of the upper bound provided by the master problem, which is practically equal to the best known solution since the first iteration. This shows the tightness of ($\mathbb{CSB}$) when the nonconvex constrains are relaxed.

Table 13 shows the problem size, fraction of blending tanks in charge mode at the solution, the normalized upper bound and the time to get it, of the master problem for the first iteration, all averaged for the 48 instances. Notice that the average time to get a good upper bound for the original GDP problem is less than a minute, only few seconds in some cases. Also, the fraction of tanks that are doing blending is low compared with those that are discharging or in idle mode.

In the subproblem, the decomposition algorithm can find at least a feasible solution to 26 of the instances in the first iteration with a relative lower bound of 0.99. This implies that the values of the Boolean variables representing the mode of operation of the tanks given by the master problem is very close to the optimal solution for half of the instances. Table 14

Table 13: Average values for the Master Problem at the first iteration

	Master Problem (MILP reformulation of LGDP relaxation)					
Instances	Variables	Const.	Binary Var.	Fraction $YB_{bt} = True$	Normalized UB	Time (s)
A(6)	1584	1896	240	0.30	1.001	5.2
B(6)	1584	2088	240	0.30	1.007	14.9
C(6)	1584	2664	240	0.29	1.007	22.5
D(6)	1584	3624	240	0.31	1.037	22.8
E(6)	2072	2528	320	0.34	1.001	15.6
F(6)	2072	2784	320	0.34	1.007	11.0
G(6)	2072	3552	320	0.34	1.004	24.3
H(6)	2072	4832	320	0.34	1.011	41.3

shows the problem size and normalized lower bound and time for the first iteration of the subproblem.

Table 14: Average values for the Subproblem at the first iteration

	Subproblem (MINLP reformulation of GDP)					
Instance	Variables	Const.	Binary Var	Bilinear Terms	Normalized LB	Time (s)
A(6)	857	1776	56	40	0.83	7.2
B(6)	951	2302	59	83	0.66	67.4
C(6)	1058	3880	57	200	0.83	68.3
D(6)	1301	6510	58	405	0.33	68.8
E(6)	1094	2408	69	45	1.00	38.3
F(6)	1186	3130	72	91	0.33	157.2
G(6)	1400	5296	75	235	0.32	207.8
H(6)	1729	8906	76	470	0.00	177.3

The reduction in the number of binary variables and bilinear terms is essential to the success of the algorithm. On average, the number of binary variables drops 70% when compared to the original MINLP reformulation of the GDP. Similarly, the number of bilinear terms decreases by 70%. These reductions are the main reasons why the MINLP global solver used in the subproblem can find feasible solutions. Note that the lower bound is very close to the best known objective function value for those instances with 6 time periods and 1, 2 and 5 specifications. However, the value of the feasible solution for instances with 10 specifications is not as good. When dealing with 8 time periods, only those with a single specification have good lower bounds. Nevertheless, the solution for those instances with 2 and 5 specifications is still good considering that it is the first iteration.

In conclusion, the decomposition algorithm performs better than SCIP 3.1 for the 48 instances generated. The algorithm is able to find a good feasible solution for 45 of the instances in less than two minutes. The tightness of the relaxation of the master problem and the reduction in the number of binary variables and bilinear terms in the subproblem are key to the success of the algorithm.

6 Conclusions

In this manuscript, we have addressed the multiperiod blending problem, which frequently arises in the petroleum and petrochemical industry. Our main goal has been to develop new formulations and new algorithms for obtaining good feasible solutions in few minutes. We have presented two principal contributions towards solving multiperiod blending problems more effectively.

First, we have presented a source based formulation. The sources in a system are the supplies and initial inventories. They can be interpreted as raw materials of known composition. The model uses flow and inventory variables to track down each one of the sources along the network. The notion of split fraction is used to guarantee that the outflows from a tank have the same composition. These are the only nonlinearities in the model. The composition of a stream is determined from the compositions of each of the sources present in the stream. Since the latter are parameters in the model, the specification requirement constraints are linear. Lastly, we found redundant linear constraints that can be added to this model in order to improve its relaxation. In the context of a branch-and-bound search, this speeds up the convergence by reducing the number of open nodes. It was shown that the number of instances for which a feasible solution can be found using a global optimization solver increases when adding the redundant constraints (from 31 % to 41 % using SCIP 3.1)

Second, we have proposed a solution procedure that takes advantage of the operational assumption of non-simultaneous inlet/outlet streams in the blending tanks. Under this assumption, we can think of two non-coincident modes of operation for each blending tank at any time period: charge mode or discharge mode. This restriction can be modeled using disjunctions. The GDP formulation leads to a reduction in the number of bilinear terms and generates a favorable structure that can be exploited in a decomposition algorithm. Thus, an iterative two-stage MILP-MINLP decomposition method for the global optimization of the multiperiod blending problem is proposed. The first stage, or master problem, is a linear GDP relaxation of the original GDP and provides rigorous upper bounds. The second stage, or subproblem, is a smaller GDP in which the set of the binary variables representing the modes of operation for the blending tanks is fixed accordingly to the solution of the master problem. The subproblem, when a feasible solution is found, provides a feasible solution to the original GDP and a rigorous lower bound. These problems are solved successively until the gap between the upper and lower bound is closed.

The decomposition algorithm was tested in 48 instances and compared against the global optimization solver SCIP 3.1. The results show that the algorithm performs better than SCIP 3.1. In fact, the algorithm is able to find feasible solutions for 45 out of the 48 instances, whereas SCIP could only find solutions for 20 instances. Feasible solutions are obtained in less than two minutes. After less than 15 minutes, the solutions obtained with the algorithm are within 3% of the best known solutions, whereas the solutions provided by SCIP are at around 60% from the optimal values. The tightness of the relaxation of the source-based formulation when the nonconvex constraints are relaxed is reinforced by the values of the upper bound given by the master problem. They are practically equal to the best known objective function value since the first iteration. The better performance of the algorithm when compared with SCIP can be explained by the reduction in the number of binary variables and bilinear terms in the subproblem.

Acknowledgments

The authors would like to acknowledge financial support from the Center for Advanced Process Decision-making (CAPD) and from ExxonMobil.

References

[1] Kelly, J.; Mann, J. Crude oil blend scheduling optimization: an application with multimillion dollar benefits-Part 1: Process/plant optimization. *Hydrocarbon Processing* **2003**, *82*, 47–53.

[2] Haverly, C. A. Studies of the behavior of recursion for the pooling problem. *ACM SIGMAP Bulletin* **1978**, 19–28.

[3] Ben-Tal, A.; Eiger, G.; Gershovitz, V. Global minimization by reducing the duality gap. *Mathematical programming* **1994**, *63*, 193–212.

[4] Tawarmalani, M.; Sahinidis, N. V. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*; Springer Science & Business Media, 2002; Vol. 65.

[5] Audet, C.; Brimberg, J.; Hansen, P.; Digabel, S. L.; Mladenović, N. Pooling problem: Alternate formulations and solution methods. *Management science* **2004**, *50*, 761–776.

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

[7] Foulds, L.; Haugland, D.; Jörnsten, K. A bilinear approach to the pooling problem. *Optimization* **1992**, *24*, 165–180.

[8] McCormick, G. P. Computability of global solutions to factorable nonconvex programs: Part IConvex underestimating problems. *Mathematical programming* **1976**, *10*, 147–175.

[9] Androulakis, I.; Maranas, C.; Floudas, C. αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization* **1995**, *7*, 337–363.

[10] Adhya, N.; Tawarmalani, M.; Sahinidis, N. V. A Lagrangian approach to the pooling problem. *Industrial & Engineering Chemistry Research* **1999**, *38*, 1956–1972.

[11] Quesada, I.; Grossmann, I. E. Global optimization of bilinear process networks with multicomponent flows. *Computers & Chemical Engineering* **1995**, *19*, 1219–1242.

[12] Sherali, H. D.; Adams, W. P. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*; Springer Science & Business Media, 1998; Vol. 31.

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

[14] Bergamini, M. L.; Aguirre, P.; Grossmann, I. Logic-based outer approximation for globally optimal synthesis of process networks. *Computers & chemical engineering* **2005**, *29*, 1914–1933.

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

[16] Gounaris, C. E.; Misener, R.; Floudas, C. A. Computational comparison of piecewise-linear relaxations for pooling problems. *Industrial & Engineering Chemistry Research* **2009**, *48*, 5742–5766.

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

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

[19] Kolodziej, S.; Castro, P. M.; Grossmann, I. E. Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization* **2013**, *57*, 1039–1063.

[20] Teles, J. P.; Castro, P. M.; Matos, H. A. Multi-parametric disaggregation technique for global optimization of polynomial programming problems. *Journal of Global Optimization* **2013**, *55*, 227–251.

[21] Kesavan, P.; Barton, P. I. Generalized branch-and-cut framework for mixed-integer nonlinear optimization problems. *Computers & Chemical Engineering* **2000**, *24*, 1361–1366.

[22] Türkay, M.; Grossmann, I. E. Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers & Chemical Engineering* **1996**, *20*, 959–978.

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

[24] Lee, H.; Pinto, J. M.; Grossmann, I. E.; Park, S. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Industrial & Engineering Chemistry Research* **1996**, *35*, 1630–1641.

[25] Shah, N Mathematical programming techniques for crude oil scheduling. *Computers & Chemical Engineering* **1996**, *20*, S1227–S1232.

[26] Shah, N. K.; Ierapetritou, M. G. Short-term scheduling of a large-scale oil-refinery operations: Incorporating logistics details. *AIChE Journal* **2011**, *57*, 1570–1584.

[27] Méndez, C. A.; Grossmann, I. E.; Harjunkoski, I.; Kaboré, P. A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. *Computers & chemical engineering* **2006**, *30*, 614–634.

[28] Jia, Z.; Ierapetritou, M. Mixed-integer linear programming model for gasoline blending and distribution scheduling. *Industrial & Engineering Chemistry Research* **2003**, *42*, 825–835.

[29] Mouret, S.; Grossmann, I. E.; Pestiaux, P. A novel priority-slot based continuous-time formulation for crude-oil scheduling problems. *Industrial & Engineering Chemistry Research* **2009**, *48*, 8515–8528.

[30] Castro, P. M.; Grossmann, I. E. Generalized disjunctive programming as a systematic modeling framework to derive scheduling formulations. *Industrial & Engineering Chemistry Research* **2012**, *51*, 5781–5792.

[31] Moro, L. F.; Pinto, J. M. Mixed-integer programming approach for short-term crude oil scheduling. *Industrial & engineering chemistry research* **2004**, *43*, 85–94.

[32] Reddy, P. C. P.; Karimi, I.; Srinivasan, R A new continuous-time formulation for scheduling crude oil operations. *Chemical Engineering Science* **2004**, *59*, 1325–1341.

[33] Grossmann, I. E.; Trespalacios, F. Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. *AIChE Journal* **2013**, *59*, 3276–3295.

[34] Sahinidis, N. V. BARON: A general purpose global optimization software package. *Journal of global optimization* **1996**, *8*, 201–205.

[35] Misener, R.; Floudas, C. A. ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations. *Journal of Global Optimization* **2014**, *59*, 503–526.

[36] Achterberg, T. *SCIP-a framework to integrate constraint and mixed integer programming*; Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2004.

[37] Alfaki, M.; Haugland, D. Strong formulations for the pooling problem. *Journal of Global Optimization* **2013**, *56*, 897–916.

[38] Gupte, A.; Ahmed, S.; Dey, S. S.; Cheon, M. S. Pooling problems: relaxations and discretizations. *School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA. and ExxonMobil Research and Engineering Company, Annandale, NJ* **2013**.

[39] Jeroslow, R. G. *Logic-based decision support: Mixed integer model formulation*; Elsevier, 1989.

[40] Laporte, G.; Louveaux, F. V. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations research letters* **1993**, *13*, 133–142.

[41] Brooke, A.; Kendrick, D.; Meeraus, A. *GAMS Release 2.25: A user's guide*; GAMS Development Corporation Washington, DC, 1996.

A Haverly Pooling problem

A well-known benchmark problem is a small problem proposed by Haverly[2]. The problem is defined as follows: There is a single pool that receives supplies from two different sources A and B, with different specification content. A third supply C is not fed into the pool but is directly mixed with the two outflows from the pool. The specification parameters for the streams going into the pool are 3% for A, 1% for B, and 2% for C. The blending of flows from the pool and from the supply stream C produces products X and Y, which have to adhere to the specifications of maximum 2.5% and 1.5%, respectively. The maximum demands for products X and Y are 100 and 200, respectively. The objective is to minimize the cost of the blending operation while meeting the demand requirements. As opposed to the multiperiod blending problem, supply and demand flows do not depend on time and they are the decision variables. Figure 7 illustrates the topology and parameters of the network.

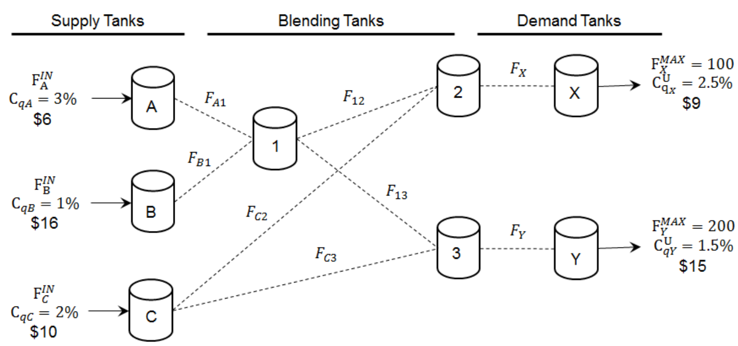

Figure 7: Sketch of Haverly Problem

The problem was modeled using p, q, pq and ($\mathbb{SB}$) formulations. They are shown in Table 16. Total flows, denoted by the variable $F_{nn'}$, are present in all models. The rest of the variables are different. Analyzing each formulation: (i) the p-formulation is based on the concentration value of the specification, C_q, in the pool and its outputs, (ii) the q and pq-formulations are based on the fraction of incoming flow to the pool that is contributed by each supply, q_s, and (iii) the source-based model is based on individual flows per source, $\tilde{F}_{snn'}$, and split fractions, ξ_{1n}, that represent the fraction of the total outgoing flow from the pool that is being sent to each of the mixers.

Another important difference is the number of variables, constraints and bilinear terms. The size of the problem and the number of bilinear terms increase from left to right in the table. Therefore, the only reason to choose the source-based model over, for example, the concentration model, is if the former had a tighter LP relaxation than the latter. Table 15 shows the size and the optimal solution of the relaxed LP for the four formulations. Two solutions are displayed. The first row corresponds to the optimal solution when the bilinear terms are replaced by their McCormick envelopes. The second row has the solutions when the nonlinear constraints are dropped entirely.

Two conclusion can be drawn. First, the optimum of the relaxed LP is the same for the p, pq and ($\mathbb{SB}$) formulations when the bilinear terms are replaced by their McCormick envelopes. However, the number of variables and constraints is larger for the latter. Secondly, when the

Table 15: Optimal solution of the relaxed LP for the Haverly problem with different formulations.

	# Variables				# Constrains				Normalized relaxation			
Relax. Bilinear Terms	**p**	**q**	**pq**	$\mathbb{SB}$	**p**	**q**	**pq**	$\mathbb{SB}$	**p**	**q**	**pq**	$\mathbb{SB}$
McCormick	14	15	15	27	19	26	30	49	1.25	6.125	1.25	1.25
w/o McCormick	12	11	11	**21**	8	6	6	**19**	5.25	9.75	9.75	**1.25**

For McCormick envelopes, the bounds on the variables are, $C_q = \{1, 3\}$, $q_s = \{0, 1\}$, $\xi_{1n} = \{0, 1\}$, $F_{nn'} = \tilde{F}_{snn'} = \{0, 300\}$

non-convex equations are eliminated from the formulations, the source-based model is tighter than the rest of the formulations. In the traditional pooling formulations, the nonconvexities appear in the specification requirements constraints, whereas in the source-based model they are only present in the "split fractions equations". This difference explains the significant improvement in the tightness of the relaxation when the non-linear terms are dropped.

Table 16 presents the p-formulation, q-formulation, pq-formulation and ($\mathbb{SB}$) formulation for the Haverly pooling problem.

Table 16: Formulations for the Haverly pooling problem. The objective function is: $min 6F_A^{IN} + 16F_B^{IN} + 10F_C^{IN} - 9F_X - 15F_Y$

Constraints	**p-formulation**	**q-formulation/pq-formulation**	**($\mathbb{SB}$)**
Mass balance	$F_A^{IN} = F_{A1}$ $F_B^{IN} = F_{B1}$ $F_C^{IN} = F_{C2} + F_{C3}$ $F_{A1} + F_{B1} = F_{12} + F_{13}$ $F_X = F_{12} + F_{C2}$ $F_Y = F_{13} + F_{C3}$	$F_A^{IN} = q_A(F_{12} + F_{13})$ $F_B^{IN} = q_B(F_{12} + F_{13})$ $F_C^{IN} = F_{C2} + F_{C3}$ $F_X = F_{12} + F_{C2}$ $F_Y = F_{13} + F_{C3}$ $q_A + q_B = 1$	$F_A^{IN} = F_{A1}$ $F_B^{IN} = F_{B1}$ $F_C^{IN} = F_{C2} + F_{C3}$ $F_{A1} + F_{B1} = F_{12} + F_{13}$ $F_X = F_{12} + F_{C2}$ $F_Y = F_{13} + F_{C3}$
Specification mass balance	$3F_{A1} + 1F_{B1} = C_q(F_{12} + F_{13})$		
Flows			$F_{A1} = \tilde{F}_{A,A1}$ $F_{B1} = \tilde{F}_{B,B1}$ $F_{C2} = \tilde{F}_{C,C2}$ $F_{C3} = \tilde{F}_{C,C3}$ $F_{12} = \tilde{F}_{A,12} + \tilde{F}_{B,12}$ $F_{13} = \tilde{F}_{A,13} + \tilde{F}_{B,13}$
Source mass balance			$\tilde{F}_{A,A1} = \tilde{F}_{A,12} + \hat{F}_{A,13}$ $\tilde{F}_{B,B1} = \tilde{F}_{B,12} + \tilde{F}_{B,13}$
Split fractions			$F_{12} = \xi_{12}(F_{A1} + F_{B1})$ $F_{13} = \xi_{13}(F_{A1} + F_{B1})$ $\tilde{F}_{A,12} = \xi_{12}\tilde{F}_{A,A1}$ $\tilde{F}_{B,12} = \xi_{12}\tilde{F}_{B,B1}$ $\tilde{F}_{A,13} = \xi_{13}\tilde{F}_{A,A1}$ $\tilde{F}_{B,13} = \xi_{13}\tilde{F}_{B,B1}$ $\xi_{12} + \xi_{13} = 1$
Redundant Constraints (for pq)		$q_A F_{12} + q_B F_{12} = F_{12}$ $q_A F_{13} + q_B F_{13} = F_{13}$ $q_A F_{12} + q_A F_{13} = 300 * q_{A1}$ $q_B F_{12} + q_B F_{13} = 300 * q_{B1}$	
Specification requirements	$C_q F_{12} + 2F_{C2} \leq 2.5F_X$ $C_q F_{13} + 2F_{C3} \leq 1.5F_Y$	$3q_A F_{12} + q_B F_{12} + 2F_{C2} \leq 2.5F_X$ $3q_A F_{13} + q_B F_{13} + 2F_{C3} \leq 1.5F_Y$	$3\tilde{F}_{A,12} + \tilde{F}_{B,12} + 2\tilde{F}_{C,C2} \leq 2.5F_X$ $3\tilde{F}_{A,13} + \tilde{F}_{B,13} + 2\tilde{F}_{C,C3} \leq 1.5F_Y$
Demands	$F_X \leq 100$ $F_Y \leq 200$	$F_X \leq 100$ $F_Y \leq 200$	$F_X \leq 100$ $F_Y \leq 200$