Systematic Solving Study for Optimization of the Multiperiod Blending Problem: A Multiple Mathematical Approach Solution Guide

Thesis presented by:

Daniel Ovalle Varela

To become:

B.Sc. in Chemical Engineering
B.Sc in Industrial Engineering

Advisors:

Jorge Mario Gómez Ph.D.
Camilo Hernándo Gómez Ph.D.

Universidad de Los Andes
Chemical Engineering Department

December 12th of 2020

Abstract

In this paper, six different approaches for the multiperiod blending problem are tested in terms of global optimality and computational time using a new set of problem instances. The solution methods discussed are the standard MINLP formulation, the relaxation created using McCormick envelopes, a Radix-Based Discretization, a generalized disjunctive programming (GDP) formulation, a Redundant Constraint GDP formulation, and a Two-Stage MILP-MINLP Decomposition (still ongoing). The addressed problem is a non-convex MINLP which has been solved for instances with a limited number of variables; hence, determining the best approach and the best solution algorithm is desirable. Results obtained show the best method is the standard MINLP, followed by the Redundant Constraint GDP and the best solution algorithms are the MIQCP algorithms provided by Gurobi. Still, results from the Two-Stage MILP-MINLP Decomposition are ongoing and have shown promising results so far.

1. Introduction

Several processes within the petrochemical industry demand the blending of various streams to obtain products with final specifications [1]. This industry is characterized by high-throughput operations and increasing regulations, resulting in small profit margins [2]. Therefore, the efficient mix of liquid materials that satisfy demanded qualities (whether for technical or environmental reasons) is desirable. By achieving optimal blending schemes that allow even the smallest improvements, large profits can be made [2][3]. This is especially important in refinery facilities in which optimal scheduling of the blend operation can lead to multimillion-dollar benefits and savings [4].

One of the first authors to address this problem was Haverly in 1978. He introduced the formulation of what is now known as the pooling problem. Briefly, the pooling problem is as follows. Various liquid streams with determined properties enter a feed-forward interconnected tank system through a supply. Then, streams are mixed perfectly in a single line of blending tanks (or "pools"), meeting the demanded characteristics before exiting the system. These characteristics are bounded by a minimum and a maximum amount for each specification. The pooling problem's goal is to assign the flows in a steady state that minimizes the cost of the operation [5][6].

Haverly's mathematical approach is denominated the 'P-formulation', which is still widely used in the chemical industry [3]. It uses compositions and total flows and considers blending tank specifications as variables [5]. Later, Ben-Tal et al. (1994) proposed a different version of the pooling problem known as the 'Q-formulation'. This model uses variables in fractions and relates how each input stream adds to the total input of the blending tanks [3][7]. Since then, several formulations, solution methods and, alternative problems have been proposed as the research area grew with time.

The problem addressed in this paper is known as the generalized pooling scheduling problem [8]. Just as the standard pooling problem proposed by Haverly, the generalized version includes bilinear terms that appear while modeling the qualities in each blending tank. However, it differs from the standard problem by not allowing pool-to-pool flow and its steady-state operation.

In the generalized pooling problem, links between mixing pools are permitted. This creates the existence of blending lines that interconnect blending tanks allowing more complex mixes to occur. Furthermore, the pooling problem operates in a steady-state, neglecting changes in supply or demand conditions. In practice, these streams may vary with time. Hence, a multiperiod scheduling formulation gives a broader approximation to the problem [6]. The wide scope of this problem enables to model operations outside the petrochemical focus by having implementations in water networks, thus applications in paper, textile, pulp, and food processing industries, among others [6][9][10].

Together with other pooling problem variations, it is an archetypical example of a mixed-integer quadratically constrained programming (MIQCP) problem. MIQCP problems are a specific case of mixed-integer quadratic programming (MIQP) which, at the same time, is contained within mixed-integer nonlinear programming (MINLP) (see Figure 1). MIQP problems allow the existence of second-order terms only, which can be either quadratic (x^2) or bilinear (xy), as nonlinear terms. If any of these MIQP problems involve a second-order term within a constraint, it is considered a MIQCP problem. The characteristics that make the multiperiod blending a MIQCP problem, and therefore an MINLP problem, will be discussed in Section 3.

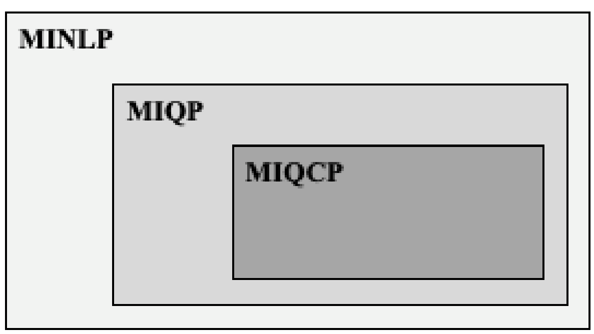


Figure 1: Mathematical classification of the problem

The research motivation of the multiperiod blending problem must not be limited to industrial interests. The multiperiod blending problem uses bilinear terms in mass balances, hence, related to equality constraints. This makes the problem non-convexifiable and, subsequently, it requires global optimization techniques to guarantee optimality [11]. Chemical Engineering optimization departments and commercial solver companies are interested in developing techniques that allow a faster solution to their MIQCP algorithms [3][12]. At the moment, this problem has been solved for cases with a limited number of variables. For example, Lotero et al. could solve problem instances with less than 1700 variables using the standard MINLP formulation [3]. Therefore, a mathematical challenge is also involved in the problem motivating further research in solving strategies.

The addressed problem will be stated in the following section. Section 3 presents the standard MINLP formulation, while Section 4 shows different mathematical approaches to solve the problem. Computational results and conclusions can be found in Sections 5 and 6, respectively.

2. The Multiperiod Blending Problem Statement

Throughout this paper, the nomenclature and problem statement are taken from those proposed by Lotero et al. as follows [3]:

Table 1: Base Nomenclature

Sets	**Dimensions***	**Symbol**	**Element**
Total tanks		N	n
Tank interconnections		A	(n, n')
Supply tanks		S	s
Blending tanks		B	b
Demand tanks		D	d
Specifications		Q	q
Time periods	$\hat{T}$	T	t
Binary Variable	**Dimensions***	**Symbol**	**Subscriptions**
Existence of flow from tank n to tank n' at the end of period t		$x_{nn't}$	$\forall\ (n, n') \in A$, $t \in T$
Continuous Variables	**Dimensions***	**Symbol**	**Subscriptions**
Flow from tank n to tank n' at the end of period t	$W/\hat{T}$	$F_{nn't}$	$\forall\ (n, n') \in A$, $t \in T$
Inventory used in tank n at the end of period t	W	I_{nt}	$\forall\ n \in N, t \in T$
Flow exiting the system through tank d at the end of period t	$W/\hat{T}$	FD_{dt}	$\forall\ d \in D, t \in T$
Composition of specification q in the blending tank b at the end of period t		C_{qbt}	$\forall\ q \in Q$, $b \in B, t \in T$

Parameters	Dimensions*	Symbol	Subscriptions
Initial value of specification q in the blending tank b		C_{qb}^{0}	$\forall\ q \in Q, b \in B$
Initial inventory used in tank n	W	I_n^0	$\forall\ n \in N$
Incoming value of specification q in the supply tank s		C_{qs}^{IN}	$\forall\ q \in Q, s \in S$
Incoming flow in supply tank s at the end of period t	$W/\hat{T}$	F_{st}^{IN}	$\forall\ s \in S, t \in T$
Bounds on flow from tank n to tank n'	$W/\hat{T}$	$[F_{nn'}^{L}, F_{nn'}^{U}]$	$\forall\ (n, n') \in A$
Bounds on the inventory of tank n	W	$[I_n^L, I_n^U]$	$\forall\ n \in N$
Bounds on flow exiting the system through tank d at the end of period t	$W/\hat{T}$	$[FD_{dt}^{L}, FD_{dt}^{U}]$	$\forall\ d \in D, t \in T$
Bounds on specification q allowed in demand tank d		$[C_{qd}^{L}, C_{qd}^{U}]$	$\forall\ q \in Q, d \in D$
Selling price of the stream produced in demand tank d	$\$/W$	β_d^T	$\forall\ d \in D$
Cost of feeding stream in supply tank s	$\$/W$	β_s^T	$\forall\ s \in S$
Cost of operating the flow from tank n to tank n'	$\$$	$\alpha_{nn'}$	$\forall\ (n, n') \in A$
Cost of total flow from tank n to tank n'	$\$/W$	$\beta_{nn'}^{N}$	$\forall\ (n, n') \in A$
A large value in terms of the considered problem		M	

$*$Dimension W stands whether for M, mol or L^3 and it must match the dimensions in which the given specifications were calculated. Dimension $\hat{T}$ stands for time.

A sketch of the problem is presented in Figure 2, in which blue arrows and terms represent parameters while black ones represent variables [3].

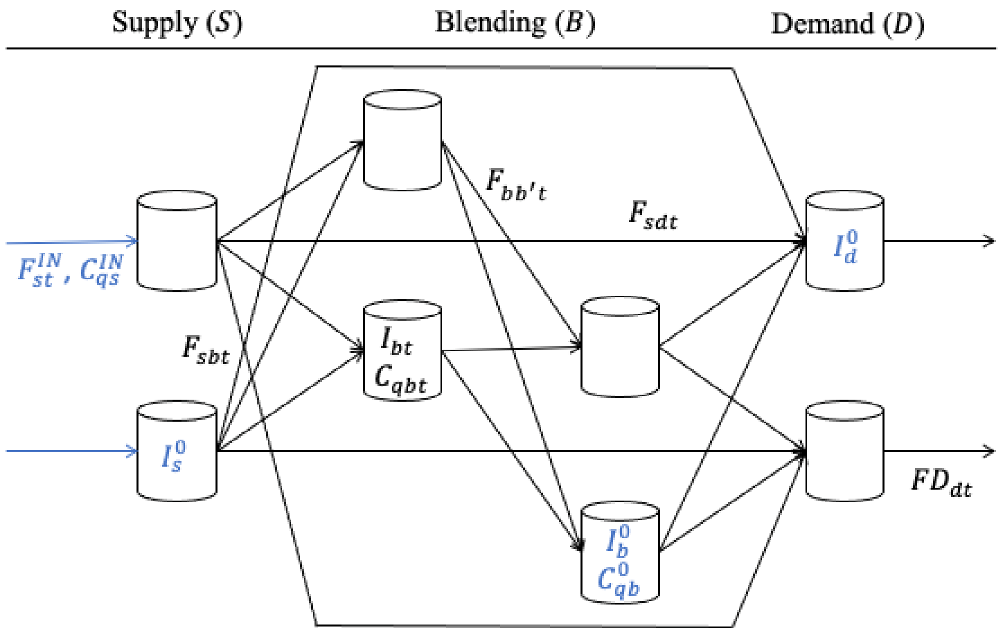


Figure 2: Blending Problem Sketch and Topology 1

The multiperiod blending problem is defined over a network of interconnected tanks (N) linked by directed streams (A). Tanks subdivide into different subsets according to their function: supply tanks (S), blending tanks (B) and, demand tanks (D). Connections between these types of tanks are generally permitted. Specifications for each substance (Q) can be calculated as qualities varying from zero to one. As in any scheduling problem, the model is considered through discrete periods of a time horizon (T) [3].

Conditions in supply such as flow (F_{st}^{IN}) and composition (C_{qs}^{IN}) are given for each tank. The initial inventory (I_n^0) and values of each specification (C_{qb}^0) are also fixed for every tank and blending tank, respectively [3].

The flow coming out of the system in each demand tank (FD_{dt}) is to be found. However, it must satisfy the flow ($[FD_{dt}^L, FD_{dt}^U]$) and composition ($[C_{qd}^L, C_{qd}^U]$) required bounds. Note that feed and output compositions (C_{qs}^{IN} and C_{qd}^L, C_{qd}^U) are assumed constant throughout the operation, while incoming and exiting flows (F_{st}^{IN} and FD_{dt}^L, FD_{dt}^U) may vary over time (thus, subscripted in t) [3].

Capacity restrictions must be contemplated as well. Tanks have a fixed minimum and maximum inventory ($[I_n^L, I_n^U]$) which can be given, for instance, by tank volume. Similarly, in case a stream flow exists between tanks, it shall be bounded ($[F_{nn'}^L, F_{nn'}^U]$) [3]. Again, these limits might be set by operational restrictions such as pumping capacity or pipe diameter.

The scheduling approach introduces decision variables that did not exist in the standard steady-state pooling problem. In a continuous operation, flows are fixed once they have been determined. Perfect mixing is assumed in the tanks allowing fixed and steady operation. Nevertheless, this assumption must not be made while scheduling for different periods. Flow cannot enter and exit a tank simultaneously; otherwise, blending will not occur [13]. This situation will be addressed by Lotero et al. by describing the "charge or discharge" mode for tanks [3]. Therefore, the decision of whether a specific flow occurs or not in a certain period ($x_{nn't}$) must be taken into consideration. The visual representation of the scheduling approach would replicate $|T|$ times the sketch shown in Figure 2 (see Figure 3) [6].

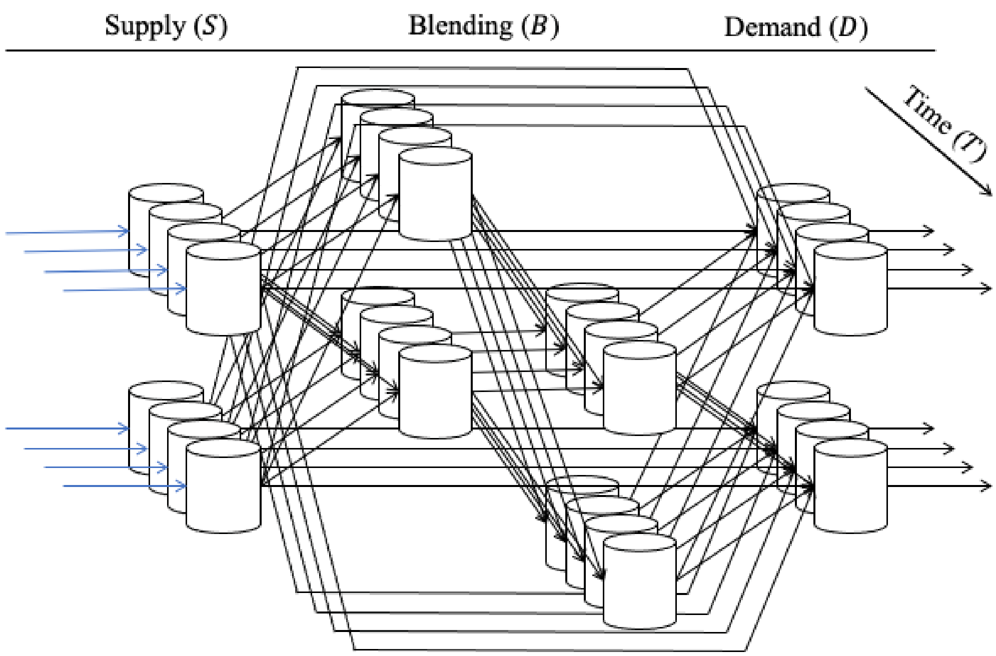


Figure 3: Multiperiod Blend Scheduling problem sketch with $T = 4$

Time periods are interdependent from one another due to the relations existing in inventories and compositions. For instance, if a particular flow exits the system through a demand tank at time t, its composition in $t - 1$ must satisfy the demanded specifications [3]. Furthermore, the nature of inventory balances requires a relationship between periods. This allows temporary storage of inventory for usage in a later period. The time period interdependence enforces the optimization to co-occur amongst all time periods [6].

The goal of the multiperiod blending problem is to maximize the profit of the entire operation. This utility equation is constructed by deducting both variable and fixed costs from an income expression. Fixed costs ($\alpha^N_{nn'}$) are associated with whether a flow exists or not, while variable costs are divided into two parts. A cost related to the amount of total flow occurring ($\beta^N_{nn'}$) and a second cost related to the incoming supply (β^T_s). Meanwhile, the income is associated with the selling of the output products and their price (β^T_d) [3].

Therefore, the problem consists of assigning the values of the flows ($F_{nn't}$) that maximize the operating profit after a given feed. These conditions must provide an output stream (FD_{dt}) within the demanded flow and composition specifications. Thus, inventories (I_{nt}) and compositions (C_{qbt}) throughout the operational horizon shall be determined [3].

3. An MINLP Direct Approach

In the multiperiod blending problem (MPBP), an MINLP problem is to be considered. As mentioned previously, the existence of a certain flow in a certain period is to be determined and modeled using binary variables. This behavior is responsible for the integer variables that, together with bilinear terms, motivate an MINLP formulation.

This problem includes a simple linear inventory mass balance to model flow inventory (4). The reason for including bilinear terms is a secondary mass balance made for each component that calculates the composition of blending tanks (5). The bilinear term appears when compositions (whether existing, entering, or leaving a tank) are multiplied by the existing inventory ($I_{bt}C_{qbt}$) or the flow in which they are moving ($F_{nbt}C_{qbt}$ or $F_{bnt}C_{qbt}$) [6]. These bilinear terms are second-order terms, hence nonlinear, making the problem MIQCP and MINLP.

An MINLP approach is the most direct way to address the problem because it does not imply extra variables or constraints. <u>*Note the term "direct" refers to the mathematical formulation but does not necessarily imply a faster solution.*</u> The MINLP formulation presents time equations in an explicit form, and it is mainly based on the proposal made by Kolodziej et al.. Still, the terminology and some expressions are from the C-Formulation in Lotero et al. [3][13].

$$\max \sum_{t \in T} \left[\sum_{(n,d) \in A} \beta_d^T F_{ndt} - \sum_{(s,n) \in A} \beta_s^T F_{snt} - \sum_{(n,n') \in A} (\alpha_{nn'}^N x_{nn't} + \beta_{nn'}^N F_{nn't}) \right] \qquad (1)$$

s.t.

$$F_{nn't} \leq F_{nn'}^U x_{nn't} \qquad \forall\, (n,n') \in A, t \in T \qquad (2a)$$

$$F_{nn't} \geq F_{nn'}^L x_{nn't} \qquad \forall\, (n,n') \in A, t \in T \qquad (2b)$$

$$C_{qbt-1} \leq C_{qd}^U + M(1 - x_{bdt}) \qquad \forall\, q \in Q, (b,d) \in A, t > 1 \in T \qquad (3a)$$

$$C_{qbt-1} \geq C_{qd}^L - M(1 - x_{bdt}) \qquad \forall\, q \in Q, (b,d) \in A, t > 1 \in T \qquad (3b)$$

$$C_{qs}^{IN} \leq C_{qd}^U + M(1 - x_{sdt}) \qquad \forall\, q \in Q, (s,d) \in A, t \in T \qquad (3c)$$

$$C_{qs}^{IN} \geq C_{qd}^{L} - M(1 - x_{sdt}) \qquad \forall\, q \in Q, (s,d) \in A, t \in T \tag{3d}$$

$$I_{st} = I_s^0 + F_{st}^{IN} - \sum_{(s,n)\in A} F_{snt} \qquad \forall\, s \in S, t = 1 \tag{4a}$$

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

$$I_{bt} = I_b^0 + \sum_{(n,b)\in A} F_{nbt} - \sum_{(b,n)\in A} F_{bnt} \qquad \forall\, b \in B, t = 1 \tag{4c}$$

$$I_{bt} = I_{bt-1} + \sum_{(n,b)\in A} F_{nbt} - \sum_{(b,n)\in A} F_{bnt} \qquad \forall\, b \in B, t > 1 \in T \tag{4d}$$

$$I_{dt} = I_d^0 + \sum_{(n,d)\in A} F_{ndt} - FD_{dt} \qquad \forall\, d \in D, t = 1 \tag{4e}$$

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

$$\begin{aligned} I_{bt}C_{qbt} = I_b^0 C_{qb}^0 &+ \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} \\ &+ \sum_{(b',b)\in A} F_{b'bt} C_{qb'}^0 - \sum_{(b,n)\in A} F_{bnt} C_{qb}^0 \end{aligned} \qquad \forall\, q \in Q, b \in B, t = 1 \tag{5a}$$

$$\begin{aligned} I_{bt}C_{qbt} = I_{bt-1}C_{qbt-1} &+ \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} \\ &+ \sum_{(b',b)\in A} F_{b'bt} C_{qb't-1} \\ &- \sum_{(b,n)\in A} F_{bnt} C_{qbt-1} \end{aligned} \qquad \forall\, q \in Q, b \in B, t > 1 \in T \tag{5b}$$

$$x_{nbt} + x_{bn't} \leq 1 \qquad \forall\, (n,b), (b,n') \in A, t \in T \tag{6}$$

$$0 \leq I_n^L \leq I_{nt} \leq I_n^U \qquad \forall\, n \in N, t \in T \tag{7a}$$

$$C_q^L \leq C_{qbt} \leq C_q^U \qquad \forall\, q \in Q, b \in B, t \in T \tag{7b}$$

$$FD_{dt}^L \leq FD_{dt} \leq FD_{dt}^U \qquad \forall\, d \in D, t \in T \tag{7c}$$

$$F_{nn't} \geq 0; \qquad x_{nn't} \in \{0,1\} \qquad \forall\, (n,n') \in A, t \in T \tag{7d}$$

In this formulation, the objective function (1) maximizes the operation's profit throughout the entire time horizon. Income is associated with the flows that exit the network, while costs come from supply and flow (both fixed and variable). Flows must be between their respective bounds if they exist (2); for a flow to exit the system, whether from a supply tank or blending tank, it should satisfy the required specification in $t-1$ (3). Equations (4) model the flow inventory balance in supply, blending, and demand tanks. The bilinear composition balance is modeled in the blending tanks as well (5). Due to the nature of mixing process scheduling, flows can either enter or exit a blending tank in a specific time period (6). Finally, every variable should satisfy its respective bounds and nature (7).

There is not much information on how to calculate the specification boundaries (7b). Kolodziej et al. suggest that compositions must vary from 0 to 1, while Lotero et al. introduce boundaries according to each specification ($[C_q^L, C_q^U]$)[3][13]. Nevertheless, Lotero et al. does not mention a method to calculate or approximate these boundaries. Subsequently, a proposal is made based on the mass balance involved in mixtures.

When blending liquids with different specifications of a particular substance, the resulting specification is a linear combination of each mixed liquid's quantity weighted by its specification. Hence, the final blend specification cannot be higher than the one in the mixed liquid with the highest specification. The same situation occurs for the lower specification bound. The following heuristic (8) is used throughout the paper as a proposal that intends to tighten the [0,1] bounds.

$$C_q^L = \min\{CIN_{qs} \quad \forall\, s \in S\} \quad \forall q \in Q \tag{8a}$$

$$C_q^U = \max\{CIN_{qs} \quad \forall\, s \in S\} \quad \forall q \in Q \tag{8b}$$

There are several commercial solution algorithms available to solve MINLP problems. The ones that will be tested in this paper are Gurobi 0, Gurobi 1, BARON, DICOPT, BONMIN, SCIP, and Couenne. Some of these algorithms will be used to solve the models generated in other approaches discussed in Sections 4.3, 4.4 and, 4.5 as well.

Gurobi has developed a nonlinear solver specialized for quadratic programming (QP) and MIQP, certifying global optimum for non-convex MICQP [12]. This algorithm uses McCormick relaxation, spatial branching, adaptative constraints, and various cutting planes [12]. This company provides two different methods for MIQCP solving, which will be addressed in this paper as Gurobi 0 and Gurobi 1. Gurobi 0 solves quadratically constrained programming (QCP) relaxations at each node, while Gurobi 1 uses a linearized outer-approximation approach [12]

Branched And Reduced Optimization Navigator (BARON) is an algorithm able to find a solution guaranteed with global optimality. This algorithm is designed for facilitating the non-convex problem solution. The BARON solving strategy integrates branch and bound with reduction tests. It also uses heuristic techniques to approximate solutions to optimization problems [14].

Discrete Continuous Optimization Package (DICOPT) is an MINLP solution algorithm that does not necessarily guarantee global optima. DICOPT uses an equality relaxation strategy based on outer-approximations. This algorithm solves series of nonlinear programming (NLP) and mixed-integer programming (MIP) subproblems [15].

Nevertheless, while solving the MPBP, it has been observed that some of the nonlinear subproblems generated result infeasible because nonlinear constraints are very restrictive [6].

Basic Open-Source Nonlinear Mixed-Integer Programming (BONMIN) is an algorithm that uses NLP branch and bound, outer approximation decompositions, branch, and cut algorithms, and hybrid approximation based on branch and cut. It also uses heuristic algorithms when problems are not convex. This algorithm is not able to certify global optima [16].

Solving Constrained Integer Program (SCIP) uses linear programming (LP) relaxations, cutting planes, and propagation to tighten variable domains. It uses primal heuristics, Benders' decomposition, branching rules, and symmetry handling in the pre-solve. This solution algorithm does not certify global optimum in the solution found [17].

Convex Over and Under Envelopes for Nonlinear Estimation (Couenne) is a solution algorithm that aims to find the global optimum for non-convex MINLP problems. It uses branching methods within a branch and bound framework, bound reduction, and linearization. Its main components are bound tightening methods, separation of linearization cuts, and branching rules [18].

4. Alternative Mathematical and Structural Approaches

The MINLP problem is still the easiest to implement and can solve instances with limited size. Still, this formulation is not sufficient for realistic medium and large-scale problems due to limitations in available algorithms and computational processing speed. Several proposals address these limitations by exploiting the mathematical structure of the problem, generating larger yet easier models. Part of the state of the art is presented in Table 2 [19].

Table 2: state of the art

Author	**Strategy**	**Type**	**Variable Formulation**
McCormick, G.P. [20]	McCormick Envelope	Convex estimator	LP
Kolodziej, S.P. et al. [6]	Radix-Based Discretization	Discretization	MILP
Lotero, I., et al. [3]	Generalized Disjunctive Programming (GDP) Formulation	GDP	MINLP
Lotero, I., et al. [3]	Redundant Constraint GDP Formulation	GDP	MINLP

Lotero, I., et al. [3]	Two-Stage MILP-MINLP Decomposition	GDP	MILP-MINLP
Lasdon, S.L., et al. [21]	Successive Linear Programming (SLP)	SLP	MINLP
Floudas, C.A., et al. [22][23]	Global Optimization Algorithms (GOP)	GOP	MINLP
Foulds, L.R., et al. [24]	Convex Envelope Branch and Bound (B&B)	Convex estimator B&B	MINLP
Quesada, I. et al. [25]	Reformulation-Linearization Techniques (RLT)	RLT	MINLP
Karuppiah, R. et al. [26]	Piece-Wise Affine Underestimators	Convex estimator B&B	MINLP

There are still many other different approaches and method combinations that have been integrated into the pooling problem; nevertheless, only the first five alternative approaches presented in Table 2 will be covered and discussed. The McCormick envelope will be tested because it was one of the first proposals chronologically, while the following four are selected due to their promissory results.

4.1 McCormick Convex Envelope Approach

The McCormick Envelope is a convex underestimator that provides an LP relaxation of the problem [20]. This is used to replace the bilinear terms involved in the specification mass balance (5) with linear approximations based on variable bounds. For each bilinear term xy, and auxiliary variable w should be introduced such as:

$$w = xy \tag{9}$$

Knowing the upper and lower bound for both variables ($x \in [x^L, x^U]$ and $y \in [y^L, y^U]$), the McCormick envelope can be expressed mathematically in the following constraints [20][21]:

$$w \leq y^U x + x^L y - x^L y^U \tag{10a}$$

$$w \leq y^L x + x^U y - x^U y^L \tag{10b}$$

$$w \geq y^L x + x^L y - x^L y^L \tag{10c}$$

$$w \geq y^U x + x^U y - x^U y^U \tag{10d}$$

This provides a convex polyhedric envelope around the bilinear term xy, and the size of the envelope depends on the distance between bounds. The maximum distance between the bilinear term xy and its auxiliary variable w is denominated $d_{\max}$ and it is proportional to the area as such [19][29]:

$$d_{\max} = \frac{1}{4}(x^U - x^L)(y^U - y^L) \tag{11}$$

In the blending problem, variables forming the bilinear term with their respective bounds are:

$$I_{bt} \in [I_b^L, I_b^U] \qquad \forall\, b \in B, t \in T \tag{12a}$$

$$F_{bnt} \in [F_{bn}^L, F_{bn}^U] \qquad \forall\, (b,n) \in A, t \in T \tag{12b}$$

$$C_{qbt} \in [C_q^L, C_q^U] \qquad \forall\, q \in Q, b \in B, t \in T \tag{12c}$$

The bilinear terms formed are shown in Table 3:

Table 3: Bilinear terms used in McCormick Envelope and Radix-Based Discretization Approaches

Bilinear Terms	**Dimensions**	**Symbol**	**Subscriptions**
Bilinear term formed by multiplying I_{bt} and C_{qbt}	W	W_{qbt}^{IC}	$\forall\, q \in Q, b \in B, t \in T$
Bilinear term formed by multiplying $F_{nn't}$ and C_{qbt}	$W/\hat{T}$	W_{qbnt}^{FC}	$\forall\, q \in Q, (b,n) \in A, t \in T$

Each bilinear term must be replaced with an auxiliary variable. Therefore, the bilinear balances (5) can be replaced with the following constraints:

$$W_{qbt}^{IC} = I_b^0 C_{qb}^0 + \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} + \sum_{(b',b)\in A} F_{b'bt} C_{qb'}^0 - \sum_{(b,n)\in A} F_{bnt} C_{qb}^0 \qquad \forall\, q \in Q, b \in B, t = 1 \tag{13a}$$

$$W^{IC}_{qbt} = W^{IC}_{qbt-1} + \sum_{(s,b)\in A} F_{sbt}\, C^{IN}_{qs} + \sum_{(b',b)\in A} W^{FC}_{qb'bt} - \sum_{(b,n)\in A} W^{FC}_{qbnt} \qquad \forall\, q \in Q, b \in B, t > 1 \in T \quad (13b)$$

$$W^{IC}_{qbt} \leq C^{U}_{q} I_{bt} + I^{L}_{b} C_{qbt} - I^{L}_{b} C^{U}_{q} \qquad \forall\, q \in Q, b \in B, t \in T \quad (14a)$$

$$W^{IC}_{qbt} \leq C^{L}_{q} I_{bt} + I^{U}_{b} C_{qbt} - I^{U}_{b} C^{L}_{q} \qquad \forall\, q \in Q, b \in B, t \in T \quad (14b)$$

$$W^{IC}_{qbt} \geq C^{L}_{q} I_{bt} + I^{L}_{b} C_{qbt} - I^{L}_{b} C^{L}_{q} \qquad \forall\, q \in Q, b \in B, t \in T \quad (14c)$$

$$W^{IC}_{qbt} \geq C^{U}_{q} I_{bt} + I^{U}_{b} C_{qbt} - I^{U}_{b} C^{U}_{q} \qquad \forall\, q \in Q, b \in B, t \in T \quad (14d)$$

$$W^{IC}_{qbt} \geq 0 \qquad \forall\, q \in Q, b \in B, t \in T \quad (14e)$$

$$W^{FC}_{qbnt} \leq C^{U}_{q} F_{bnt} + F^{L}_{bn} C_{qbt-1} - F^{L}_{bn} C^{U}_{q} \qquad \forall\, q \in Q, b \in B, t > 1 \in T \quad (15a)$$

$$W^{FC}_{qbnt} \leq C^{L}_{q} F_{bnt} + F^{U}_{bn} C_{qbt-1} - F^{U}_{bn} C^{L}_{q} \qquad \forall\, q \in Q, b \in B, t > 1 \in T \quad (15b)$$

$$W^{FC}_{qbnt} \geq C^{L}_{q} F_{bnt} + F^{L}_{bn} C_{qbt-1} - F^{L}_{bn} C^{L}_{q} \qquad \forall\, q \in Q, b \in B, t > 1 \in T \quad (15c)$$

$$W^{FC}_{qbnt} \geq C^{U}_{q} F_{bnt} + F^{U}_{bn} C_{qbt-1} - F^{U}_{bn} C^{U}_{q} \qquad \forall\, q \in Q, b \in B, t > 1 \in T \quad (15d)$$

$$W^{FC}_{qbnt} \geq 0 \qquad \forall\, q \in Q, (b,n) \in A, t \in T \quad (15e)$$

Equations (13) model de inclusion of the bilinear auxiliary variable into the model while equations (14) and (15) model the envelope (10) for W^{IC}_{qbt} and W^{FC}_{qbnt} respectively. The bilinear terms W^{FC} for flows either arriving at a blending tank (F_{nbt}) or going from a blending tank to another ($F_{bb't}$) are not modeled explicitly. Nevertheless, note that both terms $W^{FC}_{q\,nbt}$ and $W^{FC}_{qbb't}$ are implicitly included in the term W^{FC}_{qbnt}. If in the addressed problem $F^{L}_{bn} \neq 0$ the envelope must be modeled with an auxiliary bound $FMC^{L}_{bn} = 0$; otherwise, the model would be infeasible due to the possibility a flow is not activated.

The McCormick Envelope only provides an LP relaxation of the problem, which can be used as an estimator for the upper bound or initialization method. However, the solution could be infeasible in the original problem. The solution delivered depends on the tightness of the envelope, which can be improved, for instance, by reducing the distance between upper and lower bounds. Furthermore, Lotero et al. describe how changes in the formulation can affect the relaxation efficiency by providing a tighter envelope [3].

Although the McCormick Envelope approach is one of the first strategies proposed (in 1978), it is still the base of other more modern proposals. Floudas et al. developed a branch and bound algorithm for pooling problems global optimization using this strategy [3][22]. Also, Quesada and Grossmann improved the relaxation by mixing the convex envelope with RLT [3][25].

4.2 A Radix Based Discretization Approach

The objective of this method is to approximate bilinear terms with a relaxation made by adding discrete points. Then, continuous terms are introduced to fill the gaps between the discretized points, giving a continuous discretization. This strategy creates an equally spaced discretization based on the power of a numerical base (R). If the chosen base R is 10, it results in a linear increase of binary variables while the discretization precision increases by an order of magnitude [6]. The obtained form for the bilinear term x:

$$x = \sum_{l=p}^{P} \sum_{k=0}^{9} 10^l k y_{kl} \tag{16}$$

Where p is the smallest power of R to be considered, P is the largest power of R which provides an upper bound and y_{kl} is the binary variable of the discrete point that approximates the bilinear term [6]. In this general formulation, J is a set that contains all subscriptions the variables have (i,j). Therefore, a bilinear term $x_i x_j$ can be expressed in terms of an auxiliary variable w_{ij} as such [6]:

$$w_{ij} = \sum_{l=p}^{P} \sum_{k=0}^{9} 10^l k \hat{x}_{ijkl} + \sum_{k=0}^{1} 10^p k \tilde{x}_{ijk} \qquad \forall\, (i,j) \in J \tag{17a}$$

$$x_j = \sum_{l=p}^{P} \sum_{k=0}^{9} 10^l k z_{ijkl} + \sum_{k=0}^{1} 10^p k \tilde{z}_{ijk} \qquad \forall\, (i,j) \in J \tag{17b}$$

$$x_i^L z_{ijkl} \le \hat{x}_{ijkl} \le x_i^U z_{ijkl} \qquad \forall\, (i,j) \in J,\ k \in \{0, \dots, 9\},\ l \in \{p, \dots, P\} \tag{17c}$$

$$x_i^L \tilde{z}_{ijk} \le \tilde{x}_{ijk} \le x_i^U \tilde{z}_{ijk} \qquad \forall\, (i,j) \in J, k \in \{P, \dots, 9\} \tag{17d}$$

$$\sum_{k=0}^{9} \hat{x}_{ijkl} = x_i \qquad \forall\, (i,j) \in J, l \in \{p, \dots, P\} \tag{17e}$$

$$\sum_{k=0}^{1} \tilde{x}_{ijk} = x_i \qquad \forall\, (i,j) \in J \tag{17f}$$

$$\sum_{k=0}^{9} z_{ijkl} = 1 \qquad \forall\, (i,j) \in J, l \in \{p, \dots, P\} \tag{17g}$$

$$\sum_{k=0}^{1} \tilde{z}_{ijk} = 1 \qquad \forall\, (i,j) \in J \tag{17h}$$

$$0 \le \tilde{z}_{ijk'} \le 1; \qquad z_{ijkl} \in \{0,1\} \qquad \forall\, (i,j) \in J, k' \in \{0,1\},\ k \in \{0, \dots, 9\}, l \in \{p, \dots, P\} \tag{17i}$$

In this approach, z_{ijkl} is a binary variable introduced by the discretization, thus, $z_{ijkl} = 1$ only if the digit k in the l^{th} power of 10's place of x_j is activated [6]. The terms $\tilde{z}_{ijk}$ and $\tilde{x}_{ijk}$ are continuous variables introduced to fill the gap between the discretized points and be considered pseudo-slack variables [6].

Translating this strategy in terms of the MPBP, the bilinear balances (5) can be replaced with the following expressions [6]:

$$W_{qbt}^{IC} = I_b^0 C_{qb}^0 + \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} + \sum_{(b',b)\in A} F_{b'bt} C_{qb'}^0 - \sum_{(b,n)\in A} F_{bnt} C_{qb}^0 \qquad \forall\, q \in Q, b \in B, t = 1 \tag{18a}$$

$$W_{qbt}^{IC} = W_{qbt-1}^{IC} + \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} + \sum_{(b',b)\in A} W_{qb'bt}^{FC} - \sum_{(b,n)\in A} W_{qbnt}^{FC} \qquad \forall\, q \in Q, b \in B, t > 1 \in T \tag{18b}$$

$$W_{qbnt}^{FC} = \sum_{l=p}^{P} \sum_{k=0}^{9} 10^l k \hat{F}_{klqbnt} + \sum_{k=0}^{1} 10^p k \tilde{F}_{kqbnt} \qquad \forall\, q \in Q, (b,n) \in A, t \in T \tag{19a}$$

$$\hat{F}_{klqbnt} \le F_{bn}^U z_{klqbt} \qquad \forall\, k \in \{0,\dots,9\}, l \in \{p,\dots,P\}, q \in Q, (b,n) \in A, t \in T \tag{19b}$$

$$\tilde{F}_{kqbnt} \le F_{bn}^U \tilde{z}_{kqbt} \qquad \forall\, k \in \{0,1\}, q \in Q, \;\; (b,n) \in A, t \in T \tag{19c}$$

$$\sum_{k=0}^{9} \hat{F}_{klqbnt} = F_{bnt} \qquad \forall\, l \in \{p,\dots,P\}, q \in Q, \;\; (b,n) \in A, t \in T \tag{19d}$$

$$\sum_{k=0}^{1} \tilde{F}_{kqbnt} = F_{bnt} \qquad \forall\, q \in Q, (b,n) \in A, t \in T \tag{19e}$$

$$W_{qbt}^{IC} = \sum_{l=p}^{P} \sum_{k=0}^{9} 10^l k \hat{I}_{klqbt} + \sum_{k=0}^{1} 10^p k \tilde{I}_{kqbt} \qquad \forall\, q \in Q, b \in B, t \in T \tag{20a}$$

$$\hat{I}_{klqbt} \le I_b^U z_{klqbt} \qquad \forall\, k \in \{0,\dots,9\}, l \in \{p,\dots,P\}, q \in Q, b \in B, t \in T \tag{20b}$$

$$\tilde{I}_{kqbt} \le I_b^U \tilde{z}_{kqbt} \qquad \forall\, k \in \{0,1\}, q \in Q, \;\; b \in B, t \in T \tag{20c}$$

$$\sum_{k=0}^{9} \hat{I}_{klqbt} = I_{bt} \qquad \forall\, l \in \{p, \dots, P\},\; q \in Q,\;\; b \in B, t \in T \tag{20d}$$

$$\sum_{k=0}^{1} \tilde{I}_{kqbt} = I_{bt} \qquad \forall\, q \in Q, b \in B, t \in T \tag{20e}$$

$$C_{qbt} = \sum_{l=p}^{P} \sum_{k=0}^{9} 10^{l} k z_{klqbt} + \sum_{k=0}^{1} 10^{p} k \tilde{z}_{kqbt} \qquad \forall\, q \in Q, b \in B, t \in T \tag{21a}$$

$$\sum_{k=0}^{9} z_{klqbt} = 1 \qquad \forall\, l \in \{p, \dots, P\},\; q \in Q,\;\; b \in B, t \in T \tag{21b}$$

$$\sum_{k=0}^{1} \tilde{z}_{kqbt} = 1 \qquad \forall\, q \in Q, b \in B, t \in T \tag{21c}$$

$$\hat{F}_{klqbnt} \geq 0; \qquad \tilde{F}_{k'qbnt} \geq 0 \qquad \forall\, k \in \{0, \dots, 9\}, k' \in \{0,1\},\; l \in \{p, \dots, P\}, q \in Q,\; (b,n) \in A, t \in T \tag{22a}$$

$$\hat{I}_{klqbt} \geq 0; \qquad \tilde{I}_{k'qbt} \geq 0 \qquad \forall\, k \in \{0, \dots, 9\}, k' \in \{0,1\},\; l \in \{p, \dots, P\}, q \in Q,\; b \in B, t \in T \tag{22b}$$

$$0 \leq \tilde{z}_{k'qbt} \leq 1; \qquad z_{klqbt} \in \{0,1\} \qquad \forall\, k \in \{0, \dots, 9\}, k' \in \{0,1\},\; l \in \{p, \dots, P\}, q \in Q,\; b \in B, t \in T \tag{22c}$$

Just like equations (13) in the McCormick Envelope approach, equations (18) model the inclusion of the auxiliary variables for the bilinear terms into the formulation. W_{qbnt}^{FC} is modeled by the set of equations (19) while W_{qbt}^{IC} is modeled with equations (20). The second variable (C_{qbt}) and the discretization variables (z_{klqbt} and $\tilde{z}_{kqbt}$) are common to both bilinear terms and are constrained by equations (21).

The resulting model is a mixed-integer linear programming (MILP) problem due to the LP relaxation and provides a solution to the original MPBP. The number of variables in the MILP model is considerably larger than the MINLP model as discrete points, and continuous connections are added. However, the resulting MILP should be easier to solve than the original MINLP due to the model linearization [6]. Note that the linear characteristic allows solution algorithms to certify a global optimum. Besides, Kolodziej et al. develop a heuristic method (that is not discussed in this paper) to accelerate the MILP solving [6].

The results presented in this paper were obtained using values of $p = -2$ and $P = 0$. These values provide a precision in the bilinear term's discretization up to the hundredth

(0.01). A smaller value of p, -3, for instance, will provide more precision in the discretization but would increase the model size even more.

4.3 A Generalized Disjunctive Programming Approach

Generalized Disjunctive Programming (GDP) is an alternative form to formulate optimization problems that use logical expressions to relate variables. Boolean variables (True or False) are used to model an event's occurrence and its implication. For example, consider the Boolean variable X, a vector of continuous variables x, and the constraints $h(x)$ and $g(x)$ that can relate them. A GDP formulation could be as follows [28]:

$$\begin{bmatrix} X \\ h(x) \leq 0 \end{bmatrix} \vee \begin{bmatrix} \neg X \\ g(x) \leq 0 \end{bmatrix} \tag{23}$$

This formulation has an $\vee$ operator, which is known as an "inclusive or" [28]. The left side of the expression refers to the occurrence of the Boolean variable. Thus, when this variable is True (X), all the constraints above are implied ($h(x)$), and the rest ($g(x)$) are ignored. Similarly, when the Boolean variable is under negation ($\neg X$), it takes a False value and, therefore, only the constraints above are implied ($g(x)$ while neglecting $h(x)$) [28].

GDP formulations can be reformulated into a traditional MINLP model using methods such as Big-M (BM) or Hull-Reformulation [3][28]. In this paper, all GDP constraints are reformulated using BM using the parameter M, whose value may vary within expressions. Consider the following GDP formulation where X_1, X_2 and X_3 are Boolean variables; A$, B, C$ and D are continuous variables, and $\gamma, \theta, \mu,$ and π are parameters:

$$\begin{bmatrix} X_1 \\ A^L \leq A \leq A^U \end{bmatrix} \vee \begin{bmatrix} \neg X_1 \\ A = 0 \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} X_2 \\ B = \gamma C + \theta D \end{bmatrix} \vee \begin{bmatrix} \neg X_2 \\ B = \mu C + \pi D \end{bmatrix} \tag{25}$$

$$X_1 \Rightarrow X_2 \tag{26}$$

$$X_1 \Rightarrow \neg X_3 \tag{27}$$

These constraints can be reformulated using BM as [28]:

$$A^L - A \leq M(1 - X_1') \tag{28a}$$

$$A - A^U \leq M(1 - X_1') \tag{28b}$$

$$A \leq M X_1' \tag{28c}$$

$$-A \leq M X_1' \tag{28d}$$

$$X_1' \in \{0,1\} \tag{28e}$$

$$B \leq \gamma C + \theta D + M(1 - X_2') \tag{29a}$$

$$B \geq \gamma C + \theta D - M(1 - X_2') \tag{29b}$$

$$B \leq \mu C + \pi D + M X_2' \tag{29c}$$

$$B \geq \mu C + \pi D - M X_2' \tag{29d}$$

$$X_2' \in \{0,1\} \tag{29e}$$

$$X_1' \leq X_2' \tag{30}$$

$$X_1' \leq 1 - X_3' \tag{31a}$$

$$X_3' \in \{0,1\} \tag{31b}$$

In this example, disjunctive constraints (24), (25), (26), and (27) can be reformulated using the group of equations (28), (29), (30), and (31), respectively. X_1', X_2' and X_3' are the binary variables analog to X_1, X_2 and X_3 and its usage is explained below.

In the MPBP, the translated GDP-MINLP model would have fewer bilinear terms than the original MINLP approach due to logic relationships that will be discussed later [3]. For the GDP formulation, one binary and two Boolean variables are introduced.

Table 4: Boolean and Binary variables used in all GDP formulations

Boolean Variables	**Dimensions**	**Symbol**	**Subscriptions**
Boolean variable that indicates the existence of flow from tank n to tank n' when $True$ and the nonexistence of the same flow when $False$		$X_{nn't}$	$\forall\ (n, n') \in A,$ $t \in T$
Boolean variable that indicates the operating mode of blending tank b at the end of period t. $True$ value stands for 'charging' while $False$ represents 'discharging'.		YB_{bt}	$\forall\ b \in B, t \in T$
Binary Variable	**Dimensions**	**Symbol**	**Subscriptions**
Indicates the operating mode of blending tank b at the end of period t. $yb_{bt} = 1$ if the tank is in charging mode and $yb_{bt} = 0$ otherwise.		yb_{bt}	$\forall\ b \in B, t \in T$

The first variable ($X_{nn't}$) represents the occurrence of a particular flow within a specific period. This Boolean variable is analog to the binary variable $x_{nn't}$ from the original MINLP model; hence, there is no need to introduce a new binary variable to model the reformulation. Note in equations (34), (35), and (36) how only when the flow is implied (X_{nnt}), bounds are set ($F_{nn'}^{L} \leq F_{nn't} \leq F_{nn'}^{U}$). Otherwise ($\neg X_{nn't}$), the flow variable is fixed ($F_{nn't} = 0$) [3].

YB_{bt} indicates the operating mode of the tank. If $YB_{bt} = True$ then the tank is in charging mode, similarly, if $YB_{bt} = False$ the tank is in discharging mode [3]. Note in equations (37) how the complicating nonlinear mass balance is only implied if the blending tank is charging (Y_{bt}). If not ($\neg YB_{bt}$), only the leaving streams are modeled. This framework requires fewer bilinear terms than the first MINLP model, where bilinear terms were considered regardless of the tank's operation mode [3]. The analog binary variable for YB_{bt} is yb_{bt} and it is used to model the MINLP reformulation.

The explicit time expressed GDP formulation for the MINLP model is the following [3]:

$$\max \sum_{t \in T} \left[\sum_{(n,d) \in A} \beta_d^T F_{ndt} - \sum_{(s,n) \in A} \beta_s^T F_{snt} - \sum_{(n,n') \in A} (\alpha_{nn'}^N x_{nn't} + \beta_{nn'}^N F_{nn't}) \right] \tag{32}$$

s.t.

$$I_{st} = I_s^0 + F_{st}^{IN} - \sum_{(s,n) \in A} F_{snt} \qquad \forall\, s \in S, t = 1 \tag{33a}$$

$$I_{st} = I_{st-1} + F_{st}^{IN} - \sum_{(s,n) \in A} F_{snt} \qquad \forall\, s \in S, t > 1 \in T \tag{33b}$$

$$I_{dt} = I_d^0 + \sum_{(n,d) \in A} F_{ndt} - FD_{dt} \qquad \forall\, d \in D, t = 1 \tag{33c}$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d) \in A} F_{ndt} - FD_{dt} \qquad \forall\, d \in D, t > 1 \in T \tag{33d}$$

$$\begin{bmatrix} X_{nbt} \\ F_{nb}^L \leq F_{nbt} \leq F_{nb}^U \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \forall\, (n,b) \in A, t \in T \tag{34}$$

$$\begin{bmatrix} X_{sdt} \\ F_{sd}^L \leq F_{sdt} \leq F_{sd}^U \\ C_{qd}^L \leq C_{qs}^{IN} \leq C_{qd}^U \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \end{bmatrix} \qquad \forall\, (s,d) \in A, t \in T \tag{35}$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^{L} \leq F_{bdt} \leq F_{bd}^{U} \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \forall\, (b,d) \in A, t = 1 \tag{36a}$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^{L} \leq F_{bdt} \leq F_{bd}^{U} \\ C_{qd}^{L} \leq C_{qbt-1} \leq C_{qd}^{U} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \forall\, (b,d) \in A, t > 1 \in T \tag{36b}$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_{b}^{0} + \sum_{(n,b)\in A} F_{nbt} \\ I_{bt}C_{qbt} = I_{b}^{0}C_{qb}^{0} + \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} \\ + \sum_{(b',b)\in A} F_{b'bt}C_{qb'}^{0} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg YB_{bt} \\ I_{bt} = I_{b}^{0} - \sum_{(b,n)\in A} F_{bnt} \end{bmatrix} \qquad \begin{matrix} \forall\, (b,d) \\ \in A, \\ t = 1 \end{matrix} \tag{37a}$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_{bt-1} + \sum_{(n,b)\in A} F_{nbt} \\ I_{bt}C_{qbt} = I_{bt-1}C_{qbt-1} + \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} \\ + \sum_{(b',b)\in A} F_{b'bt}C_{qb't-1} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg YB_{bt} \\ I_{bt} = I_{bt-1} - \sum_{(b,n)\in A} F_{bnt} \\ C_{qbt} = C_{qbt-1} \quad \forall\, q \in Q \end{bmatrix} \qquad \begin{matrix} \forall\, (b,d) \\ \in A, \\ t > 1 \\ \in T \end{matrix} \tag{37b}$$

$$X_{nbt} \Rightarrow YB_{bt} \qquad \forall\, (n,b) \in A, t \in T \tag{38a}$$

$$X_{bnt} \Rightarrow \neg\, YB_{bt} \qquad \forall\, (b,n) \in A, t \in T \tag{38b}$$

$$0 \leq I_{n}^{L} \leq I_{nt} \leq I_{n}^{U} \qquad \forall\, n \in N, t \in T \tag{39a}$$

$$C_{q}^{L} \leq C_{qbt} \leq C_{q}^{U} \qquad \forall\, q \in Q, b \in B, t \in T \tag{39b}$$

$$FD_{dt}^{L} \leq FD_{dt} \leq FD_{dt}^{U} \qquad \forall\, d \in D, t \in T \tag{39c}$$

$$F_{nn't} \geq 0 \qquad \forall\, (n,n') \in A, t \in T \tag{39d}$$

$$X_{nn't} \in \{True, False\}; \quad YB_{bt} \in \{True, False\} \qquad \forall\, (n,n') \in A, b \in B, t \in T \tag{39d}$$

The reformulated MINLP model using BM would have twice as many bilinear terms as the original model because every equality constraint is considered two times. For instance, see how equations (29a-b) model the equality constraint $B = \gamma C + \theta D$. Nevertheless, most of these bilinear terms would belong to inactive constraints due to the relatively large values of M.

The GDP approach can be implemented on its own, generating a simpler but still nonlinear model. Straight forward MINLP models are limited by algorithm efficiency and computational power, which is why the discussion in Section 4 is relevant. However, GDP can be combined with other techniques to develop alternative strategies like the ones presented in Sections 4.4 and 4.5.

4.4 Use of Redundant Constraint GDP Approach

An essential part of solving non-convex MINLP using commercial algorithms is the tightness of its MILP convex relaxation created by the program. This MILP can be generated using, for instance, McCormick envelopes. Having an original GDP formulation, its BM reformulation of a linear GDP (LGDP) relaxation is the same as the MILP relaxation of its BM reformulation (assuming the same BM constants were used in both cases) [3]. Thus, a tighter LGDP relaxation typically means faster solving times in the GDP's MINLP reformulation [3].

Within the blending operation, the convex relaxation has a particular inconvenience. Once the non-convex model is relaxed, the only constraints modeling the flows are lineal mass balances. Therefore, the track of blending tank composition, modeled in the non-convex bilinear balances, is lost. Hence, most of the resulting solutions do not respect the demanded composition bounds being infeasible in the original problem [3].

A possible method to tighten the LGDP relaxation is to keep track of the supplies and the different sources that form a blend. Now, once the streams are blended and delivered into demand tanks, each specification's relative amount can be calculated since the composition in the sources is known [3]. This approach is used in refineries while mixing crudes for a distillation tower feed. A source (R) is the sum of both supplies and blending tanks with an initial inventory different than zero [3]:

$$R = S \cup \breve{B} \tag{40a}$$

$$\breve{B} = \{b \in B : I_b^0 > 0\} \tag{40b}$$

This introduces new variables and parameters that address the source instead of the specification [3].

Table 5: Redundant Constraint GDP Approach Terminology

Sets	**Dimensions**	**Symbol**	**Element**
Sources		R	r
Blending tanks with an existing initial inventory		$\check{B}$	b
Continuous Variables	**Dimensions**	**Symbol**	**Subscriptions**
Flow of source r from tank n to tank n' at the end of period t	$W/\hat{T}$	$\tilde{F}_{rnn't}$	$\forall\, r \in R, (n,n') \in A, t \in T$
Inventory of source used in blending tank b at the end of period t	W	$\tilde{I}_{rbt}$	$\forall\, r \in R,$ $b \in B, t \in T$
Parameter	**Dimensions**	**Symbol**	**Subscriptions**
Specification q in source r		$\hat{C}^0_{qr}$	$\forall\, q \in Q, r \in R$

The C^0_{qr} parameter can be modeled as [3]:

$$\hat{C}^0_{qs} = C^{IN}_{qs} \qquad \forall\, q \in Q, s \in S \qquad (41a)$$

$$\hat{C}^0_{qb} = C^0_{qb} \qquad \forall\, q \in Q, b \in \check{B} \qquad (41b)$$

The sum of the amount of each specification in each source equals the specification amount of the whole stream [3]:

$$C_{qbt} = \frac{\sum_{r \in R} \tilde{F}_{rbdt} \hat{C}^0_{qr}}{F_{bdt}} \qquad \forall\, q \in Q, b \in B, d \in D, t \in T \qquad (42)$$

The time explicit formulation referred to as the CSB-formulation in Lotero et al. is presented above [3]:

$$\max \sum_{t \in T} \left[\sum_{(n,d) \in A} \beta^T_d F_{ndt} - \sum_{(s,n) \in A} \beta^T_s F_{snt} - \sum_{(n,n') \in A} (\alpha^N_{nn'} x_{nn't} + \beta^N_{nn'} F_{nn't}) \right] \qquad (43)$$

s.t.

$$I_{st} = I^0_s + F^{IN}_{st} - \sum_{(s,n) \in A} F_{snt} \qquad \forall\, s \in S, t = 1 \qquad (44a)$$

$$I_{st} = I_{st-1} + F^{IN}_{st} - \sum_{(s,n) \in A} F_{snt} \qquad \forall\, s \in S, t > 1 \in T \qquad (44b)$$

$$I_{dt} = I_d^0 + \sum_{(n,d)\in A} F_{ndt} - FD_{dt} \qquad \forall\, d \in D, t = 1 \qquad (44c)$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d)\in A} F_{ndt} - FD_{dt} \qquad \forall\, d \in D, t > 1 \in T \qquad (44d)$$

$$F_{nn't} = \sum_{r\in R} \tilde{F}_{rnn't} \qquad \forall\, (n,n') \in A, t \in T \qquad (45a)$$

$$I_{bt} = \sum_{r\in R} \tilde{I}_{rbt} \qquad \forall\, b \in B, t \in T \qquad (45b)$$

$$\begin{bmatrix} X_{nbt} \\ F_{nb}^L \le F_{nbt} \le F_{nb}^U \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \forall\, (n,b) \in A, t \in T \qquad (46)$$

$$\begin{bmatrix} X_{sdt} \\ F_{sd}^L \le F_{sdt} \le F_{sd}^U \\ C_{qd}^L \le C_{qs}^{IN} \le C_{qd}^U \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \end{bmatrix} \qquad \forall\, (s,d) \in A, t \in T \qquad (47)$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^L \le F_{bdt} \le F_{bd}^U \\ C_{qd}^L F_{bdt} \le \sum_{r\in R} \tilde{F}_{rbdt} \hat{C}_{qr}^0 \le C_{qd}^U F_{bdt} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \begin{matrix} \forall\, (b,d) \in A, \\ t = 1 \end{matrix} \qquad (48a)$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^L \le F_{bdt} \le F_{bd}^U \\ C_{qd}^L \le C_{qbt-1} \le C_{qd}^U \quad \forall\, q \in Q \\ C_{qd}^L F_{bdt} \le \sum_{r\in R} \tilde{F}_{rbdt} \hat{C}_{qr}^0 \le C_{qd}^U F_{bdt} \quad \forall\, q \in Q \\ C_{qd}^L I_{bt-1} \le \sum_{r\in R} \tilde{I}_{rbt-1} \hat{C}_{qr}^0 \le C_{qd}^U I_{bt-1} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \begin{matrix} \forall\, (b,d) \in A, \\ t > 1 \in T \end{matrix} \qquad (48b)$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_b^0 + \sum_{(n,b)\in A} F_{nbt} \\ I_{bt} C_{qbt} = I_b^0 C_{qb}^0 + \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} \\ + \sum_{(b',b)\in A} F_{b'bt} C_{qb'}^0 \quad \forall\, q \in Q \\ \tilde{I}_{rbt} = I_b^0 + \sum_{(n,b)\in A} \tilde{F}_{rnbt} \quad \forall\, r \in R \end{bmatrix} \vee \qquad \forall\, b \in B, t = 1 \qquad (49a)$$

$$\begin{bmatrix} \neg YB_{bt} \\ I_{bt} = I_b^0 - \sum_{(b,n)\in A} F_{bnt} \\ \tilde{I}_{rbt} = I_b^0 - \sum_{(b,n)\in A} \tilde{F}_{rbnt} \quad \forall\, r \in R \end{bmatrix}$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_{bt-1} + \sum_{(n,b)\in A} F_{nbt} \\ I_{bt}C_{qbt} = I_{bt-1}C_{qbt-1} + \sum_{(s,b)\in A} F_{sbt}\, C_{qs}^{IN} \\ + \sum_{(b',b)\in A} F_{b'bt}C_{qb't-1} \quad \forall\, q \in Q \\ \tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum_{(n,b)\in A} \tilde{F}_{rnbt} \quad \forall\, r \in R \end{bmatrix} \bigvee \begin{bmatrix} \neg YB_{bt} \\ I_{bt} = I_{bt-1} - \sum_{(b,n)\in A} F_{bnt} \\ C_{qbt} = C_{qbt-1} \quad \forall\, q \in Q \\ \tilde{I}_{rbt} = \tilde{I}_{rbt-1} - \sum_{(b,n)\in A} \tilde{F}_{rbnt} \quad \forall\, r \in R \end{bmatrix} \qquad \forall\, b \in B, t > 1 \in T \qquad (49b)$$

$$X_{nbt} \Rightarrow YB_{bt} \qquad \forall\, (n,b) \in A, t \in T \qquad (50a)$$

$$X_{bnt} \Rightarrow \neg YB_{bt} \qquad \forall\, (b,n) \in A, t \in T \qquad (50b)$$

$$\tilde{F}_{rsnt}|_{r=s} = F_{snt} \qquad \forall\, (s,n) \in A, t \in T \qquad (51a)$$

$$\tilde{F}_{rbnt}|_{r=b} = F_{bnt} \qquad \forall\, (b,n) \in A, t = 1 \qquad (51b)$$

$$0 \le I_n^L \le I_{nt} \le I_n^U \qquad \forall\, n \in N, t \in T \qquad (52a)$$

$$C_q^L \le C_{qbt} \le C_q^U \qquad \forall\, q \in Q, b \in B, t \in T \qquad (52b)$$

$$FD_{dt}^L \le FD_{dt} \le FD_{dt}^U \qquad \forall\, d \in D, t \in T \qquad (52c)$$

$$0 \le I_b^L \le \tilde{I}_{rbt} \le I_b^U \qquad \forall\, r \in R, b \in B, t \in T \qquad (52d)$$

$$F_{nn't} \geq 0; \qquad \tilde{F}_{rnn't} \geq 0 \qquad \forall\, r \in R, (n,n') \in A, t \in T \qquad (52e)$$

$$X_{nn't} \in \{True, False\}; \quad YB_{bt} \in \{True, False\} \quad \forall\, (n,n') \in A, b \in B, t \in T \qquad (52f)$$

Equations (45) and the added constraints in equations (48-49) are redundant to the GDP model. Nevertheless, it would not be redundant for the LDGP relaxation; hence a tighter relaxation is induced [3]. The demonstration of how the LGDP relaxation of the redundant model is tighter than the LDGP relaxation of the original model will not be addressed in this paper. However, it is wholly developed and explained in Lotero et al. [3].

4.5 A Two-Stage MILP - MINLP Decomposition Approach

In Section 4.3, some advantages of the operating mode Boolean variable were discussed. In this section, these advantages will be emphasized, and others will be introduced. For instance, if a blending tank is discharging ($\dashv YB_{bt}$), there is no need for a component balance; therefore, the bilinear terms are neglected. Subsequently, if a tank is in idle mode, it can be set to discharging mode to avoid the creation of unnecessary nonlinear terms. Also, the bilinear term $F_{b\prime bt}C_{qb\prime t-1}$ is only useful if tank b is charging (YB_{bt}) and tank b' is discharging. Thus, if b has an incoming stream, but there is no blending tank b' in discharging mode ($\dashv YB_{b\prime t}$) connected to b, the bilinear term $F_{b\prime bt}C_{qb\prime t-1}$ can also be neglected [3]. Note how fixing the operation variable (YB_{bt}) can simplify the MINLP model.

This characteristic can be exploited with a two-level decomposition algorithm. The master problem is an LGDP relaxation of the original GDP in which non-convex constraints are dropped. The solution of the master problem will be used to set the operating mode variables (YB_{bt}) of the secondary problem (SP). This subproblem is a smaller GDP in which operating mode variables are fixed. Note, both master and secondary problems (when feasible) provide rigorous upper and lower bounds, respectively, to the objective. Subsequently, optimality and/or feasibility integer cuts (61) will be generated to remove regions evaluated in previous iterations. The needed sets, variables, and parameters for the decomposition are presented in Table 6 [3].

Table 6: Two-Stage MILP-MINLP Decomposition Approach Terminology

Sets	**Dimensions**	**Symbol**	**Element**
Optimality Cuts		L_O	i
Feasibility Cuts		L_F	i
Continuous Variables	**Dimensions**	**Symbol**	**Subscriptions**
Value of the objective function	$	Z	

Global upper bound	\$	UB	
Upper bound given by the objective function corresponding to the solution	\$	Z^i	$\forall\ i \in L_O$ or L_F
Parameters	**Dimensions**	**Symbol**	**Subscriptions**
Fixed operation mode binary variables when the i^{th} iteration of the master problem is optimal		$\widehat{yb}_{bt}^{i}$	$\forall\ b \in B, t \in T$
Fixed operation mode Boolean variables that are introduced in the subproblem when the master problem is optimal		$\widetilde{YB}_{bt}^{fix}$	$\forall\ b \in B, t \in T$

The time explicit formulation of the master problem (MP) is [3]:

$$\max Z \tag{53}$$

s.t.

$$Z \leq \sum_{t \in T} \left[\sum_{(n,d) \in A} \beta_d^T F_{ndt} - \sum_{(s,n) \in A} \beta_s^T F_{snt} - \sum_{(n,n') \in A} (\alpha_{nn'}^N x_{nn't} + \beta_{nn'}^N F_{nn't}) \right] \tag{54}$$

$$I_{st} = I_s^0 + F_{st}^{IN} - \sum_{(s,n) \in A} F_{snt} \qquad \forall\ s \in S, t = 1 \tag{55a}$$

$$I_{st} = I_{st-1} + F_{st}^{IN} - \sum_{(s,n) \in A} F_{snt} \qquad \forall\ s \in S, t > 1 \in T \tag{55b}$$

$$I_{dt} = I_d^0 + \sum_{(n,d) \in A} F_{ndt} - FD_{dt} \qquad \forall\ d \in D, t = 1 \tag{55c}$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d) \in A} F_{ndt} - FD_{dt} \qquad \forall\ d \in D, t > 1 \in T \tag{55d}$$

$$F_{nn't} = \sum_{r \in R} \tilde{F}_{rnn't} \qquad \forall\ (n,n') \in A, t \in T \tag{56a}$$

$$I_{bt} = \sum_{r \in R} \tilde{I}_{rbt} \qquad \forall\ b \in B, t \in T \tag{56b}$$

$$\begin{bmatrix} X_{nbt} \\ F_{nb}^L \leq F_{nbt} \leq F_{nb}^U \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \forall\ (n,b) \in A, t \in T \tag{57}$$

$$\begin{bmatrix} X_{sdt} \\ F_{sd}^{L} \leq F_{sdt} \leq F_{sd}^{U} \\ C_{qd}^{L} \leq C_{qs}^{IN} \leq C_{qd}^{U} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg\, X_{sdt} \\ F_{sdt} = 0 \end{bmatrix} \qquad \forall\, (s,d) \in A, t \in T \tag{58}$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^{L} \leq F_{bdt} \leq F_{bd}^{U} \\ C_{qd}^{L} F_{bdt} \leq \sum_{r \in R} \tilde{F}_{rbdt} \hat{C}_{qr}^{0} \leq C_{qd}^{U} F_{bdt} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg\, X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \quad \begin{matrix} \forall\, (b,d) \in A, \\ t = 1 \end{matrix} \tag{59a}$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^{L} \leq F_{bdt} \leq F_{bd}^{U} \\ C_{qd}^{L} \leq C_{qbt-1} \leq C_{qd}^{U} \qquad \forall\, q \in Q \\ C_{qd}^{L} F_{bdt} \leq \sum_{r \in R} \tilde{F}_{rbdt} \hat{C}_{qr}^{0} \leq C_{qd}^{U} F_{bdt} \quad \forall\, q \in Q \\ C_{qd}^{L} I_{bt-1} \leq \sum_{r \in R} \tilde{I}_{rbt-1} \hat{C}_{qr}^{0} \leq C_{qd}^{U} I_{bt-1} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg\, X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \quad \begin{matrix} \forall\, (b,d) \in A, \\ t > 1 \in T \end{matrix} \tag{59b}$$

$$\begin{bmatrix} YB_{bt} \\ I_{bt} = I_{b}^{0} + \sum_{(n,b) \in A} F_{nbt} \\ \tilde{I}_{rbt} = I_{b}^{0} + \sum_{(n,b) \in A} \tilde{F}_{rnbt} \quad \forall\, r \in R \\ yb_{bt} = 1 \end{bmatrix} \bigvee \begin{bmatrix} \neg\, YB_{bt} \\ I_{bt} = I_{b}^{0} - \sum_{(b,n) \in A} F_{bnt} \\ \tilde{I}_{rbt} = I_{b}^{0} - \sum_{(b,n) \in A} \tilde{F}_{rbnt} \quad \forall\, r \in R \\ yb_{bt} = 0 \end{bmatrix} \qquad \forall\, b \in B, t = 1 \tag{60a}$$

$$
\left[\begin{array}{c}
YB_{bt} \\
I_{bt} = I_{bt-1} + \sum_{(n,b)\in A} F_{nbt} \\
\tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum_{(n,b)\in A} \tilde{F}_{rnbt} \quad \forall\, r \in R \\
yb_{bt} = 1
\end{array}\right] \bigvee
\left[\begin{array}{c}
\neg\, YB_{bt} \\
I_{bt} = I_{bt-1} - \sum_{(b,n)\in A} F_{bnt} \\
C_{qbt} = C_{qbt-1} \quad \forall\, q \in Q \\
\tilde{I}_{rbt} = \tilde{I}_{rbt-1} - \sum_{(b,n)\in A} \tilde{F}_{rbnt} \quad \forall\, r \in R \\
yb_{bt} = 0
\end{array}\right] \qquad \forall\, b \in B, t > 1 \in T \qquad (60b)
$$

$$
Z \le -\left(UB - Z^i\right)\left(\sum_{\substack{b\in B,t\in T \\ \widehat{yb}_{bt}^i=1}} yb_{bt} - \sum_{\substack{b\in B,t\in T \\ \widehat{yb}_{bt}^i=0}} yb_{bt}\right) + \left(UB - Z^i\right)\left(\sum_{b\in B,t\in T} (\widehat{yb}_{bt}^i) - 1\right) + UB \qquad \forall\, i \in L_O \qquad (61a)
$$

$$
\sum_{\substack{b\in B,t\in T \\ \widehat{yb}_{bt}^i=1}} (1 - yb_{bt}) + \sum_{\substack{b\in B,t\in T \\ \widehat{yb}_{bt}^i=0}} yb_{bt} \ \ge 1 \qquad \forall\, i \in L_F \qquad (61b)
$$

$$
X_{nbt} \Rightarrow YB_{bt} \qquad \forall\, (n,b) \in A, t \in T \qquad (62a)
$$

$$
X_{bnt} \Rightarrow \neg\, YB_{bt} \qquad \forall\, (b,n) \in A, t \in T \qquad (62b)
$$

$$
\tilde{F}_{rsnt}\big|_{r=s} = F_{snt} \qquad \forall\, (s,n) \in A, t \in T \qquad (63a)
$$

$$
\tilde{F}_{rbnt}\big|_{r=b} = F_{bnt} \qquad \forall\, (b,n) \in A, t = 1 \qquad (63b)
$$

$$
0 \le I_n^L \le I_{nt} \le I_n^U \qquad \forall\, n \in N, t \in T \qquad (64a)
$$

$$C_q^L \leq C_{qbt} \leq C_q^U \qquad \forall\, q \in Q, b \in B, t \in T \tag{64b}$$

$$FD_{dt}^L \leq FD_{dt} \leq FD_{dt}^U \qquad \forall\, d \in D, t \in T \tag{64c}$$

$$0 \leq I_b^L \leq \tilde{I}_{rbt} \leq I_b^U \qquad \forall\, r \in R, b \in B, t \in T \tag{64d}$$

$$F_{nn't} \geq 0; \qquad \tilde{F}_{rnn't} \geq 0 \qquad \forall\, r \in R, (n,n') \in A, t \in T \tag{64e}$$

$$X_{nn't} \in \{True, False\}; \quad YB_{bt} \in \{True, False\} \quad \forall\, (n,n') \in A, b \in B, t \in T \tag{64f}$$

L_O is the group of optimality cuts that are added when SP is optimal while L_F enumerates the feasibility cuts added when SP is infeasible. The feasibility of the subproblem depends entirely on the solution delivered by the master problem [3]. The variable yb_{bt} is a binary variable that equals to 1 then $YB_{bt} = True$ and 0 otherwise. The corresponding subproblem time explicit formulation is [3]:

$$\max \sum_{t \in T} \left[\sum_{(n,d) \in A} \beta_d^T F_{ndt} - \sum_{(s,n) \in A} \beta_s^T F_{snt} - \sum_{(n,n') \in A} (\alpha_{nn'}^N x_{nn't} + \beta_{nn'}^N F_{nn't}) \right] \tag{65}$$

s.t.

$$I_{st} = I_s^0 + F_{st}^{IN} - \sum_{(s,n) \in A} F_{snt} \qquad \forall\, s \in S, t = 1 \tag{66a}$$

$$I_{st} = I_{st-1} + F_{st}^{IN} - \sum_{(s,n) \in A} F_{snt} \qquad \forall\, s \in S, t > 1 \in T \tag{66b}$$

$$I_{dt} = I_d^0 + \sum_{(n,d) \in A} F_{ndt} - FD_{dt} \qquad \forall\, d \in D, t = 1 \tag{66c}$$

$$I_{dt} = I_{dt-1} + \sum_{(n,d) \in A} F_{ndt} - FD_{dt} \qquad \forall\, d \in D, t > 1 \in T \tag{66d}$$

$$F_{nn't} = \sum_{r \in R} \tilde{F}_{rnn't} \qquad \forall\, (n,n') \in A, t \in T \tag{67a}$$

$$I_{bt} = \sum_{r \in R} \tilde{I}_{rbt} \qquad \forall\, b \in B, t \in T \tag{67b}$$

$$\begin{bmatrix} X_{nbt} \\ F_{nb}^L \leq F_{nbt} \leq F_{nb}^U \end{bmatrix} \vee \begin{bmatrix} \neg X_{nbt} \\ F_{nbt} = 0 \end{bmatrix} \qquad \begin{array}{c} \forall\, (n,b) \in A, t \in T, \\ \widetilde{YB}_{bt}^{fix} = True \end{array} \tag{68}$$

$$\begin{bmatrix} X_{sdt} \\ F_{sd}^{L} \le F_{sdt} \le F_{sd}^{U} \\ C_{qd}^{L} \le C_{qs}^{IN} \le C_{qd}^{U} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg X_{sdt} \\ F_{sdt} = 0 \end{bmatrix} \qquad \forall\, (s,d) \in A, t \in T \tag{69}$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^{L} \le F_{bdt} \le F_{bd}^{U} \\ C_{qd}^{L} F_{bdt} \le \sum_{r \in R} \tilde{F}_{rbdt} \hat{C}_{qr}^{0} \le C_{qd}^{U} F_{bdt} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \begin{matrix} \forall\, (b,d) \in A, \\ t = 1 \\ \widetilde{YB}_{bt}^{fix} = False \end{matrix} \tag{70a}$$

$$\begin{bmatrix} X_{bdt} \\ F_{bd}^{L} \le F_{bdt} \le F_{bd}^{U} \\ C_{qd}^{L} \le C_{qbt-1} \le C_{qd}^{U} \qquad \forall\, q \in Q \\ C_{qd}^{L} F_{bdt} \le \sum_{r \in R} \tilde{F}_{rbdt} \hat{C}_{qr}^{0} \le C_{qd}^{U} F_{bdt} \quad \forall\, q \in Q \\ C_{qd}^{L} I_{bt-1} \le \sum_{r \in R} \tilde{I}_{rbt-1} \hat{C}_{qr}^{0} \le C_{qd}^{U} I_{bt-1} \quad \forall\, q \in Q \end{bmatrix} \vee \begin{bmatrix} \neg X_{bdt} \\ F_{bdt} = 0 \end{bmatrix} \qquad \begin{matrix} \forall\, (b,d) \in A, \\ t > 1 \in T \\ \widetilde{YB}_{bt}^{fix} = False \end{matrix} \tag{70b}$$

$$I_{bt} = I_{b}^{0} + \sum_{(n,b) \in A} F_{nbt} \qquad \begin{matrix} \forall\, b \in B, t = 1 \\ \widetilde{YB}_{bt}^{fix} = True \end{matrix} \tag{71a}$$

$$I_{bt} = I_{bt-1} + \sum_{(n,b) \in A} F_{nbt} \qquad \begin{matrix} \forall\, b \in B, t > 1 \in T \\ \widetilde{YB}_{bt}^{fix} = True \end{matrix} \tag{71b}$$

$$\begin{aligned} I_{bt} C_{qbt} = I_{b}^{0} C_{qb}^{0} &+ \sum_{(s,b) \in A} F_{sbt}\, C_{qs}^{IN} \\ &+ \sum_{\substack{(b',b) \in A \\ \widetilde{YB}_{bt}^{fix} = False}} F_{b'bt} C_{qb'}^{0} \end{aligned} \qquad \begin{matrix} \forall\, q \in Q, b \in B, t = 1 \\ \widetilde{YB}_{bt}^{fix} = True \end{matrix} \tag{71c}$$

$$\begin{aligned} I_{bt} C_{qbt} = I_{bt-1} C_{qbt-1} &+ \sum_{(s,b) \in A} F_{sbt}\, C_{qs}^{IN} \\ &+ \sum_{\substack{(b',b) \in A \\ \widetilde{YB}_{bt}^{fix} = False}} F_{b'bt} C_{qb't-1} \end{aligned} \qquad \begin{matrix} \forall\, q \in Q, b \in B, t > 1 \in T \\ \widetilde{YB}_{bt}^{fix} = True \end{matrix} \tag{71d}$$

$$\tilde{I}_{rbt} = I_{b}^{0} + \sum_{(n,b) \in A} \tilde{F}_{rnbt} \qquad \begin{matrix} \forall\, r \in R, b \in B, t = 1 \\ \widetilde{YB}_{bt}^{fix} = True \end{matrix} \tag{71e}$$

$$\tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum_{(n,b)\in A} \tilde{F}_{rnbt} \qquad \begin{array}{c} \forall\, r \in R, b \in B, t > 1 \in T \\ \widetilde{YB}_{bt}^{fix} = True \end{array} \tag{71f}$$

$$I_{bt} = I_b^0 - \sum_{(b,n)\in A} F_{bnt} \qquad \begin{array}{c} \forall\, b \in B, t = 1 \\ \widetilde{YB}_{bt}^{fix} = False \end{array} \tag{72a}$$

$$I_{bt} = I_{bt-1} - \sum_{(b,n)\in A} F_{bnt} \qquad \begin{array}{c} \forall\, b \in B, t > 1 \in T \\ \widetilde{YB}_{bt}^{fix} = False \end{array} \tag{72b}$$

$$\tilde{I}_{rbt} = I_b^0 - \sum_{(b,n)\in A} \tilde{F}_{rbnt} \qquad \begin{array}{c} \forall\, r \in R, b \in B, t = 1 \\ \widetilde{YB}_{bt}^{fix} = False \end{array} \tag{72c}$$

$$\tilde{I}_{rbt} = \tilde{I}_{rbt-1} + \sum_{(b,n)\in A} \tilde{F}_{rbnt} \qquad \begin{array}{c} \forall\, r \in R, b \in B, t > 1 \in T \\ \widetilde{YB}_{bt}^{fix} = False \end{array} \tag{72d}$$

$$C_{qbt} = C_{qbt-1} \qquad \begin{array}{c} \forall\, q \in Q, b \in B, t > 1 \in T \\ \widetilde{YB}_{bt}^{fix} = False \end{array} \tag{72e}$$

$$F_{bdt} = 0 \qquad \begin{array}{c} \forall\, (b,d) \in A, t \in T \\ \widetilde{YB}_{bt}^{fix} = True \end{array} \tag{73a}$$

$$F_{nbt} = 0 \qquad \begin{array}{c} \forall\, (n,b) \in A, t \in T \\ \widetilde{YB}_{bt}^{fix} = False \end{array} \tag{73b}$$

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

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

$$0 \le I_n^L \le I_{nt} \le I_n^U \qquad \forall\, n \in N, t \in T \tag{75a}$$

$$C_q^L \le C_{qbt} \le C_q^U \qquad \forall\, q \in Q, b \in B, t \in T \tag{75b}$$

$$FD_{dt}^L \le FD_{dt} \le FD_{dt}^U \qquad \forall\, d \in D, t \in T \tag{75c}$$

$$0 \le I_b^L \le \tilde{I}_{rbt} \le I_b^U \qquad \forall\, r \in R, b \in B, t \in T \tag{75d}$$

$$F_{nn't} \ge 0; \qquad \tilde{F}_{rnn't} \ge 0 \qquad \forall\, r \in R, (n,n') \in A, t \in T \tag{75e}$$

$$X_{nn't} \in \{True, False\}; \quad YB_{bt} \in \{True, False\} \qquad \forall\, (n,n') \in A, b \in B, t \in T \tag{75f}$$

These two problems will be solved sequentially until the gap between bounds reaches a specified tolerance [3]. The resulting algorithm showed the best computing time solutions within the studied paper [3]. However, the results for this method are still ongoing and will not be presented.

5. Computational Results

In this section, the computational results of the MINLP formulation, the Radix-Based Discretization, the GDP formulation, and the Redundant Constraint GDP formulation approaches are presented and compared in terms of optimality and computational time. For the McCormick envelope method, the relative gap between the optimal (or best-found solution) and the upper bound provided by the envelope will be discussed.

A total of 24 instances (that vary from 276 to 3636 variables) with no initial inventory and a minimum flow ($F^L_{nn'} = 10 \; \forall \; (n, n') \in A$) were created and used to test all five approaches. Instances vary between three different topologies, four different specification quantities (1, 2, 5, and 10), and two time horizons (either 6 or 12 periods).

Annex 1 summarizes the design of each instance's experiments; nevertheless, an arbitrary selection of instances (instances with $Q = 10$) is summarized in Table 7 and will be shown for illustrative purposes.

Table 7: Illustrative Instance Selection Design of Experiments

Instance	**Topology***	$\lvert \boldsymbol{Q} \rvert$	$\lvert \boldsymbol{T} \rvert$
4	1	10	6
8	1	10	12
12	2	10	6
16	2	10	12
20	3	10	6
24	3	10	12

*Topologies 2 and 3 are shown in Annexes 2 and 3, respectively, while Topology 1 was already introduced in Figure 2.

The motivational example discussed in Section 2.1 in Lotero et al. will be addressed as instance 25 [3]. The topology needed for instance 25 is the same as Topology 2 but without direct flows from supply to demand (F_{sdt}). Therefore, there is no need to create nor relate these variables. All constraints that include only direct supply-demand flow are neglected in every method resulting in slightly smaller models for this instance. Also, this is the only instance tested in this paper with no minimum flow ($F^L_{nn'} = 0 \; \forall \; (n, n') \in A$). Due to these two main differences, instance 25 will be addressed separately.

All computations were performed in an Acer Spin SP315-51 computer with an Intel Core i3 processor with 2.00 at GHz, 6.00 GB of RAM, and running on Windows 10 64bit. All algorithms using Gurobi (version 9.0.1) were implemented in Python 3.8.2 while models using BARON (version 19.7.13), DICOPT (2), SCIP (version 6.0), Couenne (Library 0.5), and BONMIN (Library 1.8) were implemented in GAMS [29]. The stopping criteria for every experiment were 10800 seconds (3 hours) of computing time. Also, the gap tolerance for optimality was set at 0.01% (if applicable to the algorithm). No particular initialization was implemented; therefore, every algorithm initialized with its default settings.

5.1 MINLP Results

The MINLP approach is the most direct method for implementation; hence, the MINLP models would have the least number of variables and constraints. The algorithms used to solve the MINLP models were Gurobi 0, Gurobi 1, BARON, SCIP, BONMIN, DICOPT, and Couenne. Annex 4 presents the model size for the illustrative selection, while Table 8 shows the fraction results for all algorithms' performance. Fraction results represent the fraction of instances achieving specific criteria out of the total first 24 instances.

Table 8: Algorithm Performance in MINLP Approach (Fraction)

Algorithm	**Feasible solution found**	**Gap ≤ 0.5%**	**Gap ≤ 0.5% within 30 minutes**	**Optimal solution**	**Optimal solution within 30 minutes**
Gurobi 0	0.87	0.87	0.71	0.42	0.25
Gurobi 1	**0.87**	**0.87**	**0.75**	**0.42**	**0.29**
BARON	0.17	0.17	0.17	0.17	0.17
SCIP	0.54	0.33	0.33	0.33	0.25
BONMIN	0.08	0.08	-	0.08	0.00
Couenne	0.17	0.17	0.12	0.17	0.12
DICOPT*	*	*	*	*	*

*Infeasibility found in all cases

Due to the considerably better performance of Gurobi 1 amongst all solution algorithms, this solver would be used for all the following nonlinear models (standard GDP and Redundant Constraint GDP). The results obtained with Gurobi 1 for the illustrative selection are summarized in Annex 5. Also, detailed results obtained with all algorithms for instance 25 are presented in Table 9.

Table 9: Algorithm Performance for Instance 25 in MINLP Approach

Algorithm	Solution with Gap ≤ 0.5%			Final Solution		
	Objective [$]	Time [s]	Gap [%]	Objective [$]	Time [s]	Gap [%]
Gurobi 0	177.3	34	0.39	177.5	>10800	0.19
Gurobi 1	177.3	33	0.39	177.5	>10800	0.20
BARON	-	>10800	-	-	>10800	-
SCIP	177.2	616	0.45	177.2	>10800	0.45
BONMIN	-	>10800	-	-	>10800	-
Couenne	-	>10800	-	-	>10800	-
DICOPT*	*	*	*	*	*	*

*Infeasibility found in all cases

5.2 Approach Comparison

Alternative approaches generate larger models under the promise of a faster solution. Size attributes such as the number of variables, number of constraints, and number of bilinear constraints change significantly between approaches. Figure 4 presents, for each instance, the number of variables of the solution method in which the model is based. The number of constraints and number of bilinear constraints for each instance are presented in Annexes 6 and 7, respectively. RBD model sizes are not presented in the figure; nevertheless, these sizes can be found in Annex 8 for the illustrative selection.

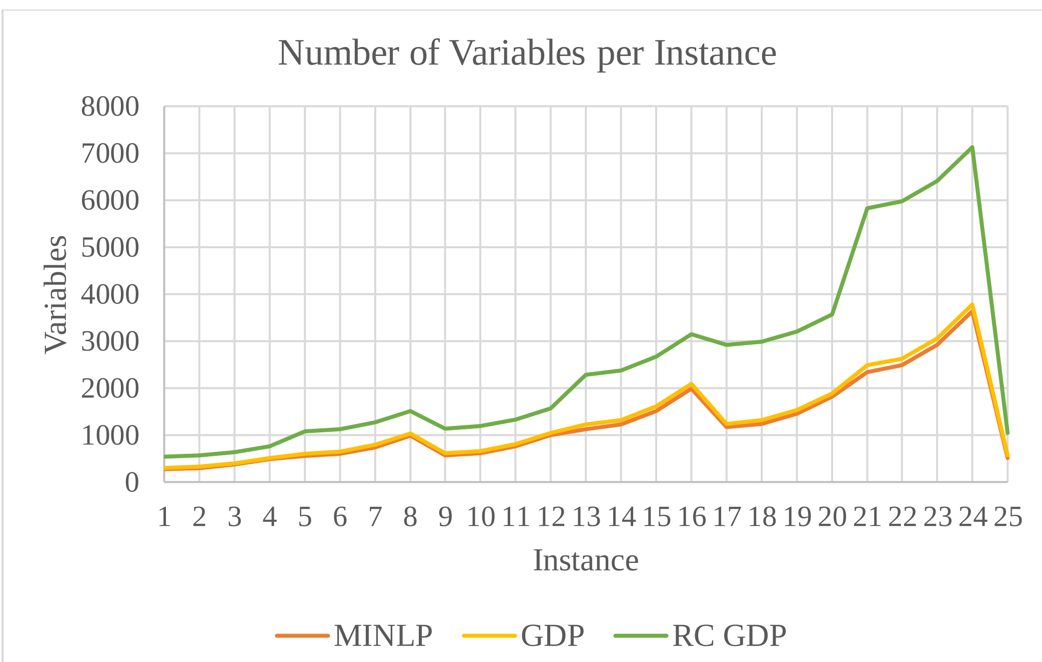


Figure 4: Instance Number of Variables for Each Approach

All methods were solved using Gurobi 1 except for the Radix-Based Discretization, which was solved using the MILP solver from Gurobi. Fraction results show how many of the first 24 instances could find a feasible solution, obtain a gap ≤ 0.5% or achieve optimality. The overall comparison between methods is summarized in Table 10.

Table 10: Overall Approach Performance Comparison (Fraction)

Approach	**Feasible solution found**	**Gap ≤ 0.5%**	**Gap ≤ 0.5% within 30 minutes**	**Optimal solution**	**Optimal solution within 30 minutes**
MINLP	0.87	**0.87**	**0.75**	**0.42**	**0.29**
RBD	0.46	0.33	0.21	0.21	0.17
GDP	0.87	0.50	0.33	0.21	0.17
RC GDP	**0.92**	0.67	0.42	0.36	0.17

For illustrative purposes, the detailed results of the illustrative selection for every method are shown in Annexes. Besides, detailed results of the performance of each approach for instance 25 are presented in Table 11. All approaches used Gurobi 1 except for RBD, which used Gurobi MILP solver.

Table 11: Approach Performance Comparison for Instance 25

Algorithm	**Solution with Gap ≤ 0.5%**			**Final Solution**		
	Objective [$]	**Time [s]**	**Gap [%]**	**Objective [$]**	**Time [s]**	**Gap [%]**
MINLP	177.3	33	0.39	177.5	>10800	0.20
RBD	-	>10800	-	-	>10800	-
GDP	-	>10800	-	177.3	>10800	22.77
RC GDP	177.1	128	0.50	177.5	>10800	0.39

5.3 McCormick Results

McCormick Envelope method resulted in a MILP model, which provided an upper bound for the problem. The relative gap between the optimal solution and the solution provided by the envelope approximation was calculated as such:

$$Gap[\%] = \frac{Solution_{McCormick} - Solution_{Optimal}}{Solution_{Optimal}} * 100 \tag{76}$$

Optimal solutions (or best-found solutions) were taken from the MINLP Gurobi 1 models' results. The statistical summary of the calculated gaps is presented above in Table 12. Also, Annex 13 shows the upper bounds and the gaps found for the illustrative instance selection.

Table 12: Statistical Summary of the Gaps [%] Found Using McCormick Envelope

Mean	Standard Deviation	Minimum	Maximum
2.25	3.06	0.00	7.93

6. Conclusions

When solving a particular instance of the multiperiod blending problem, it is highly recommended to implement the standard direct MINLP formulation. This approach is the fastest to implement and showed the best results amongst the studied methods. Nevertheless, if no optimal solution is found within the expected time, implementing the Redundant Constraint GDP (which internally includes GDP) is the second-best approach to the MPBP. Although it performed slower than the standard MINLP, the Redundant Constraint GDP approach was able to find a feasible solution in more instances. As was expected, the Redundant Constraint GDP performed better than the standard GDP, proving a larger model could still be easier to solve when generating a tighter LGDP relaxation. Still, the Two-Stage MILP-MINLP decomposition approach is ongoing and showing promising results [3].

For the standard MINLP formulation, both MIQCP algorithms provided by Gurobi were the fastest to find optimal solutions. This might be since these were the only two algorithms explicitly designed for MIQCP problems [12]. Both Gurobi 0 and Gurobi 1 found optimal solutions for the same number of instances; the difference is Gurobi 1 performed slightly faster. BARON, Couenne, BONMIN, SCIP, and DICOPT are not limited to MIQCP, enabling them to handle much more complex nonlinear terms. This feature requires complex processes within each solution algorithm that seem too slow for the larger instances. Still, the "limited" Gurobi 9.0.1 MIQCP algorithm appears to suit perfectly for this type of problem, specially Gurobi 1. In case the algorithms from Gurobi are not available, it is recommended to use SCIP to address the MPBP. Also, as stated in Kolodziej et al., DICOPT was the only algorithm that could not find any feasible solution due to infeasibility in its internal relaxations [6].

The motivating example from Lotero et al. could not be solved with an MINLP formulation in less than 30 minutes. In the paper, the authors state this was not possible using the algorithms available at that time [3]. The main difference between instance 25 and the rest is the absence of a minimum flow ($F_{nn'}^{L} = 0$). Considering Gurobi 0 and Gurobi 1 were capable of solving larger instances (in less than 30 minutes using the MINLP formulation) of the MPBP, it can be inferred that having no minimum flow bound could increase the computing time for finding an optimal solution significantly.

All solutions found using the McCormick Envelope approach were infeasible to the original problem due to the lack of rigorous calculation in the specification balance. Nevertheless, this approach provides a first insight into the problem solution and several possible values for initialization within a reasonable amount of time. The gaps provided by its upper bound are relatively small and fall within the same magnitude order as the original problem solution. Still, the usage of this method as initialization for improvement in solution time is yet to be reviewed

The multiperiod blending problem is a relevant and computationally challenging problem that still needs to be addressed. For example, instances of the problem that include an initial inventory and consider $F^{L}_{nn'} = 0$ were not considered within this paper and are still relevant to the problem. For future work, the solution methods discussed can be integrated with larger instances to evaluate performance. Based on the petrochemical industry's nature, uncertainty within supply specifications can be integrated into the problem to satisfy strict composition requirements. Besides, research can be made to combine the knowledge of this problem with multiple interconnected reactor systems.

7. References

[1] DeWitt, C.W.; Lasdon, L.S., Waren, A.D.; Brenner, D.A.; Melham, S. OMEGA: An improved gasoline blending system for Texaco. *Interfaces*, **1989**.

[2] Misener, R.; Floudas, C. A. Advances for the pooling problem: Modeling, global optimization, and computational studies. *Applied and Computational Mathematics*, **2009**.

[3] Lotero, I.; Trespalacios, F; Grossmann, I.E. An MILP-MINLP decomposition method for the global optimization of a source based model of the multiperiod blending problem. *Computers and Chemical Engineering*, **2015**.

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

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

[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**.

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

[8] Meyer, C.A. and Floudas, C.A., Global optimization of a combinatorially complex generalized pooling problem. *AIChE J*, **2006**.

[9] Galan, B; Grossmann, I. E. Optimal design of distributed wastewater treatment networks. *Industrial & Engineering Chemistry Research*, **1998**.

[10] Jeżowski, J. Review of water network design methods with literature annotations. *Industrial & Engineering Chemistry Research,* **2010**.

[11] Gounaris, C.E.; Floudas, C.A. Convexity of products of univariate functions and convexification transformations for geometric programming. *J. of Optim. Theory and Appl*, **2008**.

[12] Achterberg,T. What's new in Gurobi 9.0. *Webinar Slide Presentation*, **2019**. Accessed on: December 3, 2020. [Online]. Available: https://www.gurobi.com/wp-content/uploads/2019/12/Gurobi-9.0-Overview-Webinar-Slides-1.pdf

[13] Kolodziej, M.; Grossmann, I. E. A Novel Global Optimization Approach to the Multiperiod Blending Problem. *Computer Aided Chemical Engineering,* **2012**.

[14] Sahinidis, N. V. BARON: A general purpose global optimization software package. *Journal of Global Optimization,* **1996**.

[15] Grossmann, I. E.; Viswanathan, J.; Vecchietti, A.; Raman, R.; Kalvelagen, E. GAMS/DICOPT: A discrete continuous optimization package. **2003**.

[16] Bonami, P., Basic Open-Source Nonlinear Mixed-Integer Programming, Accessed on: December 3, 2020. [Online]. Available: https://www.coin-or.org/Bonmin/

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

[18] Belotti, P., Convex Over and Under Envelopes for Nonlinear Estimation (Couenne). Accessed on: December 3, 2020. [Online]. Available: https://github.com/coin-or/Couenne

[19] Floudas, A.; Miesner, R. Advances for the pooling problem: Modelling, global optimization, and computational studies. *Survey,* **2009**.

[20] McCormick, G.P. Computability of global solutions to factorable non-convex programs: Part 1 convex underestimating problems *Math. Program,* **1976**.

[21] Lasdon, L.S.; Waren, A.D.; Sarkar, S.; Palacios, F. Solving the pooling problem using generalized reduced gradient and successive linear programming algorithms. *ACM SIGMAP Bulletin,* **1979**.

[22] Visweswaran, V.; Floudas, C.A. A global optimization algorithm (GOP) for certain classes of non-convex NLPs: II. application of theory and test problems. *Comput. & Chem. Eng,* **1990**.

[23] Visweswaran, V.; Floudas, C.A. New formulations and branching strategies for the gop algorithm. In I. E. Grossmann, editor, *Global Optimization in Chemical Engineering,* **1996**.

[24] Foulds, L.R., Haughland, D.; Jornsten, K. A bilinear approach to the pooling problem. *Optim.,***1992**.

[25] Quesada, I.; Grossmann, I.E. Global optimization of bilinear process networks with multicomponent flows. *Comput. & Chem. Eng,* **1995**.

[26] Karuppiah, R.; Grossmann, I.E. Global optimization for the synthesis of integrated water systems in chemical processes. *Comput. & Chem. Eng.*, **2006**.

[27] Androulakis, I.P.; Maranas, C.D.; Floudas, C.A. BB: A global optimization method for general constrained non-convex problems. *J. of Glob. Optim.*, **1995**.

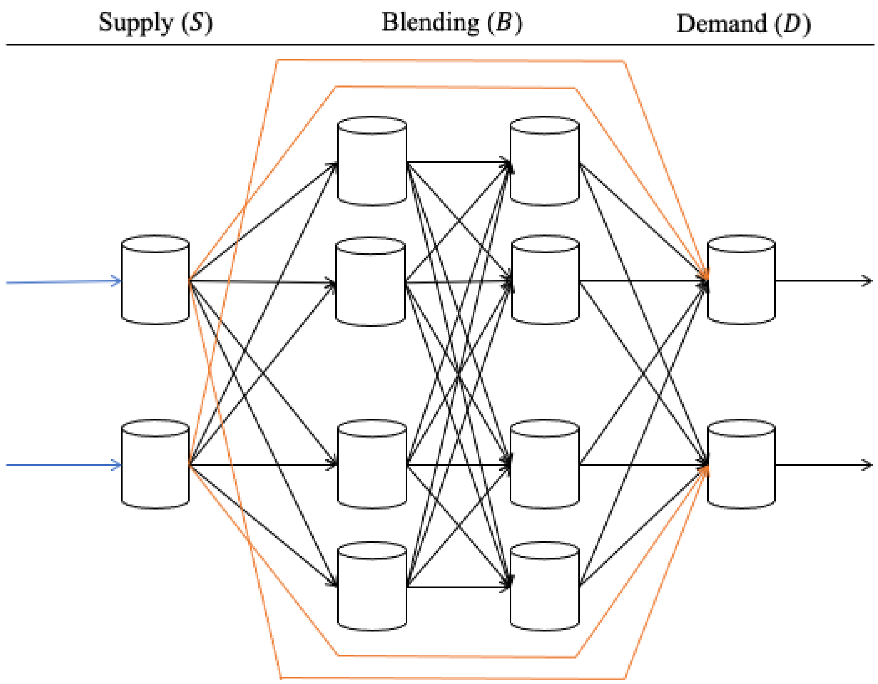
[28] Grossmann, I. E.; Trespalacios, F. Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. *AIChE Journal,* **2013**.
[29] Brooke, A.; Kendrick, D.; Meeraus, A. *GAMS Release 2.25: A user's guide*; GAMS Development Corporation Washington, DC, **1996**.

8. Annexes

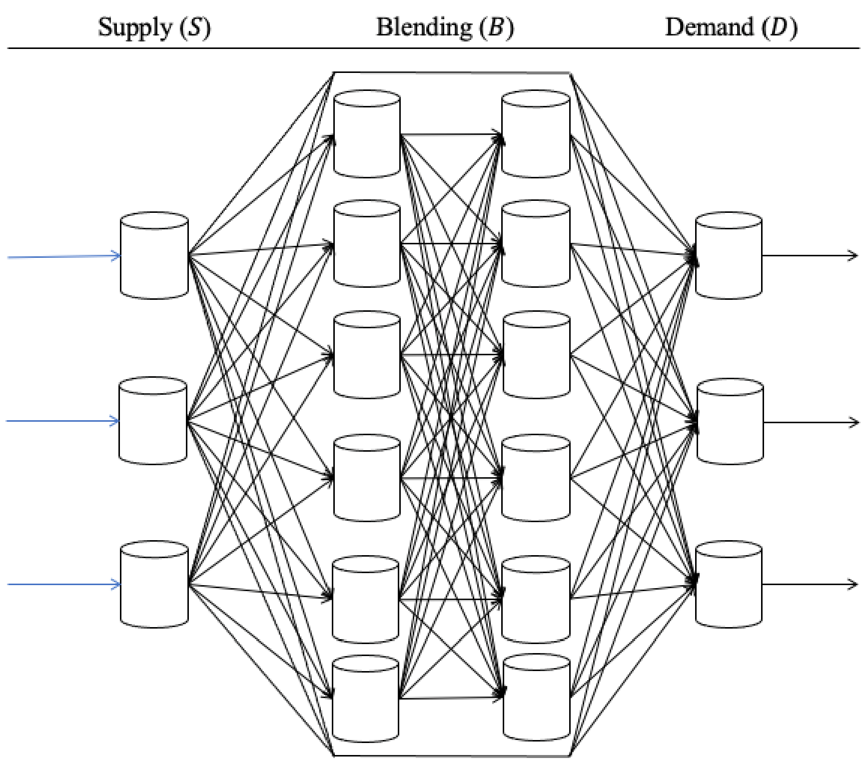
Annex 1: Instance Design of Experiments

Instances	**Topology**	$\|\boldsymbol{Q}\|$	$\|\boldsymbol{T}\|$
1	1	1	6
2	1	2	12
3	1	5	6
4	1	10	12
5	1	1	6
6	1	2	12
7	1	5	6
8	1	10	12
9	2	1	6
10	2	2	12
11	2	5	6
12	2	10	12
13	2	1	6
14	2	2	12
15	2	5	6
16	2	10	12
17	3	1	6
18	3	2	12
19	3	5	6
20	3	10	12
21	3	1	6
22	3	2	12
23	3	5	6
24	3	10	12
25	2*	1	6

*Topologies 2 and 3 are shown in Annexes 2 and 3, respectively, while Topology 1 was already introduced in Figure 2. **Topology 2 with no direct supply-demand flow (orange links) allowed.



Annex 2: Topology 2



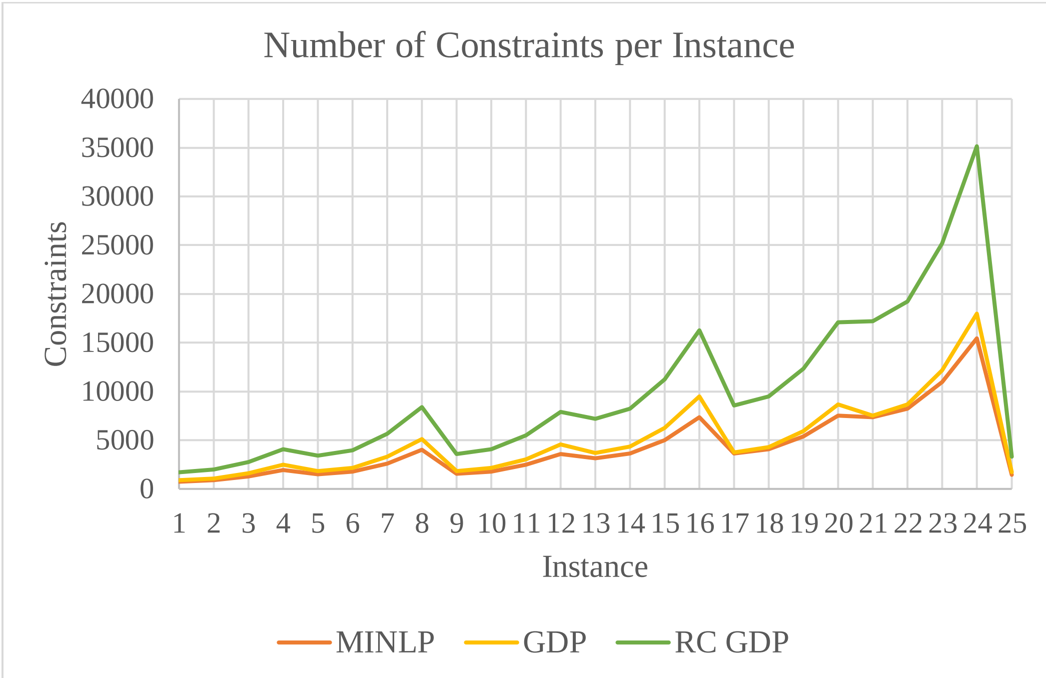
Annex 3: Topology 3

Annex 4: Illustrative Instance Selection MINLP Model Sizes

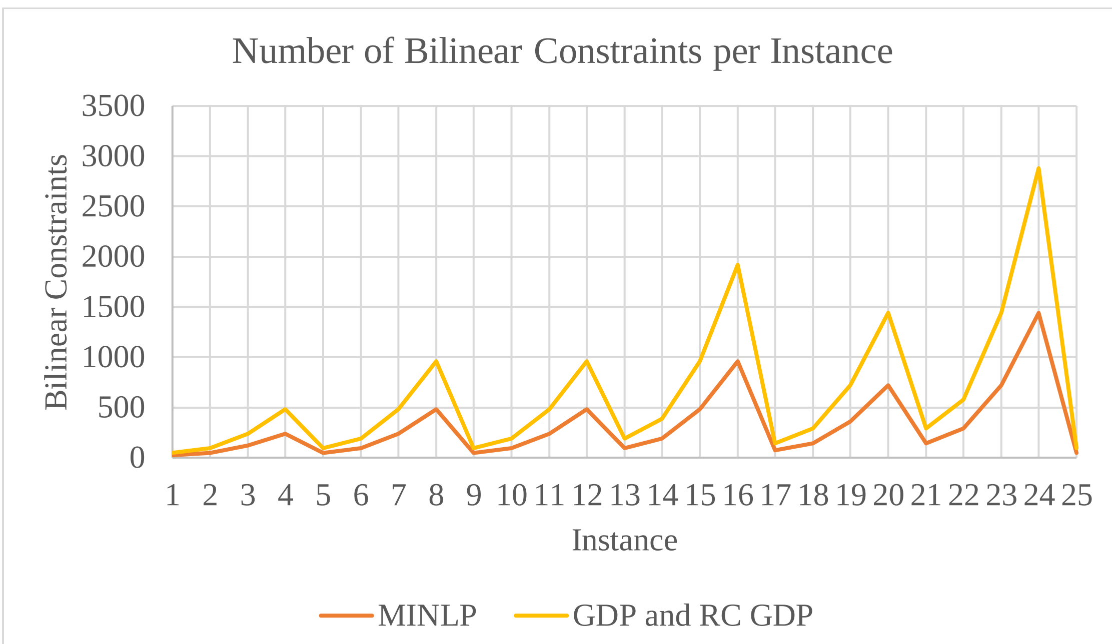
Instance	**Variables**	**Binary Variables**	**Constraints**	**Bilinear Constraints**
4	492	96	1960	240
8	984	192	4000	480
12	996	216	3584	480
16	1992	432	7328	960
20	1818	486	7542	720
24	3636	972	15444	1440

Annex 5: Gurobi 1 Results for Instance Selection in MINLP Approach

Instance	**Solution with Gap ≤ 0.5%**			**Final Solution**		
	Objective [$]	**Time [s]**	**Gap [%]**	**Objective [$]**	**Time [s]**	**Gap [%]**
4	466	15	0.43	466	16.83	0.00
8	931	1881	0.5	932	4649.03	0.00
12	457	367	0.49	457	>10800	0.22
16	914	2324	0.44	914	>10800	0.44
20	637	4211	0.10	637	>10800	0.10
24	-	>10800	-	-	>10800	-



Annex 6: Instance Number of Constraints for Each Approach



Annex 7: Instance Number of Bilinear Constraints for Each Approach

Annex 8: Illustrative Instance Selection RBD Model Sizes

Instance	Variables	Binary Variables	Constraints
4	31932	7296	54040
8	63864	14592	108160
12	79716	14616	140864
16	159432	29232	281888
20	155538	22086	287982
24	311076	44172	576324

*This method is the only approach tested that provides an entirely feasible linear model; hence, no bilinear constraints are created.

Annex 9: Gurobi 1 Results for Instance Selection in MINLP Approach

Instance	Solution with Gap ≤ 0.5%			Final Solution		
	Objective [$]	Time [s]	Gap [%]	Objective [$]	Time [s]	Gap [%]
4	466	15	0.43	466	16.83	0.00
8	931	1881	0.5	932	4649.03	0.00
12	457	367	0.49	457	>10800	0.22
16	914	2324	0.44	914	>10800	0.44
20	637	4211	0.10	637	>10800	0.10
24	-	>10800	-	-	>10800	-

Annex 10: Gurobi Results for Instance Selection in RBD Approach

Instance	Solution with Gap ≤ 0.5%			Final Solution		
	Objective [$]	Time [s]	Gap [%]	Objective [$]	Time [s]	Gap [%]
4	466	431	0.43	466	61.81	0.00
8	-	>10800	-	-	>10800	-
12	-	>10800	-	457	>10800	0.82
16	-	>10800	-	-	>10800	-
20	-	>10800	-	-	>10800	-
24	-	>10800	-	-	>10800	-

Annex 11: Gurobi 1 Results for Instance Selection in GDP Approach

Instance	Solution with Gap ≤ 0.5%			Final Solution		
	Objective [$]	**Time [s]**	**Gap [%]**	**Objective [$]**	**Time [s]**	**Gap [%]**
4	466	30	0.38	466	31.92	0.00
8	-	>10800	-	931	>10800	23.01
12	-	>10800	-	457	>10800	38.22
16	-	>10800	-	-	>10800	-
20	636	364	0.31	637	>10800	0.10
24	-	>10800	-	-	>10800	-

Annex 12: Gurobi 1 Results for Instance Selection in Redundant Constraint GDP Approach

Instance	Solution with Gap ≤ 0.5%			Final Solution		
	Objective [$]	**Time [s]**	**Gap [%]**	**Objective [$]**	**Time [s]**	**Gap [%]**
4	466	47.09	0.00	466	47.09	0.00
8	932	5831	0.50	932	>10800	0.34
12	-	>10800	-	457	>10800	16.81
16	-	>10800	-	-	>10800	-
20	-	>10800	-	637	>10800	2.60
24	-	>10800	-	-	>10800	-

Annex 13: Illustrative Instance Selection McCormick Envelope Results

Instance	Optimal Solution	Upper Bound	Computing Time [s]	Gap [%]
4	466	484	5.89	3.86
8	932	1004	27.12	7.72
12	457	458	726.26	0.22
16	914	916	2029.17	0.22
20	637	637	1865.44	0.00
24	-	1275	>10800	-