A Discretization-Based Approach for the Optimization of the Multiperiod Blend Scheduling Problem

Scott P. Kolodziej,[a] Ignacio E. Grossmann,[a*] Kevin C. Furman,[b] and Nicolas W. Sawaya[c]

[a]*Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA, 15213, USA*
[b]*ExxonMobil Upstream Research Company, 3120 Buffalo Speedway, Houston, TX 77098, USA*
[c]*ExxonMobil Gas & Power Marketing Company, 800 Bell Street, Houston, TX, 77002, USA*

Abstract

In this paper, we introduce a generalized multiperiod scheduling version of the pooling problem to represent time varying blending systems. A general nonconvex MINLP formulation of the problem is presented. The primary difficulties in solving this optimization problem are the presence of bilinear terms, as well as binary decision variables required to impose operationalconstraints . An illustrative example is presented to provide some insight into the difficulties faced by applying conventional MINLP approaches to this problem, specifically as it pertains to finding feasible solutions. A radix-based discretization scheme is developed with which the problem can be reformulated approximately as an MILP, which is incorporated in a heuristic procedure and in two rigorous global optimization methods. and requires much less computational time than existing global optimization solvers. Detailed computational results of each approach are presented on a set of examples, including a comparison with other global optimization solvers.

1. Introduction

The efficient blending of liquid fuels to meet both technical and environmental specifications has been a growing research area in recent years as stricter regulations and smaller profit margins

* Corresponding author. Tel.:+1 412 268 3642; fax:+1 412 268 7139.
E-mail address: grossmann@cmu.edu (I.E. Grossmann).

drive the need for optimal blending schemes (Misener and Floudas, 2009). Specifically in refineries, the blending of different distilled fractions to meet specifications - without waste - is of great importance.

This area of research within mathematical programming became active with the introduction of what has become known as the pooling problem, as posed by Haverly in 1978. In short, the problemis as follows: multiple liquid streams with various properties (called qualities) enter supply tanks, are fed into blending tanks where they are assumed to be perfectly mixed in some proportion to meet a set of specifications, and are then fed into demand tanks. The goal is then to select the flows that minimize the overall cost of the blending process.

The traditional pooling problem is assumed to operate at steady state, and thus inventory and other dynamics are neglected. In practice, however, supply and demand vary with time, and therefore the inventory in each tank varies as well. Further, operational rules for inventory management constrain the manner of the movement of materials into and out of the tanks. This gives rise to a multiperiod blend scheduling problem. Therefore, unlike the pooling problem, the supply and demand flows in the multiperiod problem are specified as a function of time. The multiperiod blend scheduling problem allows for the modeling of a blending system that varies over time, as is the case in liquid fuel and crude oil blending in refineries. It is, in fact, these dynamics that differentiate the multiperiod blend scheduling problem from the traditional pooling problem. Indeed, despite the presence of mixing constraints that involve bilinear constraints as in the pooling problem, the additional dynamics present in the multiperiod blend scheduling problem give rise to certain complications that will be discussed and addressed in this paper.

This paper is organized as follows. A brief review of previous work in this area is presented, divided by research area (general bilinear programs and global optimization, the pooling problem, water network optimization, blend scheduling). The multiperiod blend scheduling problem is then introduced and described in detail, and a motivating example problem is presented to illustrate the failure of conventional MINLP approaches for solving this problem. A small example isthen solved using a variety of current global optimization solvers to further illustrate these issues. The concept of radix-based discretization is then introduced, and novel reformulations of the multiperiod blend scheduling problem (originally anonconvex MIQCP) are presented that result in larger but easier to solve MILPs. These reformulations are progressively derived, resulting in heuristic and exact algorithms that require significantly reduced computational time relative to existing solvers. Finally, these new approaches are compared with each other, as well as with the original nonconvex MIQCP formulation to show the effectiveness of these new methods.

1.1. Global Optimization of General Bilinear Programs and Pooling Problems

The global optimization of bilinear programs is important in such areas as water networks and petroleum blending operations (Misener, Thompson, & Floudas, 2011; Misener & Floudas, 2009; Bagajewicz, 2000; Jeżowski, 2010). These problems, which can be classified as bilinear process networks, are generally difficult to solve to global optimality. Nonconvex, bilinear constraints are required to model the mixing of various streams in these systems, and are in some cases the only nonlinearities in the models.

General global optimization of nonconvex bilinear terms has seen many recent advances (Sherali & Alameddine, 1992; Liberti, Cafieri, & Tarissan, 2009; Liberti & Pantelides, 2006; Ruiz &

Grossmann, 2011; Wicaksono & Karimi, 2008; Al-Khayyal & Falk, 1983; Smith & Pantelides, 1997; Horst & Tuy, 1996; Floudas & Visweswaran, 1995; Shor, 1990; Xu, 2003; Yu, 1998; Ye, 1999; Adhya, Tawarmalani, & Sahinidis, 1999). The convex McCormick envelopes (McCormick, 1976) coupled with spatial branch and bound schemes have been the basis for many of these global optimization techniques, with piecewise McCormick envelopes being a recently used approach. Variations of this approach have been suggested, generalizing the convex envelopes to piecewise over- and under-estimators (Misener, Thompson, & Floudas, 2011; Wicaksono & Karimi, 2008). Novel ways of representing bilinear terms through reformulation have been a common focus in the literature (Ruiz & Grossmann, 2011). Misener, Thompson, & Floudas, (2011), building on the work of Vielma and Nemhauser (Vielma, Ahmed, & Nemhauser, 2010; Vielma & Nemhauser, 2010), have shown that a relaxation of the bilinear terms can be achieved with a logarithmic number of binary variables. Teles, Castro, & Matos (2011) have introduced a novel approximation of polynomial constraints that exhibit a similar property. Sherali & Alameddine (1992) proposed a reformulation-linearization technique (RLT), which subsumes the McCormick convex envelopes for bilinear terms to tighten bounds on bilinear terms in NLP and MINLP formulations. A subset of these RLT constraints were incorporated into a branch and bound framework by Quesada & Grossmann (1995). Karuppiah & Grossmann (2006) demonstrated the utility of adding redundant mass balance constraints as a bound tightening mechanism on the McCormick convex envelopes and applied this technique to the water network problem. Novel bounding techniques based on vector properties have been put forward by Ruiz & Grossmann (2011), as well as applications of generalized disjunctive programming to bounding bilinearities (Ruiz & Grossmann, 2010). Adhya, Tawarmalani, and Sahinidis (1999) utilized Lagrangean relaxation to obtain tighter bounds for the pooling problem, and novel piecewise under- and over-estimators for bilinear programs have been proposed by Wicaksono & Karimi (2008). Recently, refinement of previous approaches have been reported by Floudas et al. (Gounaris, Misener, & Floudas, 2009; Misener & Floudas, 2010; Misener, Thompson, & Floudas, 2011), especially using McCormick convex envelopes in piecewise approximations of bilinear terms.

Process networks involving blending, such as the pooling problem and water network design problems, are often modeled as bilinear programming problems. Much work has been devoted to finding efficient algorithms and formulations for solving this class of problem.

The pooling problem, stemming from the original Haverly paper (Haverly, 1978) has received much attention in the literature (Quesada & Grossmann, 1995; Misener, Thompson, & Floudas, 2011; Tawarmalani & Sahinidis, 2002; Meyer & Floudas, 2006; Misener & Floudas, 2009; Misener & Floudas, 2010; Pham, Laird, & El-Halwagi, 2009). Multiple liquid streams with various properties (called qualities) enter supply tanks, are fed into blending tanks where they are assumed to be perfectly mixed in some proportion to meet a set of specifications, and are then fed into demand tanks (see Figure 1). The goal is then to select the flows that minimize the overall cost of the blending process. The main assumption of the pooling problem is that it is assumed to operate at steady state; there is no accumulation in the blending system, and variations in supply and demand over time are not modeled. Both Audet et al. (2004) and Meyer & Floudas (2006) recently generalized the pooling problem to allow for a more general network topology, such as connections between blending tanks. Several different formulations have been used to solve the pooling problem, with the p-formulation being the original formulation posed by Haverly (1978), the q-formulation introduced by Ben-Tal et al. (1994), and the pq-formulation introduced by Tawarmalani & Sahinidis (2002). The p-formulation is perhaps the most intuitive modeling approach, representing the blending network with overall flows between tanks. Individual qualities in these streams are represented with mass fractions of the overall flows. The q-formulation takes a different approach by representing individual qualities with split fractions leaving the tanks, resulting in a more compact formulation. The pq-formulation combines the two formulations by adding convexification or RLT constraints to the q-formulation resulting in still better performance (Quesada & Grossmann, 1995; Sherali and Adams, 1998; Sherali et al., 1998). Other formulations, such as the TP and STP formulations, have also been introduced (Alfaki and Haugland, 2012). Recently, Misener, Thompson, & Floudas (2011) have demonstrated a novel logarithmic relaxation for modeling bilinear terms with piecewise McCormick envelopes while addressing various classes of pooling problems.

Water network optimization problems containing bilinear terms have also received much attention in the literature (Karuppiah & Grossmann, 2006; Teles, Castro, and Matos, 2012; Bagajewicz, 2000; Jeżowski, 2010; Ahmetović & Grossmann, 2010). Given a set of process units that use water, and a set of treatment units for removing the contaminants, the problem consists in finding a network configuration involving reuse and recycle to minimize the freshwater consumption. Freshwater sources are provided and wastewater disposal (or discharge) sites are also given, subject to certain constraints (e.g. an upper limit on certain contaminants) (Jeżowski, 2010). The same blending constraints present in the pooling problem are present in water network problems, and thus numerous advances in solving bilinear programs have been made addressing these problems. The synthesis of water networks has traditionally been done using pinch analysis for single contaminants (Wang & Smith, 1994a,b), but recent work has focused on using mathematical programming approaches for multiple contaminants (Karuppiah

& Grossmann, 2006; Zamora & Grossmann, 1998; Galan & Grossmann, 1998; Huang et al., 1999). These optimization problems can also be extended to consider mass and heat exchange networks (MEN and HEN, respectively), as well as other design problems (Bagajewicz et al., 2002; Zhou et al., 2009; Lim et al., 2007). Because of these extensions and the broad-reaching applications of the water network problem, the water network problem has had applications in petrochemical, food processing, pulp and paper, textile, and other industries (Jeżowski, 2010).

1.2. The Blend Scheduling Problem

Although this paper introduces a generalized problem statement and mathematical formulation for multiperiod blend scheduling, there has been a great deal of study in the literature focused on similar problems or problems that could be classified as special cases. As a scheduling extension of the blending and pooling problem, blend scheduling problems naturally arise in the petroleum refining and petrochemical industries. As such, all of the applications described in this section could potentially be formulated using the generalized model proposed in this paper. The seminal works of Shah (1996) and Lee et al. (1996) discuss similar problems with respect to mathematical programming approaches to the scheduling of crude oil unloading and inventory management at the front end of a refinery. Subsequently, there have been many papers looking at the same or similar problems (Jia et al., 2003; Kelly and Mann, 2003a; Kelly and Mann, 2003b; Moro & Pinto, 2004; Reddy et al., 2004a; Reddy et al., 2004b; Li et al., 2007; Karuppiah et al., 2008; Mouret et al., 2009; Saharidis & Ierapetritou, 2009; Misener & Floudas, 2012; Liang et al., in press). Other efforts have focused on the blending of end-products of a refinery (Moro et al., 1998; Jia & Ierapetritou, 2003; Ierapetritou and Jia, 2008). Finally, some articles have looked at general refinery models or to combine models for crude feed blending and end-product blending (Pinto et al., 2000; Jia and Ierapetritou, 2004; Shah and Ierapetritou, 2011).

In a related application similar to crude oil scheduling for a refinery, Tjoa et al. (1997) look at the raw material feed scheduling for an ethylene plant. In a very complex application, Balasubramanian et al. (2010) combine very complex maritime inventory routing with the onshore blending of the delivered products in a combined routing and blend scheduling problem. Blend scheduling is an important component in multiple industries, and could become an important stepping stone to enterprise-wide optimization applications when combined with models for other parts of the supply chain.

Both continuous time and discrete time scheduling models have been proposed for addressing this type of problem, however most of the literature focuses on application specific mathematical programming formulations. Furman et al. (2007) posed a generalized event-based continuous time model for blend tank scheduling of a very similar nature to the problem discussed in this paper. Due to the complications in synchronizing the continuous time events pointed out by Furman et al. (2007) and the tendency for discrete time scheduling models to have tighter relaxations than their continuous time counterparts (Maravelias & Papalamprou, 2009), the model proposed in this paper incorporates discrete time periods. However, it should be noted that the radix-based discretization techniques discussed in later sections could also potentially be applied to continuous time models.

2. Problem Statement

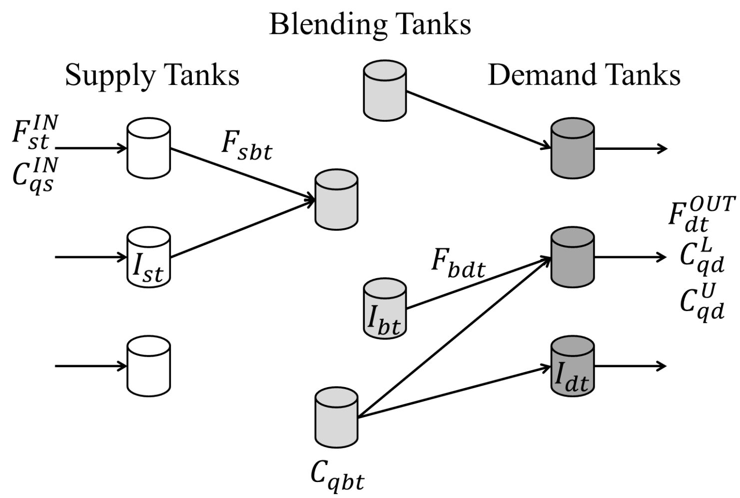


Figure 1: Detailed diagram of one period of the multiperiod blend scheduling problem

The multiperiod blend scheduling problem addressed in this paper can be stated as follows. A network is given whose nodes $n \in TA$ consist of subsets of supply, blending, and demand tanks $TA = S \cup B \cup D, s \in S, b \in B, d \in D$. These nodes are interconnected by directed arcs $(nn') \in N$ corresponding to streams between the tanks. It should be noted that interconnections between the supply and demand tanks, as well as between blending tanks, are allowed by the model. The network operates over a time horizon defined by a set of time periods $T = \{0, .. t, .. NT\}$ as shown in Figure 1. At time $t = 0$, an initial inventory is specified for each tank, as well as initial values for the qualities. Given the specified network topology, the optimal flows between the tanks in each time period must be determined, as well as the corresponding inventory levels, which are carried over from one period to another. It is important to note that each time period $t \in T$ is not independent of the others due to the coupling created by the inventories. For example, flow may be diverted to a tank for temporary storage to be used in a later period, or there may be no direct path from a source tank to a demand tank. Thus, the optimization must be performed simultaneously over all time periods $t \in T$. Specifically, incoming supply flows $F_{st}^{IN}, s \in S$, enter supply tanks each time period, and demand flows

F_{dt}^{OUT}, $d \in D$, are withdrawn from the demand tanks each time period. The supply flows to a given tank are assumed for simplicity to have the same quality or composition over all time C_{qs}^{IN}, $q \in Q$, but can vary in amount (hence the subscript t in F_{st}^{IN}). Likewise, the concentration of flows leaving the demand tanks must be within specified bounds, C_{qd}^{L} and C_{qd}^{U}, but the flows can also vary in amount (hence the subscript t in F_{dt}^{OUT}). Bounds on inventories are also given for each tank, I_n^L and I_n^U, and for each flow $F_{nn't}$, between each pair of tanks n, n' ($F_{nn'}^{L}$, usually zero, and $F_{nn'}^{U}$). Lastly, costs for the supply flows, β_s, $s \in S$, prices for demand flows, β_d, $d \in D$, and fixed and variable costs for flows within the network, α and β, respectively, are taken into account with the goal of maximizing the profit (or minimizing the costs) of the network schedule to most efficiently mix the fuels to meet demand specifications. Because of the operational constraint that flow cannot both enter and exit a blending tank in the same time period, as well as to represent the fixed costs, binary decision variables $y_{nn't}$ must be introduced into the problem such that $y_{nn't} = 1$ when any flow exists between tank n and tank n' in time period t, and $y_{nn't} = 0$ otherwise.

3. Model Formulation

As stated earlier, the multiperiod blend scheduling problem naturally involves binary variables $y_{nn't}$ for all the streams in each time period, and bilinearities for the mass balance constraints involving mixing. This leads to the following nonconvex mixed-integer nonlinear programming (MINLP) model (**MPBP**):

$$\text{Max} \sum_{t \in T} \left[\sum_{n \in S \cup B} \sum_{d \in D} \beta_d F_{ndt} - \sum_{s \in S} \sum_{n \in B \cup D} \beta_s F_{snt} - \sum_{nn' \in N} (\alpha_{nn'} y_{nn't} + \beta_{nn'} F_{nn't}) \right] \tag{1}$$

Subject to

$$F_{nn't} \leq F^U_{nn'} y_{nn't} \quad \forall \; nn' \in N; t \in T \tag{2a}$$

$$F_{nn't} \geq F^L_{nn'} y_{nn't} \quad \forall \; nn' \in N; t \in T \tag{2b}$$

$$C_{qbt-1} \leq C^U_{qd} + M(1 - y_{bdt}) \;\; \forall \; q \in Q; d \in D; t \in T \tag{3a}$$

$$C_{qbt-1} \geq C^L_{qd} - M(1 - y_{bdt}) \;\; \forall \; q \in Q; d \in D; t \in T \tag{3b}$$

$$C^{IN}{}_{qs} \leq C^U_{qd} + M(1 - y_{sdt}) \;\; \forall \; q \in Q; s \in S; t \in T \tag{3c}$$

$$C^{IN}{}_{qs} \geq C^L_{qd} - M(1 - y_{sdt}) \;\; \forall \; q \in Q; s \in S; t \in T \tag{3d}$$

$$I_{st} = I_{st-1} + F^{IN}{}_{st} - \sum_{n \in B \cup D} F_{snt} \quad \forall \; s \in S; t \in T \tag{4a}$$

$$I_{bt} = I_{bt-1} + \sum_{n \in S \cup B} F_{nbt} - \sum_{n \in B \cup D} F_{bnt} \quad \forall \; b \in B; t \in T \tag{4b}$$

$$I_{dt} = I_{dt-1} + \sum_{n \in S \cup B} F_{ndt} - F^{OUT}{}_{dt} \quad \forall \; d \in D; t \in T \tag{4c}$$

$$I_{bt} C_{qbt} = I_{bt-1} C_{qbt-1} + \sum_{s \in S} F_{sbt} C^{IN}{}_{qs} + \sum_{b' \in B} F_{b'bt} C_{qb't-1} - \sum_{n \in B \cup D} F_{bnt} C_{qbt-1} \qquad \forall \; q \in Q; b \in B; t \in T \tag{5}$$

$$y_{nbt} + y_{bn't} \leq 1 \quad \forall \; b \in B; n \in S \cup B; n' \in B \cup D \tag{6}$$

$$I^L_n \leq I_{nt} \leq I^U_n \quad \forall \; n \in TA; t \in T$$

$$y_{nn't} \in \{0,1\} \;\; \forall \; nn' \in N; t \in T$$

$$F_{nn't} \geq 0; \quad I_{nt} \geq 0; \quad 0 \leq C_{qbt} \leq 1 \quad \forall \; b \in B; nn' \in N; n \in TA; \; t \in T; q \in Q$$

Equation 1 is the objective function that maximizes the profit made from delivering fuel to the demand tanks, minus the costs associated with supply flows as well as fixed and variable costs of pumping the fuel between tanks. Equations (2a) and (2b) ensure that upper and lower bounds on flows between tanks are enforced, and equations (3a-d) ensure that any flow into the demand tanks satisfies the quality specificationsgiven by the tank's upper and lower concentration bounds. Equations (4a-c) are total mass balances over the supply tanks, blending tanks, and demand tanks, respectively. Equation (5) is a quality balance over the blending tanks, and lastly, equation (6) ensures that flow can enter a tank, leave a tank, or neither, but never both within the same time period of the schedule. It should also be noted that all upper and lower bounds, C^U,

C^L, F^U, F^L, I^U, I^L, cost coefficients, α, β, flows into the supply tanks, ${F^{IN}}_{st}$, and flows out of the demand tanks, F^{IN}_{dt}, are constants. This renders the entire model linear, with the exception of equation (5), due to the presence of the nonconvex bilinear terms $F \cdot C$ and $I \cdot C$, and the presence of the binary variable $y_{nn\prime t}$ in equations (1), (2a-b), (3a-d), and (6).

4. Motivating Examples

In this section we present first a very small example whose solution can be analyzed analytically to provide some insight into the nature of the difficulties faced in solving these problems. We then present a small numerical example to illustrate the difficulties faced by standard solvers in the solution of these problems.

4.1. Small Analytical Example

In the following example, we present a small instance to illustrate some of the difficulties associated with finding feasible solutions to this problem using traditional MINLP techniques. Specifically, we present the MINLP formulation and the MILP McCormick relaxation for an instance consisting of 2 supply tanks $s_1, s_2 \in S$, 1 blending tank $b_1 \in B$, 1 demand tank $d_1 \in D$, 2 time periods $t_1, t_2 \in T$ and 1 quality $q_1 \in Q$, and with the following data: $\beta_{d_1} = 10$; $\beta_{s_1} = 1$; $\beta_{s_2} = 13$; $\alpha_{nn\prime}, \beta_{nn\prime} = 0$; $F^L_{nn\prime} = 0$; $F^U_{sb_1} = 1$; $F^U_{b_1 d_1} = 2$; $C^U_{q_1 d_1} = 0.5$; $C^L_{q_1 d_1} = 0.3$; $C^{IN}_{q_1 s_1} = 0.8$; $C^{IN}_{q_1 s_2} = 0.2$; $F^{IN}_{st_1} = 1$; $F^{IN}_{st_2} = 0$; $F^{OUT}_{d_1 t} = 0$; $I^L_n = 0$; $I^U_n = 2$; furthermore, we assume that all tanks are empty at time 0.

$$\text{Max} \sum_{t=t_1,t_2} \left[10F_{b_1 d_1 t} - F_{s_1 b_1 t} - 13F_{s_2 b_1 t}\right]$$

Subject to

$$0 \le F_{sb_1 t} \le y_{sb_1 t} \qquad \text{s} = s_{1,} s_2; \text{t} = t_1, t_2$$

$$
\begin{aligned}
&0 \le F_{b_1d_1t} \le 2y_{b_1d_1t} && t = t_1, t_2\\
&0.3 - M\left(1 - y_{b_1d_1t}\right) \le C_{q_1b_1t-1} \le 0.5 + M\left(1 - y_{b_1d_1t}\right) && t = t_1, t_2\\
&I_{st_1} = 1 - F_{sb_1t_1} && s = s_1, s_2\\
&I_{st_2} = I_{st_1} - F_{sb_1t_2} && s = s_1, s_2\\
&I_{b_1t_1} = \sum_{s=s_1,s_2} F_{sb_1t_1} - F_{b_1d_1t_1}\\
&I_{b_1t_2} = I_{bt_1} + \sum_{s=s_1,s_2} F_{sb_1t_2} - F_{b_1d_1t_2}\\
&I_{d_1t_1} = F_{b_1d_1t_1}\\
&I_{d_1t_2} = I_{d_1t_1} + F_{b_1d_1t_2}\\
&I_{b_1t_1}C_{q_1b_1t_1} = \sum_{s=s_1,s_2} F_{sb_1t_1}C_{q_1s}^{IN}\\
&I_{b_1t_2}C_{q_1b_1t_2} = I_{b_1t_1}C_{q_1b_1t_1} + \sum_{s=s_1,s_2} F_{sb_1t_2}C_{q_1s}^{IN} - F_{b_1d_1t_2}C_{q_1b_1t_1}\\
&y_{sb_1t} + y_{b_1d_1t} \le 1 \quad s = s_1, s_2; t = t_1, t_2\\
&0 \le I_{nt} \le 2 \quad n = s_1, s_2, b_1, d_1;\ t = t_1, t_2\\
&0.2 \le C_{q_1b_1t} \le 0.8 \quad t = t_1, t_2\\
&y_{sb_1t}, y_{b_1d_1t} \in \{0,1\} \quad s = s_1, s_2; t = t_1, t_2
\end{aligned}
\tag{5'}
$$

The optimal value to this problem is 6, with a solution of $F_{b_1d_1t} = [0,2]$; $F_{s_1b_1t} = [1,0]$; $F_{s_2b_1t} = [1,0]$; $y_{s_1b_1t} = [1,0]$; $y_{s_2b_1t} = [1,0]$; ; $y_{b_1d_1t} = [0,1]$; $I_{s_1t}, I_{s_2t} = [0,0]$; $I_{d_1t} = [0,2]$; $I_{b_1t} = [2,0]$; $C_{q_1b_1t} = [0.5,0]$. The optimizer is thus transferring 1 unit of flow from each supply stream (for a total of 2 units) into the blend tank in time period 1 such that the quality of the resulting material is fit to discharge to the demand stream (blended material has quality of 0.5), and discharging 2 units of blended material in time period 2.

If we attempt to solve the above MINLP through traditional global optimization techniques such as spatial B&B (for example), we must first replace the bilinear set of constraints (5′) by a set of bounding convex constraints. Although there are multiple ways of doing so, we propose (in typical fashion) to replace the non-convex constraints with their appropriate set of McCormick envelopes. We thus replace (5′) with the following set of constraints:

$$u_{q_1b_1t_1} = \sum_{s=s_1,s_2} F_{sb_1t_1} C^{IN}_{q_1s}$$

$$u_{q_1b_1t_2} = u_{q_1b_1t_1} + \sum_{s=s_1,s_2} F_{sb_1t_2} C^{IN}_{q_1s} - w_{q_1b_1d_1t_2} \tag{5′b}$$

$$u_{q_1b_1t} \geq 0.2\, I_{b_1t} \quad \mathrm{t} = t_1, t_2 \tag{5′c}$$

$$u_{q_1b_1t} \geq 0.8\, I_{b_1t} + 2\, C_{q_1b_1t} - (2)(0.8) \quad \mathrm{t} = t_1, t_2 \tag{5′d}$$

$$u_{q_1b_1t} \leq 0.2\, I_{b_1t} + 2\, C_{q_1b_1t} - (0.2)(2) \quad \mathrm{t} = t_1, t_2 \tag{5′e}$$

$$u_{q_1b_1t} \leq 0.8\, I_{b_1t} \quad \mathrm{t} = t_1, t_2 \tag{5′f}$$

$$w_{q_1b_1d_1t_2} \geq 0.2\, F_{b_1d_1t_2} \tag{5′g}$$

$$w_{q_1b_1d_1t_2} \geq 0.8\, F_{b_1d_1t_2} + 2C_{q_1b_1t_1} - (2)(0.8) \tag{5′h}$$

$$w_{q_1b_1d_1t_2} \leq 0.2\, F_{b_1d_1t_2} + 2C_{q_1b_1t_1} - (0.2)(2) \tag{5′i}$$

$$w_{q_1b_1d_1t_2} \leq 0.8\, F_{b_1d_1t_2} \tag{5′j}$$

If constraints (5′) are replaced by constraints (5′a) – (5′j), and the resulting McCormick MILP relaxation is solved, the optimal value to this problem is 9, with a solution of $F_{b_1d_1t} = [0,1]$; $F_{s_1b_1t} = [1,0]$; $F_{s_2b_1t} = [0,0]$; $y_{s_1b_1t} = [1,0]$; $y_{s_2b_1t} = [0,0]$; ; $y_{b_1d_1t} = [0,1]$; $I_{s_1t}, I_{s_2t} = [0,1]$; $I_{d_1t} = [0,1]$; $I_{b_1t} = [1,0]$; $u_{q_1b_1t} = [0.8, 0]$; $w_{q_1b_1d_1t} = [0, 0.8]$ $C_{q_1b_1t} = [0.5,0]$. The optimizer is thus transferring 1 unit of flow from the cheaper supply stream (stream 1) into the blend tank in time period 1 and discharging that same unit in time period 2 to the demand stream. Note however that this solution is MINLP infeasible, since no blending in the tank occurs, and the quality of stream 1 is "off-spec" ($C^{IN}_{q_1s_1} = 0.8$) relative to the demand requirements if a discharge is to occur ($0.3 \leq C_{q_1b_1t} \leq 0.5$). In essence, constraint (5′) in the MINLP is being violated in the MILP relaxation. Furthermore, due to the nature of the objective function that incentivizes the optimizer to exploit the "gap" that exists between the bilinear set of constraints (5′) and their McCormick relaxation (5′a) – (5′j), it is often the case in practice that traditional techniques used for non-convex MINLP problems struggle to find feasible solutions after many iterations/nodes. In the following section, we show this for larger instances of the multiperiod blend scheduling problem using existing global optimization solvers, which motivates the need for developing novel methods to address this issue. In Section 5, we discuss such novel methods based on a radix-based discretization scheme that addresses this issue in a conceptually different manner than traditional techniques. Indeed, instead of relying on an MILP relaxation that is necessarily feasible – assuming an MINLP feasible solution exists – and trying to obtain an MINLP feasible solution subsequently, we develop an MILP restriction (by discretizing) that is necessarily MINLP feasible –assuming an MILP feasible solution exists. This change in perspective seems to be a promising strategy for solving the multiperiod blend scheduling problem, as demonstrated in section 7 on a series of randomized examples.

4.2. Small Numerical Example

Table 1: Supply tank specifications

	C_s (% mass)		F_s (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	$\boldsymbol{\beta_s}$ **($/kg)**
1	0.4	0.1	1.0	0.1	0.4	0.1
2	0.1	0.9	0.6	0.2	0.8	0.2

Table 2: Demand tank specifications

	Bounds on C (% mass)		F_d (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	$\boldsymbol{\beta_d}$ **($/kg)**
7	0.1 - 0.4	0.2 - 1.0	0.02	0.17	0.73	5.1
8	0.2 - 0.9	0.4 - 0.7	0.04	0.65	0.65	4.0

Table 3: Initial conditions and costs of flows

		C_0 (% mass)		Cost Coefficients α (10^3 $), β ($/ kg) to Tank					
Tank	$\boldsymbol{I_0}$	**Qual. 1**	**Qual. 2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0.3	--	--	0.30, 0.47	0.23, 0.84	0.19, 0.23	0.17, 0.23	--	0.44, 0.31
2	1.7	--	--	--	0.92, 0.43	0.18, 0.90	0.98, 0.44	--	0.11, 0.26
3	1.2	0.5	0.3	--	0.41, 0.59	0.26, 0.60	0.71, 0.22	0.12, 0.30	0.32, 0.42
4	1.1	0.9	0.4	0.51, 0.09	--	0.26, 0.80	0.03, 0.93	0.73, 0.49	0.58, 0.24
5	0.3	0.1	0.8	0.46, 0.96	0.55, 0.52	--	0.23, 0.49	0.62, 0.68	0.40, 0.37
6	1.7	0.4	0.2	0.99, 0.04	0.89, 0.91	0.80, 0.10	--	0.26, 0.34	0.68, 0.14
7	1.2	--	--	--	--	--	--	--	--
8	0.7	--	--	--	--	--	--	--	--

To provide some insight into the nature of the computational difficulties encountered in the multiperiod blend scheduling problem, we consider a small example. Assume the blending system shown in Figure 2 represents the gasoline blending unit in a small refinery. It contains 8 tanks, and is to be optimized over 3 time periods (24 hours each), and contains 2 qualities (properties) of importance. Two supply tanks are filled over time with incoming fuel from the distillation unit, and feed four blending tanks to meet the demand and specifications of two products. Given the network topology in Figure 2, and all associated values given in Tables 1, 2, and 3, the schedule for blending the fuels is sought that maximizes total profit.

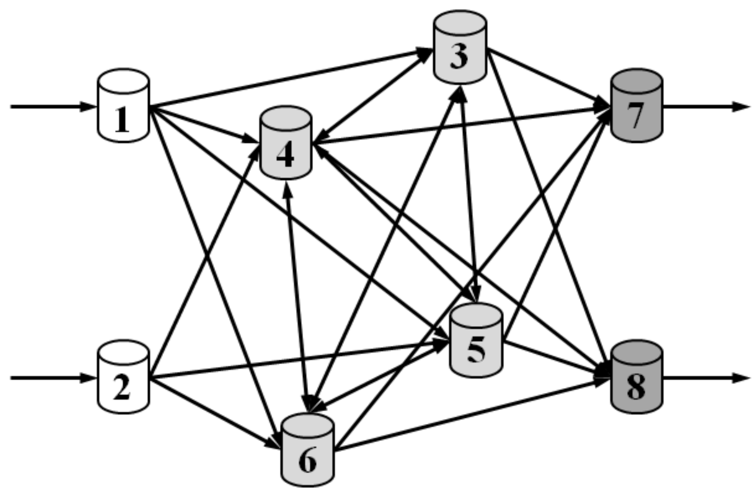


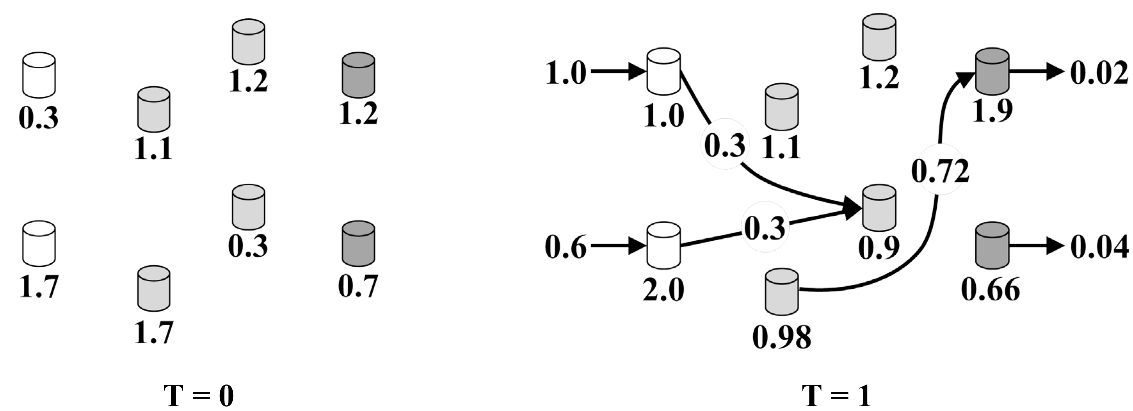
Figure 2: Example Network Topology

The nonconvex MINLP model was implemented in the modeling language GAMS 23.8.1 (Brook, Kendrick, and Meeraus, 1988) and solved using a variety of solvers. The global solution has an objective function of $13,527, and the flow schedule is shown in Figure 3. Note that only the blending tanks have the restriction that flows cannot simultaneously receive and deliver flow, because the concentration in these tanks can vary. This is not true of the supply and demand tanks. The planned tank transfers quickly fill the demand tanks to their maximum inventory in just three time periods. Three commercially available MINLP solvers that guarantee global optima, BARON 10.2.0 (Sahinidis, 1996), LINDOGlobal (LINDO API, 2011), and GloMIQO (Misener & Floudas, 2012) were used to solve the 8 tank problem. As seen in Table 4, LINDOGlobal was able to converge in 96 minutes, while BARON and GloMIQO were both unable to converge to the global optimum within two hours. Two other solvers that do not guarantee global optima but solve MINLPs under convexity assumptions, DICOPT (Grossmann, Viswanathan, Vecchietti, Raman, & Kalvelagen, 2003) and sBB (Bussieck & Drud, 2001), failed to find the global optimum. Note that due to the nonconvexity of the problem, the outer approximation (OA) algorithm used in DICOPT has particular difficulty even finding a feasible solution as was discussed in section 4.1. The results for these solvers are shown in Table 4. All computations were performed on a dual CPU computer with two Intel Xeon X5650 processors at 2.66GHz each, 16 GB of RAM, and running Ubuntu Linux 10.04.

Table 4: Results of solving the 8 tank example problem

	BARON	**LINDOGlobal**	**GloMIQO**	**DICOPT**	**sBB**
Wall Time (s)	>7200	5788.5	>7200	>7200	17.0
Objective Function Value ($)	13.104	13.527	13.026	*	13.443

* No feasible solution found



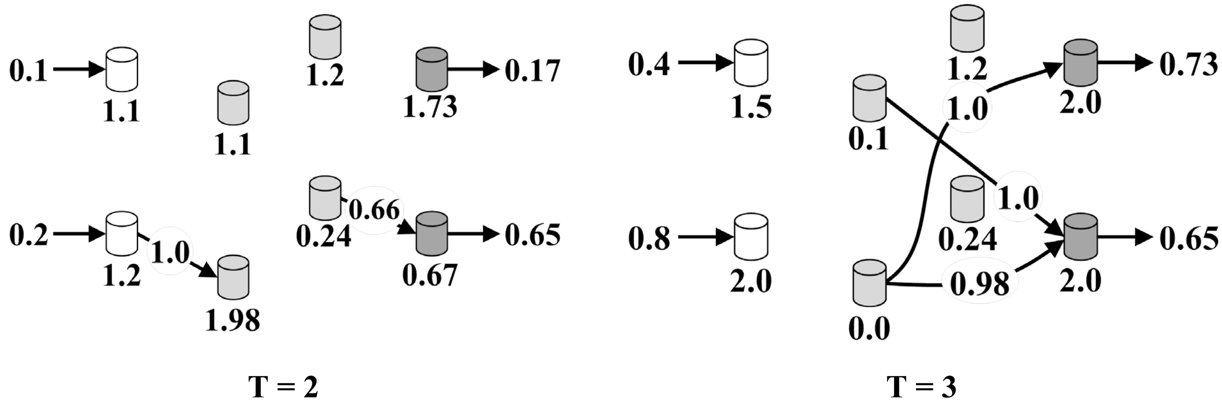


Figure 3: Solution to example 8 tank problem. Flows and inventories are in 10^3 kg.

Table 5: Results of BARON solving 7 sample multiperiod blend scheduling problems

Tanks	6	8	8	8	8	8	8
Time Periods	3	3	3	3	4	4	4
Qualities	2	2	2	2	2	2	2
Wall Time (s)	21.12	>7200	>7200	>7200	>7200	>7200	>7200
Relative Gap	52.3%	23.7%	205.6%	13.4%	13.4%	409.5%	17.2%
Constraints	214	625	607	628	861	885	737
Continuous Variables	67	136	136	136	185	189	169
Binary Variables	36	87	87	87	120	124	104
Non-Zero Elements	543	1722	1672	1766	2413	2469	2107
Nonlinear Non-Zeros	64	256	244	256	376	376	358

This problem contains only 8 tanks, 3 time periods, and 2 qualities, and yet takes over one hour to solve to global optimality when the solver is able to converge at all. Similar problems with different numbers of tanks, qualities, and time periods yield comparable results when solved with BARON (see Table 5). Real-world applications contain a larger number of tanks, qualities, and time periods. Additionally, it can be useful for its extension to a two-stage stochastic programming model, which would greatly increase its size. In order to solve larger problems, it is clear that a new approach is needed.

5. Radix-Based Discretization

We introduce in this section our basic approach to the solution of the multiperiod blend scheduling problem, a technique known as radix-based discretization, or multiparametric disaggregation as coined in the paper by Teles, Castro, & Matos (2011). This technique, and the algorithms that follow, have been refined and described in a recent paper by Kolodziej et al. (2012). As we treat the technique primarily as a discretization technique, we will call it radix-based discretization.

The general principle behind radix-based discretization is an equally-spaced discretization based on powers of some radix or numerical base. This results in a linear increase of binary variables for an order of magnitude increase in discretization precision. For example, a traditional discretization technique over a set of points $a_m, m = 1..M$, would take the following form,

$$x = \sum_{m=1}^{M} a_m \cdot y_m \tag{7}$$

where $y_m \in \{0,1\}$, $y_m = 1$ if $x = a_m$ and $y_m = 0$ otherwise, and $m = \{1,..M\}$ defining the number of discretization points. In contrast, the radix-based discretization technique takes the form

$$x = \sum_{\ell=p}^{P} \sum_{k=0}^{R-1} R^{\ell} \cdot k \cdot y_{k\ell} \tag{8}$$

where R is the chosen numerical base, p is the smallest power of R to be considered (the finest level of discretization), and P is the largest power of R (which provides an upper bound on the discretization). Most commonly, a numerical base of 10 is chosen, which results in the following form:

$$x = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \tag{9}$$

This form allows for a very intuitive analogy where the first summation ($\ell = \{p, .. P\}$) represents the place value and the second summation represents each digit in the decimal system {1,..9}. Some properties of this approach are shown in Figure 4 and Table 6 using $R = 10$, $P = 0$, and $p = \{0, -1, -2\}$.

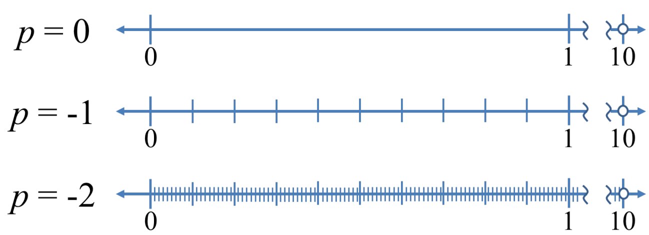


Figure 4: Discretized axes using radix-based discretization for $P = 0$, $p = \{0,-1,-2\}$.

Table 6: Characteristics of Radix-Based (Base 10) Discretization at $P = 0$, $p = \{0,-1,-2\}$.

p (Smallest Power)	0	-1	-2
P (Largest Power)	0	0	0
Range	0-9	0-9.9	0-9.99
Increment	1	0.1	0.01
Significant Digits	1	2	3
Binary Variables (RBD)	10	20	30
Binary Variables (Traditionally)	10	100	1000

For the least number of binary variables, a numerical base of 2 can be chosen. This requires a minor modification to the discretization formula to convert back to a decimal system:

$$x = 10^{\Pi} \sum_{\ell=p}^{P} \sum_{k=0}^{1} 2^{\ell} \cdot k \tag{10}$$

In this expression, Π is the power of 10 such that $10^{\Pi} \cdot 2^{P} \geq x_j^U$, where x_j^U is the upper bound on x_j. The properties of this binary (base 2) approach are shown below in Table 7 for $p = 0$ and $P = \{4,7,10\}$. Note that for approximately the same precision as the discretization levels used in Table 6, fewer binary variables are required.

Table 7: Characteristics of Radix-Based (Base 2) Discretization at $p = 0$, $P = \{4,7,10\}$

p (Smallest Power)	0	0	0
P (Largest Power)	4	7	10
Range	0-15	0-12.7	0-10.23
Increment	1	0.1	0.01
Significant Digits	1	2	3
Binary Variables (RBD)	8	14	20
Binary Variables (Traditionally)	10	100	1000

For the purpose of this technique, maximizing bilinear programs can be written in the following form (Kolodziej et al., 2012):

$$\text{Max } z = f_0(x, y)$$

Subject to

$$f_q(x, y) \le 0 \quad q \in Q \backslash \{0\}$$

$$f_q(x, y) = \sum_{(i,j) \in BL_q} a_{ijq} x_i x_j + h_q(x, y) \quad q \in Q \qquad \textbf{(P)}$$

$$x \in S \cap \Omega \subset \mathbb{R}^n$$

$$y \in \{0,1\}$$

where $h_q(x,y)$ is convex and twice differentiable, a_{ijq} is a scalar with $i \in I$, $j \in J$, and $q \in Q$ represents the set of all functions f_q, including the objective function f_0 and all constraints. BL_q is an (i,j)-index set which defines the bilinear terms present in the problem. The set $S \subset \mathbb{R}^n$ is convex, and $\Omega \subset \mathbb{R}^n$ is an n-dimensional hyperrectangle defined in terms of the initial variable bounds x^L and x^U:

$$\Omega = \{x \in \mathbb{R}^n : 0 \le x^L \le x \le x^U\}$$

This problem can be reformulated by discretizing one of the variables involved in each bilinearity, x_j (Kolodziej et al., 2012). Using this discretization, which is summarized in Appendix A, we can then reformulate the problem (**P**) into an MILP approximation of the original problem (**P**):

$$\text{Max } z' = f_0(x,y) = \sum_{(i,j)\in BL_0} a_{ij0} w_{ij} + h_0(x,y)$$

Subject to

$$f_q(x,y) = \sum_{(i,j)\in BL_q} a_{ijq} w_{ij} + h_q(x,y) \le 0 \quad q \in Q\backslash\{0\}$$

$$w_{ij} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{x}_{ijk\ell} \quad \forall \ (i,j) \in BL_q; q \in Q$$

$$x_j = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot z_{jk\ell} \quad \forall \ j \in \{j|(i,j) \in BL_q, q \in Q\}$$

(P′)

$$\hat{x}_{ijk\ell} \le x_i^U \cdot z_{jk\ell} \quad \forall \ (i,j) \in BL_q; q \in Q; k \in K; \ell \in L$$

$$\hat{x}_{ijk\ell} \ge x_i^L \cdot z_{jk\ell} \quad \forall \ (i,j) \in BL_q; q \in Q; k \in K; \ell \in L$$

$$\sum_{k=0}^{9} \hat{x}_{ijk\ell} = x_i \quad \forall \ (i,j) \in BL_q; q \in Q; k \in K; \ell \in L$$

$$\sum_{k=0}^{9} z_{jk\ell} = 1 \quad \forall \ j \in \{j|(i,j) \in BL_q, q \in Q\}; k \in K; \ell \in L$$

$$z_{jk\ell} \in \{0,1\} \quad \forall \ j \in \{j|(i,j) \in BL_q, q \in Q\}; k \in K; \ell \in L$$

$$x \in S \cap \Omega \subset \mathbb{R}^n$$

$$y \in \{0,1\}$$

This formulation discretizes one variable in the bilinear term using the radix-based discretization scheme described earlier. x_j is a vector of discretized variables that are involved in a bilinear term and w_{ij} represents the bilinear products such that $w_{ij} = x_i \cdot x_j$. $z_{jk\ell}$ is a binary variable introduced in the discretization, such that $z_{jk\ell} = 1$ if the digit k in the ℓ^{th} power of 10's place of x_j is activated, and $z_{jk\ell} = 0$ otherwise.

By adding a pseudo-slack term to the above discretized problem to fill the gap between discretization points, we can derive a relaxation of the original problem (**P**) that is also an MILP:

$$\text{Max } z^R = f_0(x,y) = \sum_{(i,j)\in BL_0} a_{ij0} w_{ij} + h_0(x,y) \qquad \textbf{(PR)}$$

Subject to

$$f_q(x,y) = \sum_{(i,j)\in BL_q} a_{ijq} w_{ij} + h_q(x,y) \leq 0 \quad q \in Q\backslash\{0\}$$

$$w_{ij} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{x}_{ijk\ell} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{x}_{ijk} \quad \forall \ (i,j) \in BL_q; q \in Q$$

$$x_j = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot z_{ijk\ell} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{z}_{ijk} \quad \forall j \in \{j|(i,j) \in BL_q, q \in Q\}$$

$$\hat{x}_{ijk\ell} \leq x_i^U \cdot z_{ijk\ell} \quad \forall \ (i,j) \in BL_q; q \in Q; \ell \in L; k \in K$$

$$\hat{x}_{ijk\ell} \geq x_i^L \cdot z_{ijk\ell} \quad \forall \ (i,j) \in BL_q; q \in Q; \ell \in L; k \in K$$

$$\tilde{x}_{ijk} \leq x_i^U \cdot \tilde{z}_{ijk} \quad \forall \ (i,j) \in BL_q; q \in Q; k \in \{0,1\}$$

$$\tilde{x}_{ijk} \geq x_i^L \cdot \tilde{z}_{ijk} \quad \forall \ (i,j) \in BL_q; q \in Q; k \in \{0,1\}$$

$$\sum_{k=0}^{9} \hat{x}_{ijk\ell} = x_i \quad \forall \ (i,j) \in BL_q; q \in Q; \ell \in L$$

$$\sum_{k=0}^{1} \tilde{x}_{ijk} = x_i \quad \forall \ (i,j) \in BL_q; q \in Q$$

$$\sum_{k=0}^{9} z_{ijk\ell} = 1 \quad \forall \ (i,j) \in BL_q; q \in Q; \ell \in L$$

$$\sum_{k=0}^{1} \tilde{z}_{ijk} = 1 \quad \forall (i,j) \in BL_q; q \in Q$$

$$z_{ijk\ell} \in \{0,1\} \quad \forall \ (i,j) \in BL_q; q \in Q; \ell \in L; k \in K$$

$$0 \leq \tilde{z}_{ijk} \leq 1 \quad \forall \ (i,j) \in BL_q; q \in Q; k \in \{0,1\}$$

$$x \in S \cap \Omega \subset \mathbb{R}^n$$

$$y \in \{0,1\}$$

This problem (**PR**) is a relaxation of problem (**P**) because it includes at least (and in most cases more than) the entire feasible region of problem (**P**). The bilinear terms have been relaxed by adding continuous terms to the discretized terms, and constraints have been added (similar to the constraints from the discretized problem) to incorporate these continuous terms into the formulation. In other words, $\tilde{x}_{ijk}$ and $\tilde{z}_{ijk}$ are analogous to $\hat{x}_{ijk\ell}$ and $z_{ijk\ell}$, respectively, and require similar constraints.

6. Improved Global Optimization of the Multiperiod Blend Scheduling Problem

Because the multiperiod MINLP blending problem has bilinear terms, the MINLP formulations (**P**) and (**P′**) in Section 4 can be applied to this problem and solved in a significantly decreased amount of time as will be shown later in the paper.

6.1. Model Reformulation – Decimal

In order to utilize the global optimization algorithms, problems (**MPBP′**) and (**MPBPR**) (the discretized and relaxed problems, respectively) must be derived. Two bilinear terms, $F_{bnt}C_{qbt}$ and $I_{bt}C_{qbt}$, must be reformulated to mixed-integer linear constraints as was shown in Section 4. Additionally, since C_{qbt} is present in both bilinear terms, it is a natural choice for discretization.

We start with individually reformulating the bilinear terms $w^{FC}_{qbnt} = F_{bnt}C_{qbt}$ and $w^{IC}_{qbt} = I_{bt}C_{qbt}$. For simplicity, we will denote the bilinearities $w^{FC} = F \cdot C$ and $w^{IC} = I \cdot C$:

$$w^{FC} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{F}_{k\ell} \tag{11}$$

$$C = \sum_{\ell=p}^{P} \sum_{j=0}^{9} 10^{\ell} \cdot k \cdot z_{k\ell} \tag{12}$$

$$\hat{F}_{k\ell} \leq F^{U} \cdot z_{k\ell} \qquad \forall\ \ell \in \{p, \ldots, P\}; k \in \{0, \ldots, 9\} \tag{13}$$

$$\sum_{k=0}^{9} \hat{F}_{k\ell} = F \qquad \forall\ \ell \in \{p, \ldots, P\} \tag{14}$$

$$\sum_{k=0}^{9} z_{k\ell} = 1 \qquad \forall\ \ell \in \{p, \ldots, P\} \tag{15}$$

$$\hat{F}_{k\ell} \geq 0 \qquad \forall\ \ell \in \{p, \ldots, P\}; k \in \{0, \ldots, 9\}$$

$$z_{k\ell} \in \{0,1\}$$

$$w^{IC} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{I}_{k\ell} \tag{16}$$

$$C = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot z_{k\ell} \tag{17}$$

$$\hat{I}_{k\ell} \leq I^{U} \cdot z_{k\ell} \qquad \forall\ \ell \in \{p, \ldots, P\}; k \in \{0, \ldots, 9\} \tag{18}$$

$$\sum_{k=0}^{9} \hat{I}_{k\ell} = I \qquad \forall\ \ell \in \{p, \ldots, P\} \tag{19}$$

$$\sum_{k=0}^{9} z_{k\ell} = 1 \qquad \forall\ \ell \in \{p, \ldots, P\} \tag{20}$$

$$\hat{I}_{k\ell} \geq 0 \qquad \forall\ \ell \in \{p, \ldots, P\}, k \in \{0, \ldots, 9\}$$

$$z_{k\ell} \in \{0,1\}$$

Note that equations 12 and 17 are identical, as are equations 15 and 20. Thus, we can remove both equations 17 and 20. Integrating constraints (11)-(16), (18)-(19) into problem (**MPBP**), we obtain the final form of (**MPBP′**):

$$\text{Max} \sum_{t \in T} \left[\sum_{n \in S \cup B} \sum_{d \in D} \beta_d F_{ndt} - \sum_{s \in S} \sum_{n \in B \cup D} \beta_s F_{snt} - \sum_{nn' \in N} (\alpha_{nn'} y_{nn't} + \beta_{nn'} F_{nn't}) \right]$$

Subject to

$$F_{nn't} \leq F^{U}_{nn'} y_{nn't} \quad \forall\ nn' \in N; t \in T$$

$$F_{nn't} \geq F^{L}_{nn'} y_{nn't} \quad \forall\ nn' \in N; t \in T$$

$$C_{qbt-1} \leq C^{U}_{qd} + M(1 - y_{bdt}) \quad \forall\ q \in Q; d \in D; t \in T$$

$$C_{qbt-1} \geq C^{L}_{qd} - M(1 - y_{bdt}) \quad \forall\ q \in Q; d \in D; t \in T$$

$$C^{IN}{}_{qs} \leq C^{U}_{qd} + M(1 - y_{sdt}) \quad \forall\ q \in Q; s \in S; t \in T$$

$$C^{IN}{}_{qs} \geq C^{L}_{qd} - M(1 - y_{sdt}) \quad \forall\ q \in Q; s \in S; t \in T$$

$$I_{st} = I_{st-1} + F^{IN}{}_{st} - \sum_{n \in B \cup D} F_{snt} \quad \forall\ s \in S; t \in T$$

$$I_{bt} = I_{bt-1} + \sum_{n \in S \cup B} F_{nbt} - \sum_{n \in B \cup D} F_{bnt} \quad \forall\ b \in B; t \in T \qquad (\textbf{MPBP}')$$

$$I_{dt} = I_{dt-1} + \sum_{n \in S \cup B} F_{ndt} - F^{OUT}{}_{dt} \quad \forall\ d \in D; t \in T$$

$$w^{IC}_{qbt} = w^{IC}_{qbt-1} + \sum_{s \in S} F_{sbt} C^{IN}{}_{qs} + \sum_{b' \in B} w^{FC}_{qb'bt} - \sum_{n \in B \cup D} w^{FC}_{qbnt} \quad \forall\ q \in Q; b \in B; t \in T$$

$$y_{nbt} + y_{bn't} \leq 1 \quad \forall\ b \in B; n \in S \cup B; n' \in B \cup D$$

$$w^{FC}_{qbnt} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{F}_{k\ell qbnt} \quad \forall\ q \in Q; b \in B; n \in B \cup D; t \in T$$

$$C_{qbt} = \sum_{\ell=p}^{P} \sum_{j=0}^{9} 10^{\ell} \cdot k \cdot z_{k\ell qbt} \quad \forall\ q \in Q; b \in B; t \in T$$

$$\hat{F}_{k\ell qbnt} \leq F^{U}_{bn} \cdot z_{k\ell qbt} \quad \forall\ \ell \in \{p, \ldots, P\}; k \in \{0, \ldots, 9\}; q \in Q; b \in B; n \in B \cup D; t \in T$$

$$\sum_{k=0}^{9} \hat{F}_{k\ell qbnt} = F_{bnt} \quad \forall\ \ell \in \{p,\dots,P\}; q \in Q; b \in B; n \in B \cup D; t \in T$$
$$\sum_{k=0}^{9} z_{k\ell qbt} = 1 \quad \forall\ \ell \in \{p,\dots,P\}; q \in Q; b \in B; t \in T$$
$$w_{qbt}^{IC} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{I}_{k\ell qbt} \quad \forall\ q \in Q; b \in B; t \in T$$
$$\hat{I}_{k\ell qbt} \le I_{bt}^{U} \cdot z_{k\ell qbt} \quad \forall\ \ell \in \{p,\dots,P\}; k \in \{0,\dots,9\}; q \in Q; b \in B; t \in T$$
$$\sum_{k=0}^{9} \hat{I}_{k\ell qbt} = I_{bt} \quad \forall\ \ell \in \{p,\dots,P\}; q \in Q; b \in B; t \in T$$
$$I_n^L \le I_{nt} \le I_n^U \quad \forall\ n \in TA; t \in T$$
$$y_{nn't} \in \{0,1\} \quad \forall\ nn' \in N; t \in T$$
$$F_{nn't} \ge 0; \quad I_{nt} \ge 0; \quad 0 \le C_{qbt} \le 1 \quad \forall\ b \in B; nn' \in N; n \in B \cup D;\ t \in T; q \in Q$$
$$\hat{I}_{k\ell qbt} \ge 0 \quad \hat{F}_{k\ell qbnt} \ge 0 \quad \forall\ \ell \in \{p,\dots,P\}; k \in \{0,\dots,9\}; q \in Q; b \in B; n \in B \cup D; t \in T$$
$$z_{k\ell qbt} \in \{0,1\} \quad \forall\ \ell \in \{p,\dots,P\}; k \in \{0,\dots,9\}; q \in Q; b \in B; t \in T$$

Because Problem (**MPBP′**) is a maximization problem, it yields a lower bound on (**MPBP**) provided that it finds a feasible solution. Next, we can similarly derive problem (**MPBPR**), which yields a relaxation (and thus an upper bound) of (**MPBP**).

$$\text{Max} \sum_{t\in T} \left[\sum_{n\in S\cup B} \sum_{d\in D} \beta_d F_{ndt} - \sum_{s\in S} \sum_{n\in B\cup D} \beta_s F_{snt} - \sum_{nn'\in N} (\alpha_{nn'} y_{nn't} + \beta_{nn'} F_{nn't}) \right]$$

Subject to

$$F_{nn't} \le F_{nn'}^{U} y_{nn't} \quad \forall\ nn' \in N; t \in T$$
$$F_{nn't} \ge F_{nn'}^{L} y_{nn't} \quad \forall\ nn' \in N; t \in T$$
$$C_{qbt-1} \le C_{qd}^{U} + M(1 - y_{bdt}) \quad \forall\ q \in Q; d \in D; t \in T$$
$$C_{qbt-1} \ge C_{qd}^{L} - M(1 - y_{bdt}) \quad \forall\ q \in Q; d \in D; t \in T$$
$${C^{IN}}_{qs} \le C_{qd}^{U} + M(1 - y_{sdt}) \quad \forall\ q \in Q; s \in S; t \in T$$
$${C^{IN}}_{qs} \ge C_{qd}^{L} - M(1 - y_{sdt}) \quad \forall\ q \in Q; s \in S; t \in T \qquad \textbf{(MPBPR)}$$
$$I_{st} = I_{st-1} + {F^{IN}}_{st} - \sum_{n\in B\cup D} F_{snt} \quad \forall\ s \in S; t \in T$$
$$I_{bt} = I_{bt-1} + \sum_{n\in S\cup B} F_{nbt} - \sum_{n\in B\cup D} F_{bnt} \quad \forall\ b \in B; t \in T$$
$$I_{dt} = I_{dt-1} + \sum_{n\in S\cup B} F_{ndt} - {F^{OUT}}_{dt} \quad \forall\ d \in D; t \in T$$
$$w_{qbt}^{IC} = w_{qbt-1}^{IC} + \sum_{s\in S} F_{sbt} {C^{IN}}_{qs} + \sum_{b'\in B} w_{qb'bt}^{FC} - \sum_{n\in B\cup D} w_{qbnt}^{FC} \quad \forall\ q \in Q; b \in B; t \in T$$
$$y_{nbt} + y_{bn't} \le 1 \quad \forall\ b \in B; n \in S \cup B; n' \in B \cup D$$

$$w_{qbnt}^{FC} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{F}_{k\ell qbnt} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{F}_{kqbnt} \quad \forall\ q \in Q; b \in B; n \in B \cup D; t \in T$$

$$C_{qbt} = \sum_{\ell=p}^{P} \sum_{j=0}^{9} 10^{\ell} \cdot k \cdot z_{k\ell qbt} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{z}_{kqbt} \quad \forall\ q \in Q; b \in B; t \in T$$

$$\hat{F}_{k\ell qbnt} \leq F_{bn}^{U} \cdot z_{k\ell qbt} \quad \forall\ \ell \in \{p, \dots, P\}; k \in \{0, \dots, 9\}; q \in Q; b \in B; n \in B \cup D; t \in T$$

$$\tilde{F}_{kqbnt} \leq F_{bn}^{U} \cdot \tilde{z}_{kqbt} \quad \forall\ k \in \{0, \dots, 9\}; q \in Q; b \in B; n \in B \cup D; t \in T$$

$$\sum_{k=0}^{9} \hat{F}_{k\ell qbnt} = F_{bnt} \quad \forall\ \ell \in \{p, \dots, P\}; q \in Q; b \in B; n \in B \cup D; t \in T$$

$$\sum_{k=0}^{1} \tilde{F}_{kqbnt} = F_{bnt} \quad \forall\ q \in Q; b \in B; n \in B \cup D; t \in T$$

$$\sum_{k=0}^{9} z_{k\ell qbt} = 1 \quad \forall\ \ell \in \{p, \dots, P\}; q \in Q; b \in B; t \in T$$

$$\sum_{k=0}^{1} \tilde{z}_{kqbt} = 1 \quad \forall\ q \in Q; b \in B; t \in T$$

$$w_{qbt}^{IC} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{I}_{k\ell qbt} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{I}_{k\ell qbt} \quad \forall\ q \in Q; b \in B; t \in T$$

$$\hat{I}_{k\ell qbt} \leq I_{bt}^{U} \cdot z_{k\ell qbt} \quad \forall\ \ell \in \{p, \dots, P\}; k \in \{0, \dots, 9\}; q \in Q; b \in B; t \in T$$

$$\tilde{I}_{kqbt} \leq I_{bt}^{U} \cdot \tilde{z}_{kqbt} \quad \forall\ k \in \{0, \dots, 9\}; q \in Q; b \in B; t \in T$$

$$\sum_{k=0}^{9} \hat{I}_{k\ell qbt} = I_{bt} \quad \forall\ \ell \in \{p, \dots, P\}; q \in Q; b \in B; t \in T$$

$$\sum_{k=0}^{1} \tilde{I}_{kqbt} = I_{bt} \quad \forall\ q \in Q; b \in B; t \in T$$

$$I_{n}^{L} \leq I_{nt} \leq I_{n}^{U} \quad \forall\ n \in TA; t \in T$$

$$y_{nn't} \in \{0,1\} \quad \forall\ nn' \in N; t \in T$$

$$F_{nn't} \geq 0; \quad I_{nt} \geq 0; \quad 0 \leq C_{qbt} \leq 1 \quad \forall\ b \in B; nn' \in N; n \in TA;\ t \in T; q \in Q$$

$$\hat{I}_{k\ell qbt} \geq 0 \quad \hat{F}_{k\ell qbnt} \geq 0 \quad \forall\ \ell \in \{p, \dots, P\}; k \in \{0, \dots, 9\}; q \in Q; b \in B; n \in B \cup D; t \in T$$

$$\tilde{I}_{kqbt} \geq 0 \quad \tilde{F}_{kqbnt} \geq 0 \quad \forall\ k \in \{0, \dots, 9\}; q \in Q; b \in B; n \in B \cup D; t \in T$$

$$z_{k\ell qbt} \in \{0,1\} \quad \forall\ \ell \in \{p, \dots, P\}; k \in \{0, \dots, 9\}; q \in Q; b \in B; t \in T$$

$$0 \leq \tilde{z}_{kqbt} \leq 1 \quad \forall\ k \in \{0,1\}; q \in Q; b \in B; t \in T$$

6.2. Model Reformulation – Binary

As mentioned earlier, using a numerical base of 2 will yield the minimum number of binary variables in the discretization. While the models (**MPBP′**) and (**MPBPR**) remain largely the same, summations from $k = \{0..9\}$ are replaced with $k = \{0, 1\}$, and the constraints for C_{qbt},

w_{qbnt}^{FC}, and w_{qbt}^{IC} are replaced with the following in (**MPBP′**):

$$C_{qbt} = 10^{\Pi} \sum_{\ell=p}^{P} \sum_{j=0}^{9} 2^{\ell} \cdot k \cdot z_{k\ell qbt} \quad \forall \ q \in Q; b \in B; t \in T \tag{21}$$

$$w_{qbnt}^{FC} = 10^{\Pi} \sum_{\ell=p}^{P} \sum_{k=0}^{9} 2^{\ell} \cdot k \cdot \hat{F}_{k\ell qbnt} \quad \forall \ q \in Q; b \in B; n \in TA; t \in T \tag{22}$$

$$w_{qbt}^{IC} = 10^{\Pi} \sum_{\ell=p}^{P} \sum_{k=0}^{9} 2^{\ell} \cdot k \cdot \hat{I}_{k\ell qbt} \quad \forall \ q \in Q; b \in B; t \in T \tag{23}$$

And in (**MPBPR**), these constraints are replaced with

$$C_{qbt} = 10^{\Pi} \sum_{\ell=p}^{P} \sum_{j=0}^{9} 2^{\ell} \cdot k \cdot z_{k\ell qbt} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{z}_{kqbt} \qquad \forall \ q \in Q; b \in B; t \in T \tag{24}$$

$$w_{qbnt}^{FC} = 10^{\Pi} \sum_{\ell=p}^{P} \sum_{k=0}^{9} 2^{\ell} \cdot k \cdot \hat{F}_{k\ell qbnt} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{F}_{kqbnt} \quad \forall \ q \in Q; b \in B; n \in TA; t \in T \tag{25}$$

$$w_{qbt}^{IC} = 10^{\Pi} \sum_{\ell=p}^{P} \sum_{k=0}^{9} 2^{\ell} \cdot k \cdot \hat{I}_{k\ell qbt} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{I}_{k\ell qbt} \quad \forall \ q \in Q; b \in B; t \in T \tag{26}$$

Where the power of 10 in these expressions, Π, is such that $10^{\Pi} \cdot 2^{P} \geq C_{qbt}^{U}$.

6.3. Global Optimization Algorithms

In order to use the previously described formulations to solve the multiperiod blend scheduling problem, various heuristics and global optimization algorithms can be used. The simplest and perhaps most intuitive method is as follows:

Heuristic 1

Step 0. Choose $p = P \geq \lceil log_{10} x_j^U \rceil$

Step 1. Solve (**P′**) to obtain an approximation of the solution z'.

Step 2. Fix the binary variables y in (**P**) to the values found by the solution of (**P′**) in Step 1, and solve (**P**) with these fixed binary variables using a local (or global) NLP algorithm to obtain some z using the solution to (**P′**) as a starting point. If (**P′**) is infeasible, let $z' = +\infty$.

Step 3. Optionally, set $p = p - 1$, and return to step 1.

Heuristic 1 first solves the discretized problem, (**P′**), to obtain an approximate solution. However, as this solution is limited to the discretization points, we can assume the process binary variables y (as opposed to the binary variables introduced by the discretization, which are ignored) to be correct, fix them, and solve the resulting NLP. This method, while simple and intuitive, is strictly a heuristic method, as it lacks any rigorous termination criteria, and the assumption that the correct process binary variables were chosen could be incorrect. The accuracy of the solution can be increased by decreasing p, but p must still be chosen a priori. Larger values of p will result in smaller problems (fewer discretization points), but in a less accurate solution. Smaller values of p will conversely result in higher accuracy, but larger problems. However, rigorous algorithms can be derived as shown below.

As (**P′**) is an inner approximation of the problem, and (**PR**) is a relaxation, they correspond to a lower and upper bound, respectively, on problem (**P**) when maximizing. Thus, two global optimization algorithms can be introduced to allow the solution of problem (**P**) to some tolerance ε. The first solves the upper and lower bounding problems in tandem:

Algorithm 1

Step 0. Choose $p = P \geq \lceil log_{10} x_j^U \rceil$. Set $z^L = -\infty$.

Step 1. Solve (**PR**) to obtain the upper bound z^R.

Step 2. Solve (**P′**) to obtain the lower bound z'. If (**P′**) is infeasible, let $z' = -\infty$. Update overall lower bound $z^L = \max\{ z', z^L\}$

Step 3. If $(z^R - z^L)/z^R \leq \varepsilon$, STOP, the solution is globally optimal. Otherwise, set $p = p - 1$, and return to step 1.

This algorithm is intuitive, but has the major shortcoming that it must solve two MILPs at each iteration. To improve performance, the lower bounding problem in step 2 can be replaced by the use of a local NLP solver after fixing the binary variables y, yielding Algorithm 2:

Algorithm 2

Step 0. Choose $p = P \geq \lceil log_{10} x_j^U \rceil$

Step 1. Solve (**PR**) to obtain the upper bound z^R.

Step 2. Fix the binary variables y in (**P**) to the values found by the solution of (**PR**) in Step 1, reducing it to an NLP. Solve (**P**) with these fixed binary variables using a local NLP algorithm to obtain some lower bound z^L using the solution to (**PR**) as a starting point.

Step 3. If $(z^R - z^L)/z^R \leq \varepsilon$, STOP, the solution is globally optimal. Otherwise, set $p = p - 1$ and return to step 1.

These algorithms have been shown to be effective in solving nonconvex MINLPs to global optimality in significantly less time than the leading global optimization solvers, namely BARON and GloMIQO. In the following sections, we apply Heuristic 1, and Algorithms 1 and 2, in addition to BARON and GloMIQO, to the multiperiod blend scheduling problem.

7. Computational Results

Several multiperiod blend scheduling problems were solved using Heuristic 1 and Algorithms 1 and 2 described in Section 5.3, as well as commercial solvers BARON and GloMIQO for comparison. Details regarding these multiperiod blend scheduling problems are shown in Appendix B. In the results that follow, wall times (i.e. "The time that passes according to the clock on the wall") are reported as compared to CPU time, as the MILP solvers are able to take advantage of multiple threads. All computations were performed on a dual CPU computer with two Intel Xeon X5650 processors at 2.66GHz each, 16 GB of RAM, and running Ubuntu Linux 10.04. The models were implemented and solved using GAMS 23.8.1, BARON 10.2.0, GloMIQO 1.0.0, and Gurobi 4.6.1 (Gurobi, 2011).

7.1. Heuristic 1

Heuristic 1 solves (**MPBP′**), fixes the process binary variables y, and solves the resulting NLP to yield an approximate solution to the globally optimal solution. However, as there is no way to determine the optimality gap in this approach, a discretization level must be chosen carefully. For the multiperiod blend scheduling problem, C is discretized and ranges between 0 and 1 in the sample problems (concentrations outside of this range, or in a subset of this range, can be scaled). A sufficiently fine discretization level is chosen for all problems: $p = -3$ (i.e. 10^{-3}). Results for solving the test suite of multiperiod blend scheduling problems using Heuristic 1 at this discretization level are shown in Table 8. For all but one problem, Heuristic 1 can find an approximate solution in less than 1 hour, and the remaining case required less than 2 hours. Note that compared to the problem sizes reported in Table 5, Heuristic 1 requires many more variables and constraints due to the discretized reformulation.

Table 8: Computational results for Heuristic 1

Tanks	6	8	8	8	8	8	8
Time Periods	3	3	3	3	4	4	4
Total Wall Time (s)	9.96	839.0	245.0	250.2	337.3	4800.6	978.5

MILP Solve (Gurobi)							
Wall Time (s)	9.94	838.8	244.9	250.1	336.9	4800.1	978.1
Objective Function	13.3594	45.2554	7.3818	13.5268	53.9496	9.2051	19.9806
Constraints	1886	7089	6811	7092	9557	9581	9083
Continuous Variables	1319	4632	4466	4632	6713	6717	6449
Binary Variables	420	855	855	855	1144	1148	1128
NLP Post-Solve (BARON)							
Wall Time (s)	0.02	0.16	0.03	0.19	0.32	0.5	0.37
Objective Function	13.3594	45.2804	7.3936	13.5268	53.9627	9.2266	20.039
MILP-NLP Difference	0.00%	0.06%	0.16%	0.00%	0.02%	0.23%	0.29%
Constraints	215	459	657	579	467	447	633
Continuous Variables	131	327	465	419	323	327	457

The MILP-NLP difference is also reported in the table, which shows the relative difference between the solutions of (**MPBP′**) and the NLP post-solve using BARON as the NLP solver. In some cases, the NLP solution is the same as the MILP solution (0% difference), but in other cases there is a very small gap. This gap can be shown to decrease with finer discretization, but can be large for coarse discretization levels.

Using Heuristic 1, the multiperiod blend scheduling problems can be solved in significantly less time than using MINLP solvers such as BARON or GloMIQO. However, the major shortfall of this method is that it lacks any way of determining the optimality gap, and so it cannot be properly compared to commercial global solvers. This method does give a fairly accurate approximate solution to the problem that could be used as a lower bound (in a maximization problem) or as a starting point in an algorithm.

7.2. Algorithm 1

As described in Section 5.3, Algorithm 1 solves both (**MPBP′**) and (**MPBPR**) in succession, increasing the discretization, and repeating until the gap is closed. As both Problems (**MPBP′**) and (**MPBPR**) are MILPs, Gurobi can be used to solve both while taking advantage of 12 threads for maximum parallelization. The results for this algorithm are shown in Table 9. Note that a limit of 4 iterations is imposed in order to avoid numerical difficulties at high precision.

Table 9: Computational results for Algorithm 1

Tanks		6	8	8	8	8	8	8
Time Periods		3	3	3	3	4	4	4
Total Wall Time (s)		2.17	4573.5	2588.7	426.5	338.7	12309.0	1525.9
Relative Gap		0	$5.19 \cdot 10^{-4}$	$1.60 \cdot 10^{-3}$	0	$6.86 \cdot 10^{-4}$	$2.34 \cdot 10^{-3}$	$3.39 \cdot 10^{-3}$
Iterations		2	4	4	4	3	4	4
Lower Bound	$\boldsymbol{p = 0}$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
	$\boldsymbol{p = -1}$	13.36	44.69	7.13	13.08	53.60	8.33	$-\infty$
	$\boldsymbol{p = -2}$		45.19	7.38	13.49	53.93	9.15	19.68
	$\boldsymbol{p = -3}$		45.26	7.38	13.53		9.21	19.97
Upper Bound	$\boldsymbol{p = 0}$	15.38	47.49	19.63	14.33	54.46	9.71	20.94
	$\boldsymbol{p = -1}$	13.36	45.90	9.49	13.53	53.96	9.45	20.53
	$\boldsymbol{p = -2}$		45.37	7.39	13.53	53.96	9.24	20.04
	$\boldsymbol{p = -3}$		45.28	7.39	13.53		9.23	20.04
Wall Time (s) Lower Bound	$\boldsymbol{p = 0}$	0.02	0.03	0.04	0.03	0.45	0.05	0.03
	$\boldsymbol{p = -1}$	0.78	190.38	31.24	11.59	40.39	457.56	0.12
	$\boldsymbol{p = -2}$		491.16	126.9	55.52	118.38	2947.53	436.28
	$\boldsymbol{p = -3}$		2199.86	413.99	115.01		7201.38	764.07
Wall Time (s) Upper Bound	$\boldsymbol{p = 0}$	0.08	0.3	0.54	0.21	1.16	4.13	0.68
	$\boldsymbol{p = -1}$	1.2	40.42	133.99	13.67	27.43	49.15	8.23
	$\boldsymbol{p = -2}$		422.71	534.47	57.65	150.53	659.04	85.74
	$\boldsymbol{p = -3}$		1408.33	1347.15	172.53		989.73	230.3

Algorithm 1 is clearly an improvement over BARON, as can be seen by comparing the results in Tables 5 and 9. Note that the times reported in the lower portion of the table are for each individual subproblem (they are not cumulative). While the MINLP formulation required more than two hours to solve in all except the smallest case, Algorithm 1 can solve the same problems in less than two hours in all but one case. For most problems, less than an hour (and in some cases, significantly less) is required to solve the example multiperiod blend scheduling problems to global optimality.

7.3. Algorithm 2

Algorithm 2 is similar to Algorithm 1, but instead of using problem (**MPBP′**), the binary variables from the original problem (**MPBP**) are fixed to the values returned by the solution of (**MPBPR**). The lower bound is then obtained using a local NLP solver. Results for this algorithm are shown in Table 10.

Table 10: Computational results for Algorithm 2

Tanks		6	8	8	8	8	8	8
Time Periods		3	3	3	3	4	4	4
Total Wall Time (s)		3.97	2839.9	120.5	15.5	41.3	2831.1	44.2
Relative Gap		0	$1.03 \cdot 10^{-4}$	0	0	$7.99 \cdot 10^{-7}$	0	0
Iterations		2	4	3	2	2	4	3
Lower Bound	$p = 0$	7.49	32.18	11.26	11.04	39.68	8.92	16.89
	$p = -1$	13.36	30.29	5.08	13.53	53.96	8.67	19.5
	$p = -2$		45.3	7.39			9.2	20.03
	$p = -3$		45.3				9.23	
Upper Bound	$p = 0$	15.38	47.49	19.63	14.33	54.46	9.71	20.94
	$p = -1$	13.36	45.9	9.49	13.53	53.96	9.45	20.53
	$p = -2$		45.37	7.39			9.24	20.03
	$p = -3$		45.29				9.23	
Wall Time (s) Lower Bound	$p = 0$	2.41	0.9	0.62	3.13	17.49	2.14	2.53
	$p = -1$	13.36	30.29	5.08	13.53	53.96	8.67	19.5
	$p = -2$		1.36	0.09			1.3	0.51
	$p = -3$		0.53				1.02	
Wall Time (s) Upper Bound	$p = 0$	0.26	0.54	0.42	0.26	0.95	3.23	0.86
	$p = -1$	1.23	94.37	24.43	11.86	21.34	43.91	7.79
	$p = -2$		867.33	94.75			783.62	31.91
	$p = -3$		1872.71				1993.72	

Using Algorithm 2, the multiperiod blend scheduling problems are solved even more quickly than using Algorithm 1. In most cases, less than a minute is required to solve the problems to global optimality. Even the problems that require more time to converge still finish in less than one hour. This is, in some cases, computational reductions of more than two orders of magnitude are achieved.

The comparative performance of Heuristic 1 and Algorithms 1 and 2 shown in Table 11 is worth noting. Algorithm 1 exhibits the worst performance, which is to be expected as it solves two MILPs at each iteration, (**MPBP′**) and (**MPBPR**), while both Heuristic 1 and Algorithm 2 solve only 1 MILP at each iteration. However, the more surprising result is that Algorithm 2 generally performs faster than Heuristic 1. Even when solving multiple MILPs (one for each iteration), Algorithm 2 generally performs better than Heuristic 1, which only solves a single MILP. This is largely explained by Heuristic 1 solving a single large MILP (representing a very fine discretization) while Algorithm 2 solves several smaller MILPs (starting at a coarse discretization and increasing precision with each iteration).

Table 11: Comparison of performance between Heuristic 1 and Algorithms 1 and 2

	Tanks	6	8	8	8	8	8	8
	Time Periods	3	3	3	3	4	4	4
Total	**Heuristic 1**	9.96	839.0	245.0	250.2	337.3	4800.6	978.5
Wall	**Algorithm 1**	2.17	4573.5	2588.7	426.5	338.7	12309.0	1525.9
Time (s)	**Algorithm 2**	3.97	2839.9	120.5	15.5	41.3	2831.1	44.2

7.4. Algorithm 2 – Binary Formulation

The seven problems in the previous section were solved using the binary formulation of Section 6.2 with Algorithm 2. As seen in Table 12, solving the problems with Algorithm 2 using the binary (base 2) formulation, performance is generally better than the decimal formulation, but not always. Five out of the seven problems were solved slightly faster using the binary formulation, but the decimal formulation outperformed the binary formulation in the other two cases by a significant margin.

While the binary formulation yields slightly faster solve times, likely due to the smaller problem sizes (see Table 13), it is only a marginal improvement. However, one approach that could be used to increase the performance of the binary formulation is to skip discretization levels (e.g. $p = p + 2$ instead of $p = p + 1$ after each iteration).

Table 12: Computational results for Algorithm 2 using the binary formulation

Tanks		6	8	8	8	8	8	8
Time Periods		3	3	3	3	4	4	4
Total Wall Time (s)		1.89	4492.71	95.78	10.23	26.9	4787.7	35.57
Relative Gap		0.00%	-0.03%	0.00%	0.00%	0.00%	0.00%	-0.02%
Iterations		5	11	8	5	5	11	8
Lower Bound	*p = 0*	-∞	-∞	-∞	-∞	-∞	-∞	-∞
	p = 1	-∞	-∞	-∞	-∞	-∞	-∞	-∞
	p = 2	-∞	-∞	-∞	-∞	-∞	-∞	-∞
	p = 3	-∞	-∞	-∞	-∞	-∞	-∞	-∞
	p = 4	13.3594	-∞	5.0826	13.5268	53.9627	-∞	-∞
	p = 5		-∞	5.0826			-∞	-∞
	p = 6		-∞	5.0826			-∞	-∞
	p = 7		45.29657	7.3936			9.201263	20.039
	p = 8		45.29657				9.201263	
	p = 9		45.29657				9.201263	
	p = 10		45.29657				9.2266	
Upper Bound	*p = 0*	15.3796	47.49117	19.6306	14.3266	54.4587	9.705844	20.9379
	p = 1	15.3796	47.49117	19.6306	14.3266	54.4587	9.705844	20.9379
	p = 2	15.3796	47.49117	19.6306	14.3266	54.4587	9.705844	20.9379
	p = 3	15.3796	47.49117	19.6306	14.3266	54.4587	9.705844	20.9379
	p = 4	13.3594	45.89925	9.4916	13.5268	53.9627	9.445313	20.5272
	p = 5		45.89925	9.4916			9.445313	20.5272
	p = 6		45.89925	9.4916			9.445313	20.5272
	p = 7		45.36722	7.3936			9.239259	20.03524
	p = 8		45.36722				9.239259	
	p = 9		45.36722				9.239259	
	p = 10		45.28203				9.2266	
Lower Bound Wall Time (s)	*p = 0*	0.14	0.12	0.04	0.03	0.03	0.02	0.03
	p = 1	0.02	0.04	0.03	0.04	0.06	0.03	0.04
	p = 2	0.02	0.05	0.03	0.04	0.05	0.03	0.04
	p = 3	0.02	0.04	0.03	0.04	0.04	0.03	0.04
	p = 4	0.02	0.02	0.02	0.02	0.02	0.04	0.02
	p = 5		0.02	0.02			0.02	0.02
	p = 6		0.02	0.02			0.02	0.02
	p = 7		0.02	0.02			0.02	0.03
	p = 8		0.02				0.02	
	p = 9		0.02				0.02	
	p = 10		0.02				0.03	
Upper Bound Wall Time (s)	*p = 0*	0.28	0.66	0.55	0.23	0.77	3.14	0.95
	p = 1	0.14	0.73	0.43	0.49	1.02	2.91	0.95
	p = 2	0.13	0.76	0.46	0.47	1.1	2.97	0.69
	p = 3	0.15	0.76	0.59	0.42	0.97	3.04	0.85
	p = 4	0.97	39.89	20.11	8.45	22.84	86.34	4.49
	p = 5		29.57	20.33			132.33	4.14
	p = 6		56.29	20.43			96.05	4.54
	p = 7		984.48	32.67			471.16	18.72
	p = 8		1505.68				426.01	
	p = 9		451.14				677.55	
	p = 10		1422.36				2885.92	

Table 13: Problem size statistics for base 2 and base 10 formulations of Algorithm 2 when solving an 8 tank, 3 time period problem (8T-3P-2Q-721).

Numerical Base		2	10
Total Wall Time (s)		10.23	15.5
	Iteration		
Equations	1	2404	3556
	2	2836	5140
	3	3268	
	4	3700	
	5	4132	
Variables	1	987	2230
	2	1243	2742
	3	1499	
	4	1755	
	5	2011	
Binary Variables	1	135	327
	2	183	567
	3	231	
	4	279	
	5	327	

7.5. Fixed Precision

If a satisfactory level of precision is known, a single iteration of Algorithm 2 can be used to significantly decrease computational time. Table 14 shows the problem sizes and computational performance of this approach, comparing the original MINLP solved by BARON and GloMIQO with a single iteration of Algorithm 2 at the precision level $p = -3$. Note that even at 7200s, the gap remaining for the global optimization solvers is large in most cases, and significantly larger than the optimality gaps of Algorithm 2. It is also interesting to note that while both global optimization solvers take significantly longer to converge than Algorithm 2, especially compared to a single iteration of Algorithm 2, GloMIQO tends to have smaller gaps at 7200s than BARON.

7.6. Additional Remarks

While the algorithms and approaches presented are a significant improvement over solving the original MINLP with global optimization solvers, several approaches and modifications can be used to potentially improve performance.

- Discretizing other variables, such as F and I in the multiperiod blend scheduling problem, can yield smaller problems depending on the number of qualities in the problem, resulting in potentially better performance in these cases. For the problems presented, though, we have found that discretizing the concentrations, C, tends to yield the best performance.
- Using numerical bases other than 10 and 2 is possible, but performance is likely to be comparable to the results already shown.
- The discretized variables can have arbitrary upper and lower bounds, and the discretization scheme can be translated or scaled accordingly to accommodate these bounds.
- Redundant constraints and McCormick envelopes can be added to tighten the relaxation of the bilinear terms (Kolodziej et al, 2012).
- Regarding the discretization scheme and relaxation, the slack variables introduced in the relaxed problems can alternatively be bounded by McCormick envelopes or other convex envelopes in order to decrease problem size.

In addition, several variations in the algorithmic approaches can be introduced, especially variations of the relaxation. These remain open for investigation in the future.

Table 14: Computational results for fixed-precision radix-based discretization

	Original MINLP			BARON		GloMIQO		Algorithm 2 (*p* = -3)				
Problem	**Binary Variables**	**Total Variables**	**Equations**	**Wall Time (s)**	**Gap**	**Wall Time (s)**	**Gap**	**Binary Variables**	**Total Variables**	**Equations**	**Wall Time (s)**	**Gap**
6T-3P-2Q-029	36	103	214	21.12	0.10%	1771	0.10%	420	1819	1990	1.25	0.00%
8T-3P-2Q-146	87	223	624	*7200*	18.0%	*7200*	1.96%	855	5743	7441	870.88	0.16%
8T-3P-2Q-718	87	223	607	*7200*	62.8%	*7200*	76.2%	855	5569	7151	97.69	0.00%
8T-3P-2Q-721	87	223	628	*7200*	8.82%	*7200*	1.04%	855	5743	7444	11.78	0.00%
8T-4P-2Q-480	124	313	885	*7200*	134%	*7200*	0.39%	1148	8233	10093	741.28	0.41%
8T-4P-2Q-531	104	273	737	*7200*	12.1%	*	*	1128	7933	9577	31.84	0.00%
8T-4P-2Q-852	120	305	861	*7200*	11.3%	*7200*	0.26%	1144	8225	10069	22.10	0.00%

*Solver Failure

8. Conclusion

This paper has addressed the multiperiod blend scheduling problem, an important extension to the pooling problem in order to account for time varying supply and demand and operational constraints, which is important in scheduling applications in process networks. The multiperiod blend scheduling problem has been formulated as a nonconvex MINLP which is difficult to solve as was initially shown with a small analytical example and a small numerical example. The alternative approaches developed in this paper, based on radix discretization, perform significantly better than traditional MINLP techniques, with decreased computational time of more than two orders of magnitude observed in some cases. While Heuristic 1 is an intuitive approach using the discretization approach, and Algorithm 1 is rigorous and intuitive with both an upper and lower bounding problem, they do not perform as well as Algorithm 2 in solving the multiperiod blend scheduling problem. The approach used in Algorithm 2 exhibits superior performance compared to both Heuristic 1 and Algorithm 1 and to other global optimization solvers, such as BARON and GloMIQO. This novel discretization scheme and the algorithms derived from it allow for the efficient solution of bilinear programming problems by reformulating them to MILPs and further reducing their problem size compared to other discretization schemes.

The multiperiod blend scheduling problem, however, is a complex problem to solve since the proposed global optimization techniques were only able to solve problems with 8 tanks, 4 time periods, and 2 product qualities within a reasonable time period. Realistic problems with tens or hundreds of time periods are still untractable by the proposed methods. In order to address larger multiperiod blend scheduling problems, decomposition methods could be investigated to create smaller and more easily solved subproblems.

Acknowledgements

The authors gratefully acknowledge financial support from the National Science Foundation under grant OCI-0750826, and from ExxonMobil Corporate Research.

References

Adhya, N., Tawarmalani, M., and Sahinidis, N.V. (1999). A Lagrangian Approach to the Pooling Problem. Industrial & Engineering Chemistry Research. 38 (5), 1956-1972.

Ahmetović, E. and Grossmann, I. E. (2010). Global superstructure optimization for the design of integrated process water networks. AIChE Journal. 57 (2), 434-457.

Al-Khayyal, F. A., and Falk J. E. (1983). Jointly Constrained Biconvex Programming. Mathematics of Operations Research, 8 (2), 273-286.

Alfaki, M. and Haugland, D. (2012). Strong formulations for the pooling problem. Journal of Global Optimization, 1–20. 10.1007/s10898-012-9875-6.

Audet, C., Brimberg, J., Hansen, P., Digabel, S. L. and Mladenovic, N. (2004). Pooling problem: Alternate formulations and solution methods. Management Science 50(6), 761–776.

Bagajewicz, M. (2000). A review of recent design procedures for water networks in refineries and process plants. Computers & Chemical Engineering 24(910), 2093 – 2113.

Bagajewicz, M., Rodera, H. and Savelski, M. (2002). Energy efficient water utilization systems in process plants. Computers & Chemical Engineering, 26(1), 59 – 79.

Balas, E. (1985). Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. SIAM Journal of Algebraic and Discrete Mathematics 6, 466-486.

Balasubramanian, J., Sawaya, N., Furman, K.C., Kocis, G.R., Margolies, M.F., Miranda, M.J., McDonald, M.K., Song, J.-H, and Warrick, P.H. Tools for Assisting in Petroleum Product Transportation Logistics. U.S. Patent Application Pub. No.: US2010/0332273 A1, Pub. Date: Dec. 30, 2010.

Ben-Tal, A., Eiger, G. and Gershovitz, V. (1994). Global minimization by reducing the duality gap. Mathematical Programming, 63, 193–212.

Brook, A., Kendrick, D. and Meeraus, A. (1988). GAMS, a user's guide, SIGNUM Newsletter 23, 10–11. URL: http://doi.acm.org/10.1145/58859.58863

Bussieck, M. R. and Drud, A. (2001). SBB: A new solver for mixed integer nonlinear programming. URL: http://www.gams.com/presentations/present_sbb.pdf

Floudas, C. A. and Visweswaran, V. (1995). Quadratic Optimization. In Horst, R. and Pardalos, P. M. (eds.) Handbook of Global Optimization. Kluwer.

Furman, K.C., Jia, Z., and Ierapetritou, M.G. (2007). A robust event-based continuous time formulation for tank transfer scheduling. Industrial & Engineering Chemistry Research, 46, 9126.

Galan, B. and Grossmann, I. E. (1998). Optimal design of distributed wastewater treatment networks. Industrial & Engineering Chemistry Research, 37(10), 4036–4048.

Gounaris, C. E., Misener, R., and Floudas, C. A. (2009). Computational Comparison of Piecewise−Linear Relaxations for Pooling Problems. Industrial & Engineering

Chemistry Research 48 (12), 5742-5766.

Grossmann, I. E., Viswanathan, J., Vecchietti, A., Raman, R. and Kalvelagen, E. (2003). GAMS/DICOPT: A discrete continuous optimization package.

Grossmann, I.E. and Ruiz, J.P. (2011). Generalized Disjunctive Programming: A Framework for Formulation and Alternative Algorithms for MINLP Optimization. IMA Volume 154, Mixed Integer Nonlinear Programming, (eds., Jon Lee and Sven Leyffer).

Guignard, M., and Kim, S. (1987). Lagrangean decomposition: A model yielding stronger Lagrangean bounds. Mathematical Programming 39:215–228. DOI: 10.1007/BF02592954.

Gurobi Optimizer Reference Manual Version 4.5. Gurobi Optimization. http://www.gurobi.com/doc/45/refman/ (2011)

Haverly, C. A. (1978). Studies of the behavior of recursion for the pooling problem. SIGMAP Bull. 25, 19-28.

Horst, R., and Tuy, H. (1996). Global Optimization: Deterministic Approaches. Springer.

Huang, C.-H., Chang, C.-T., Ling, H.-C. and Chang, C.-C. (1999). A mathematical programming model for water usage and treatment network design. Industrial & Engineering Chemistry Research 38(7), 2666–2679.

Ierapetritou, M.G. and Jia, Z. (2008). Gasoline Blending and Distribution Scheduling: An MILP model. Encyclopedia of Optimization Second Edition, Floudas, C.A. and Pardalos, P.M. (eds), Springer.

Jeżowski, J. (2010). Review of water network design methods with literature annotations. Industrial & Engineering Chemistry Research 49(10), 4475–4516. URL: http://pubs.acs.org/doi/abs/10.1021/ie901632w

Jia, Z. and Ierapetritou, M.G. (2003a). Mixed Integer Linear Programming Model for Gasoline Blending and Distribution Scheduling. Industrial & Engineering Chemistry Research, 42, 825.

Jia, Z., Ierapetritou, M.G., and Kelly, J.D. (2003b). Refinery Short-term Scheduling Using Continuous Time Formulation Crude Oil Operations. Industrial & Engineering Chemistry Research, 42, 3085.

Jia, Z. and Ierapetritou, M.G. (2004). Efficient short-term scheduling of refinery operations based on a continuous time formulation. Computers and Chemical Engineering, 28, 1001.

Karuppiah, R., Grossmann, I.E. (2006). Global optimization for the synthesis of integrated water systems in chemical processes. Computers and Chemical Engineering 30, 650-673.

Karuppiah, R., Furman, K.C., and Grossmann, I.E. (2008). Global optimization for scheduling refinery crude oil operations. Computers and Chemical Engineering, 32, 2745-2766.

Kelly, J.D. and Mann, J.L. (2003a). Crude oil blend scheduling optimization: An application with multi-million dollar benefits - Part 1. Hydrocarbon Processing, 82(6), 47-51.

Kelly, J.D. and Mann, J.L. (2003b). Crude oil blend scheduling optimization: An application with multi-million dollar benefits - Part 2. Hydrocarbon Processing, 82(7), 72-79.

Kolodziej, S. P., Castro, P.M., and Grossmann, I. E. (2012). Global Optimization of Bilinear Programs with a Multiparametric Disaggregation Technique. Submitted for Publication.

Lee, H., Pinto, J.M., Grossmann, I.E., and Park, S. (1996). Mixed-integer linear programming model for refinery short-term scheduling of crude-oil unloading with inventory management. Industrial Engineering and Chemical Research, 35, 1630.

Liang, B.,Yongheng, J., and Dexian, H. A Novel Two-Level Optimization Framework Based on Constrained Ordinal Optimization and Evolutionary Algorithms for Scheduling of Multipipeline Crude Oil Blending. Industrial & Engineering Chemistry Research, *in press*.

Li, J., Misener, R., and Floudas, C.A. (2012). Continuous-time modeling and global optimization approach for scheduling of crude oil operations. AIChE Journal, 58(1), 205–226.

Li, J., Li, W.K., Karimi, I.A., and Srinivasan, R. (2007). Improving the robustness and efficiency of crude scheduling algorithms. AIChE Journal, 53(10), 2659-2680.

Liberti, L., Pantelides, C.C. (2006). An Exact Reformulation Algorithm for Large Nonconvex NLPs Involving Bilinear Terms. Journal of Global Optimization, 36, 161.

Liberti, L., Cafieri, S., Tarissan, F. (2009). Reformulations in Mathematical Programming: A Computational Approach. In Abraham, A., Hassanien, A., Siarry, P., and Engelbrecht, A. (eds.) Foundations of Computational Intelligence Volume 3, pp. 153-234. Springer Berlin, Heidelberg.

Lim, S.-R., Park, D. and Park, J. M. (2007). Synthesis of an economically friendly water network system by maximizing net present value. Industrial & Engineering Chemistry Research, 46(21), 6936–6943.

LINDO API 7.0 User Manual. (2011). URL: http://www.lindo.com/downloads/PDF/APIMAN7.pdf

Maravelias, C. T., and Papalamprou, K. (2009). Polyhedral Results for Discrete-time MIP Formulations for Scheduling and Production Planning. Computers and Chemical Engineering, 33(11), 1890-1904.

McCormick, G.P. (1976). Computability of global solutions to factorable nonconvex programs. Part I. Convex underestimating problems. Mathematical Programming 10, 146.

Meyer, C. A., and Floudas, C. A. (2006). Global optimization of a combinatorially complex generalized pooling problem. AIChE Journal 52(3), 1027–1037.

Misener, R. and Floudas, C. A. (2009). Advances for the pooling problem: Modeling, global optimization,and computational studies. Applied and Computational Mathematics 8(1), 3 – 22.

Misener, R., and Floudas, C. A. (2010). Global optimization of large-scale generalized pooling problems: Quadratically constrained MINLP models. Industrial & Engineering Chemistry Research 49(11), 5424–5438.

Misener, R., Thompson, J. P. and Floudas, C. A. (2011). APOGEE: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes. Computers & Chemical Engineering 35(5), 876 – 892. Selected

Papers from ESCAPE-20 (European Symposium of Computer Aided Process Engineering - 20), 6-9 June 2010, Ischia, Italy. URL: http://www.sciencedirect.com/science/article/pii/S0098135411000366

Misener, R. and Floudas, C. (2012). GloMIQO: Global mixed-integer quadratic optimizer, Journal of Global Optimization pp. 1–48. 10.1007/s10898-012-9874-7. URL: http://dx.doi.org/10.1007/s10898-012-9874-7

Moro, L.F.L., Zanin, A.C., and Pinto, J.M. (1998). A planning model for refinery diesel production. Computers and Chemical Engineering, 22, S1039.

Moro, L.F.L. and Pinto, J.M. (2004). Mixed-integer programming approach for short-term crude oil scheduling. Industrial and Engineering Chemistry Research, 43, 85-94.

Mouret, S., Grossmann, I.E., and Pestiaux, P. (2009). A novel priority-slot based continuous-time formulation for crude-oil scheduling problems. Industrial and Engineering Chemistry Research, 48, 8515-8528.

Oral, M., and Kettani, O. (1992). A Linearization Procedure for Quadratic and Cubic Mixed-Integer Problems. Operations Research. 40, Supplement 1: Optimization, S109-S116.

Pham, V., Laird, C., and El-Halwagi, M. (2009). Convex Hull Discretization Approach to the Global Optimization of Pooling Problems. Industrial & Engineering Chemistry Research. 48 (4), 1973-1979.

Pinto, J.M., Joly, M., and Moro, L.F.L. (2000). Planning and scheduling models for refinery operations. Computers and Chemical Engineering, 24, 2259.

Quesada, I. and Grossmann, I.E. (1995). Global optimization of bilinear process networks with multicomponent flows. Computers & Chemical Engineering, 19, Issue 12, 1219-1242.

Ruiz, J.P., Grossmann, I.E. (2011). Exploiting vector space properties to strengthen the relaxation of bilinear programs arising in the global optimization of process networks. Optimization Letters, 5, 1.

Reddy, P.C.P., Karimi, I.A., and Srinivasan, R. (2004a). A new continuous-time formulation for scheduling crude oil operations. Chemical Engineering Science, 59(6), 1325-1341.

Reddy, P.C.P, Karimi, I.A., and Srinivasan, R. (2004b). A novel solution approach for optimizing scheduling of crude oil operations. AIChE Journal, 50(6), 1177-1197.

Ruiz, J. P. and Grossmann, I. E. (2010). Strengthening of lower bounds in the global optimization of bilinear and concave generalized disjunctive programs. Computers & Chemical Engineering, 34(6), 914 – 930.

Ruiz, J.P. and Grossmann, I.E. (2011). Exploiting vector space properties to strengthen the relaxation of bilinear programs arising in the global optimization of process networks. Optimization Letters 5, 1–11.

Saharidis, G. and Ierapetritou, M.G. (2009). Scheduling of Loading and Unloading of Crude oil in a refinery with combine mixture preparation. Industrial & Engineering Chemistry Research, 48, 2624.

Sahinidis, N. V. (1996). BARON: A general purpose global optimization software package. Journal of Global Optimization 8, 201–205. 10.1007/BF00138693. URL: http://dx.doi.org/10.1007/BF00138693

Sherali, H. D. and Adams, W. P. (1998). A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems, Springer.

Sherali, H. D., Adams, W. P. and Driscoll, P. J. (1998). Exploiting special structures in constructing a hierarchy of relaxations for 0-1 mixed integer problems, Oper. Res. 46(3), 396–405.

Sherali, H.D., Alameddine, A. (1992). A new reformulation linearization technique for bilinear programming problems. Journal of Global Optimization 2, 379-410.

Shah, N. (1996). Mathematical programming techinques for crude-oil scheduling. Computers and Chemical Engineering, 20, S1227.

Shah, N. and Ierapetritou, M.G. (2011). Short-term scheduling of a large-scale refinery operations: incorporating logistics. AIChE Journal, 57(6) , 1570.

Shor, N. (1990). Dual quadratic estimates in polynomial and Boolean programming. Annals of Operations Research. 25 (1), 163-168.

Smith, E. M. B., and Pantelides, C. C. (1997). Global optimisation of nonconvex MINLPs. Computers & Chemical Engineering. 21, S791-S796.

Tawarmalani, M., and Sahinidis, N. V. (2002). Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming. Kluwer Academic Publishers. 254–284.

Teles, J.P., Castro, P.M., Matos, H.A. (2011). Multiparametric disaggregation technique for global optimization of polynomial programming problems. Journal of Global Optimization. DOI: 10.1007/s10898-011-9809-8.

Teles, J.P., Castro, P.M., Matos, H.A. (2012). Global Optimization of Water Networks Design using Multiparametric Disaggregation. Computers and Chemical Engineering 40, 132-147.

Tjoa, I.B., Ota, Y., Matsuo, H., and Natori, Y. (1997). Ethylene Plant Scheduling System Based on a MINLP Formulation. Computers and Chemical Engineering, 21, Supplement 1, 1077(5), 1073.

Vielma, J. P., Ahmed, S., Nemhauser, G. (2010). Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. Operations Research, 58, 303–315.

Vielma, J. P., Nemhauser, G. (2010). Modeling disjunctive constraints with a logarithmic number of binary variables and constraints, Mathematical Programming.

Wang, Y.-P. and Smith, R. (1994a). Design of distributed effluent treatment systems. Chemical Engineering Science, 49(18), 3127 – 3145.

Wang, Y.-P. and Smith, R. (1994b). Wastewater minimization. Chemical Engineering Science, 49(7), 981 – 1006.

Wicaksono, D. S. and Karimi, I. A. (2008). Piecewise MILP under- and overestimators for global optimization of bilinear programs. AIChE Journal. 54 (4), 991-1008.

Xu, H.K. (2003). An Iterative Approach to Quadratic Optimization. Journal of Optimization Theory and Applications. 116 (3), 659- 678.

Ye, Y. (1999). Approximating quadratic programming with bound and quadratic constraints. Mathematical Programming. 84 (2), 219-226.

Yu, N. (1998). Semidefinite relaxation and nonconvex quadratic optimization. Optimization Methods and Software. 9 (1-3), 141-160.

Zamora, J. M. and Grossmann, I. E. (1998). Continuous global optimization of structured process systems models. Computers & Chemical Engineering, 22(12), 1749 – 1770.

Zhou, R.-J., Li, L.-J., Xiao,W. and Dong, H.-G. (2009). Simultaneous optimization of batch process schedules and water-allocation network. Computers & Chemical Engineering, 33(6), 1153 – 1168.

Appendix A – Derivation of Radix-Based Discretization

We present in this appendix a summarized derivation of the radix-based discretization method given in Kolodziej et al (2012). Given a nonconvex bilinear term $w_{ij} = x_i{\cdot}x_j$, the multiparametric disaggregation technique described by Teles et al. (2012) can be used to discretize a bilinear programming problem. This reformulation can be derived in terms of generalized disjunctive programming (GDP) (Grossmann & Ruiz, 2011) and exact linearization (Oral & Kettani, 1992). For simplicity in the notation, we first rewrite the bilinear product $w_{ij} = x_i{\cdot}x_j$ as a single bilinear term $w = u{\cdot}v$. This product can be represented exactly with the following constraints and disjunction:

$$w = u \cdot v \tag{A1}$$

$$v = \sum_{\ell \in \mathbb{Z}} \lambda_\ell \tag{A2}$$

$$\bigvee_{k=0}^{9} \left[\lambda_\ell = 10^\ell \cdot k\right] \quad \forall\ \ell \in \mathbb{Z} \tag{A3}$$

where v is discretized through the disjunction in (A3) that selects one digit k for each power ℓ

$\in \mathbb{Z}$.

Considering the convex hull reformulation of the disjunction in (A3) (Balas, 1985) by introducing the disaggregated variables $\hat{\lambda}_{k\ell}$ and the binary variables $z_{k\ell}$, leads to the following equations where $K = \{k \mid k = 0,1,\ldots 9\}$:

$$\lambda_\ell = \sum_{k=0}^{9} \hat{\lambda}_{k\ell} \quad \forall \ \ell \in \mathbb{Z} \tag{A4}$$

$$\hat{\lambda}_{k\ell} = 10^\ell \cdot k \cdot z_{k\ell} \quad \forall \ \ell \in \mathbb{Z}; k \in K \tag{A5}$$

$$\sum_{k=0}^{9} z_{k\ell} = 1 \quad \forall \ \ell \in \mathbb{Z} \tag{A6}$$

$$z_{k\ell} \in \{0,1\} \quad \forall \ \ell \in \mathbb{Z}; k \in K$$

Substituting equation (A5) into equation (A4) and substituting equation (A7) into equation (A2) leads to the fully discretized (but still exact representation of) v:

$$v = \sum_{\ell \in \mathbb{Z}} \sum_{k=0}^{9} 10^\ell \cdot k \cdot z_{k\ell} \tag{A8}$$

$$\sum_{k=0}^{9} z_{k\ell} = 1 \quad \forall \ \ell \in \mathbb{Z} \tag{A6}$$

$$z_{k\ell} \in \{0,1\} \quad \forall \ \ell \in \mathbb{Z}; k \in K$$

Considering the product $w = u \cdot v$ by substituting equation (A8) into equation (A1) leads to

$$w = u \cdot \left[\sum_{\ell \in \mathbb{Z}} \sum_{k=0}^{9} 10^\ell \cdot k \cdot z_{k\ell} \right] = \sum_{\ell \in \mathbb{Z}} \sum_{k=0}^{9} 10^\ell \cdot k \cdot u \cdot z_{k\ell}$$

which involves the nonlinear term $u \cdot z_{k\ell}$. Performing an exact linearization (Oral & Kettani, 1992), we introduce a new continuous variable, $\hat{u}_{k\ell} = u \cdot z_{k\ell}$, yielding

$$w = \sum_{\ell \in \mathbb{Z}} \sum_{k=0}^{9} 10^\ell \cdot k \cdot \hat{u}_{k\ell} \tag{A9}$$

and the following lower and upper bound constraints:

$$\hat{u}_{k\ell} \leq u^U \cdot z_{k\ell} \quad \forall \ \ell \in \mathbb{Z}; k \in K \tag{A10}$$

$$\hat{u}_{k\ell} \geq u^L \cdot z_{k\ell} \quad \forall \ \ell \in \mathbb{Z}; k \in K \tag{A11}$$

where u^U and u^L are the non-negative upper and lower bounds on u. Furthermore, to relate u to $\hat{u}_{k\ell}$, we derive one additional constraint from equation (A6):

$$\sum_{k=0}^{9} z_{k\ell} = 1 \quad \forall \ \ell \in \mathbb{Z} \tag{A6}$$

$$\sum_{k=0}^{9} u \cdot z_{k\ell} = u \quad \forall \ \ell \in \mathbb{Z}$$

$$\sum_{k=0}^{9} \hat{u}_{k\ell} = u \quad \forall \ \ell \in \mathbb{Z} \tag{A12}$$

In this way, we arrive at the final set of mixed-integer linear constraints for representing the bilinear product $w = u \cdot v$. Since it is infeasible to compute the infinite sums over all integers, we represent v to a finite level of precision using a maximum power of 10 (P) and a minimum power of 10 (p). This yields the constraints proposed by Teles, Castro, & Matos (2012):

$$\begin{aligned}
&w = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{u}_{k\ell} \\
&v = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot z_{k\ell} \\
&\hat{u}_{k\ell} \leq u^U \cdot z_{k\ell} \quad \forall \ \ell \in L; k \in K \\
&\hat{u}_{k\ell} \geq u^L \cdot z_{k\ell} \quad \forall \ \ell \in \mathbb{Z}; k \in K \\
&\sum_{k=0}^{9} \hat{u}_{k\ell} = u \quad \forall \ \ell \in L \\
&\sum_{k=0}^{9} z_{k\ell} = 1 \quad \forall \ \ell \in L \\
&z_{k\ell} \in \{0,1\} \quad \forall \ \ell \in L; k \in K
\end{aligned} \tag{A13}$$

where $L = \{\ell = p, p+1, \ldots P\}$.

Because of this finite level of precision, these constraints are no longer an exact representation of the product $w = u \cdot v$. When we incorporate the constraints (A13) into problem (**P**) by redefining $w_{ij} = x_i \cdot x_j$, and selecting x_j as the variable on which discretization is performed, the resulting problem (**P′**) shown below will represent a mixed-integer approximation to the original problem:

$$\text{Max } z' = f_0(x,y) = \sum_{(i,j)\in BL_0} a_{ij0} w_{ij} + h_0(x,y)$$

Subject to

$$f_q(x,y) = \sum_{(i,j)\in BL_q} a_{ijq} w_{ij} + h_q(x,y) \le 0 \quad q \in Q\backslash\{0\}$$

$$w_{ij} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{x}_{ijk\ell} \quad \forall\ (i,j) \in BL_q; q \in Q$$

$$x_j = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot z_{jk\ell} \quad \forall\ j \in \{j|(i,j) \in BL_q, q \in Q\}$$

$$\hat{x}_{ijk\ell} \le x_i^U \cdot z_{jk\ell} \quad \forall\ (i,j) \in BL_q; q \in Q; k \in K; \ell \in L \qquad \textbf{(P′)}$$

$$\hat{x}_{ijk\ell} \ge x_i^L \cdot z_{jk\ell} \quad \forall\ (i,j) \in BL_q; q \in Q; k \in K; \ell \in L$$

$$\sum_{k=0}^{9} \hat{x}_{ijk\ell} = x_i \quad \forall\ (i,j) \in BL_q; q \in Q; k \in K; \ell \in L$$

$$\sum_{k=0}^{9} z_{jk\ell} = 1 \quad \forall\ j \in \{j|(i,j) \in BL_q, q \in Q\}; k \in K; \ell \in L$$

$$z_{jk\ell} \in \{0,1\} \quad \forall\ j \in \{j|(i,j) \in BL_q, q \in Q\}; k \in K; \ell \in L$$

$$x \in S \cap \Omega \subset \mathbb{R}^n$$

$$y \in \{0,1\}$$

where x_j and w_{ij} represents the discrete and continuous approximations to x_j and w_{ij}, respectively, in the constraints (A9)-(A12) and (A6).

To derive the relaxed problem (**PR**) using radix-based discretization, we again switch to the notation $w = u \cdot v$ for the bilinear term and begin with the discretization of v from the truncated set of constraints (A13) by adding the slack term Δv (bounded between 0 and 10^p) to represent a continuous v, denoted as v^R:

$$v^R = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot z_{k\ell} + \Delta v$$
$$\sum_{k=0}^{9} z_{k\ell} = 1 \quad \forall \ \ell \in L \tag{A14}$$
$$z_{k\ell} \in \{0,1\} \quad \forall \ \ell \in L; k \in K$$
$$0 \leq \Delta v \leq 10^p$$

We can rewrite Δv using a form similar to the discretization scheme already described. Given the following relationship,

$$0 \leq \sum_{k=0}^{1} 10^p \cdot k \cdot \tilde{z}_k \leq 10^p$$
$$0 \leq \tilde{z}_k \leq 1$$

we use this expression to represent Δv:

$$\Delta v = \sum_{k=0}^{1} 10^p \cdot k \cdot \tilde{z}_k$$
$$0 \leq \tilde{z}_k \leq 1$$

The constraints (A14) can now be rewritten as

$$v^R = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot z_{k\ell} + \sum_{k=0}^{1} 10^p \cdot k \cdot \tilde{z}_k$$
$$\sum_{k=0}^{9} z_{k\ell} = 1 \quad \forall \ \ell \in L \tag{A15}$$
$$z_{k\ell} \in \{0,1\} \quad \forall \ \ell \in L; k \in K$$
$$0 \leq \tilde{z}_k \leq 1 \quad \forall \ k \in \{0,1\}$$

Following a similar derivation as for problem (**P**) we obtain the new optimization problem, (**PR**):

$$\text{Max } z^R = f_0 = \sum_{(i,j) \in BL_0} a_{ij0} w_{ij} + h_0(x) \tag{\textbf{PR}}$$

Subject to

$$f_q(x) = \sum_{(i,j)\in BL_q} a_{ijq} w_{ij} + h_q(x) \leq 0 \quad q \in Q\backslash\{0\}$$

$$w_{ij} = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot \hat{x}_{ijk\ell} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{x}_{ijk} \quad \forall\ (i,j) \in BL_q; q \in Q$$

$$x_j = \sum_{\ell=p}^{P} \sum_{k=0}^{9} 10^{\ell} \cdot k \cdot z_{ijk\ell} + \sum_{k=0}^{1} 10^{p} \cdot k \cdot \tilde{z}_{ijk} \quad \forall j \in \{j|(i,j) \in BL_q, q \in Q\}$$

$$\hat{x}_{ijk\ell} \leq x_i^U \cdot z_{ijk\ell} \quad \forall\ (i,j) \in BL_q; q \in Q; \ell \in L; k \in K$$

$$\hat{x}_{ijk\ell} \geq x_i^L \cdot z_{ijk\ell} \quad \forall\ (i,j) \in BL_q; q \in Q; \ell \in L; k \in K$$

$$\tilde{x}_{ijk} \leq x_i^U \cdot \tilde{z}_{ijk} \quad \forall\ (i,j) \in BL_q; q \in Q; k \in \{0,1\}$$

$$\tilde{x}_{ijk} \geq x_i^L \cdot \tilde{z}_{ijk} \quad \forall\ (i,j) \in BL_q; q \in Q; k \in \{0,1\}$$

$$\sum_{k=0}^{9} \hat{x}_{ijk\ell} = x_i \quad \forall\ (i,j) \in BL_Q; q \in Q; \ell \in L$$

$$\sum_{k=0}^{1} \tilde{x}_{ijk} = x_i \quad \forall\ (i,j) \in BL_Q; q \in Q$$

$$\sum_{k=0}^{9} z_{ijk\ell} = 1 \quad \forall\ (i,j) \in BL_q; q \in Q; \ell \in L$$

$$\sum_{k=0}^{1} \tilde{z}_{ijk} = 1 \quad \forall (i,j) \in BL_q; q \in Q$$

$$z_{ijk\ell} \in \{0,1\} \quad \forall\ (i,j) \in BL_q; q \in Q; \ell \in L; k \in K$$

$$0 \leq \tilde{z}_{ijk} \leq 1 \quad \forall\ (i,j) \in BL_q; q \in Q; k \in \{0,1\}$$

$$x \in S \cap \Omega \subset \mathbb{R}^n$$

While (**PR**) does not exactly represent the product $w_{ij} = x_i \cdot x_j$ and is feasible for values of w_{ij}, x_i, and x_j that do not satisfy $w_{ij} = x_i \cdot x_j$, the bilinear term is feasible in (**PR**). Thus, (**PR**) is a relaxation of (**P**).

Appendix B – Sample Multiperiod blend scheduling Problems

The suite of 7 test problems were randomly generated. Each is denoted by the number of tanks, the number of time periods, and the number of product qualities, as well as a random three-digit code to distinguish different problems of the same size. For example, the first sample problem, 6T-3P-2Q-029, is a 6 tank, 3 time period, 2 quality problem.

B.1. Problem 6T-3P-2Q-029

Table 15: Supply tank specifications

	C_s (% mass)		F_s (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	β_s **($/kg)**
1	0.6	0.4	0.0	1.0	0.0	0.9
2	0.2	0.4	0.6	0.1	0.9	0.9

Table 16: Demand tank specifications

	Bounds on C (% mass)		F_d (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	β_d **($/kg)**
5	0.1-0.9	0.4-0.4	0.25	0.81	0.88	8.6
6	1.0-1.0	0.6-0.8	0.01	0.14	0.10	7.1

Table 17: Initial conditions and costs of flows

		C_0 (% mass)		Cost Coefficients α (10^3 $), β ($/ kg) to Tank			
Tank	I_0	**Qual. 1**	**Qual. 2**	**3**	**4**	**5**	**6**
1	1.0	--	--	0.84, 0.84	0.05, 0.55	0.94, 0.32	0.81, 0.60
2	0.5	--	--	0.79, 0.80	0.05, 0.28	0.65, 0.49	0.97, 0.75
3	0.2	0.2	0.5	--	0.57, 0.30	0.26, 0.89	0.45, 0.82
4	0.1	0.7	0.6	--	--	0.10, 0.86	--
5	1.8	--	0.8	--	--	--	--
6	0.5	--	0.2	--	--	--	--

Globally optimal objective function value: 13.3594

B.2. Problem 8T-3P-2Q-146

Table 18: Supply tank specifications

	C_s (% mass)		F_s (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	β_s **($/kg)**
1	0.7	0.2	0.3	0.1	0.7	0.2
2	0.9	0.0	0.0	0.8	0.3	1.0

Table 19: Demand tank specifications

	Bounds on C (% mass)		F_d (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	β_d **($/kg)**
7	0.7-1.0	0.5-0.9	0.95	0.44	0.77	9.6
8	0.8-1.0	0.1-0.7	0.03	0.38	0.80	9.6

Table 20: Initial conditions and costs of flows

		C_0 (% mass)		Cost Coefficients α (10^3 $), β ($/ kg) to Tank					
Tank	I_0	**Qual. 1**	**Qual. 2**	**3**	**4**	**5**	**6**	**7**	**8**
1	1.6	--	--	--	0.20, 0.25	0.62, 0.47	0.35, 0.83	0.59, 0.55	0.92, 0.29
2	1.8	--	--	0.76, 0.75	0.38, 0.57	0.08, 0.05	--	--	0.53, 0.78
3	0.3	0.7	0.0	--	0.93, 0.13	0.57, 0.47	0.01, 0.34	0.16, 0.79	0.31, 0.53
4	1.8	0.8	0.9	0.17, 0.60	--	0.26, 0.65	0.69, 0.75	0.45, 0.08	0.23, 0.91
5	1.3	0.7	0.8	0.15, 0.83	0.54, 1.00	--	0.08, 0.44	0.11, 0.96	0.00, 0.77
6	0.2	0.7	0.4	0.82, 0.87	0.08, 0.40	0.26, 0.80	--	0.43, 0.91	0.18, 0.26
7	0.6	--	--	--	--	--	--	--	--
8	1.1	--	--	--	--	--	--	--	--

Globally optimal objective function value: 45.2966

B.3. Problem 8T-3P-2Q-718

Table 21: Supply tank specifications

	C_s (% mass)		F_s (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	β_s **($/kg)**
1	0.7	0.1	1.0	0.8	0.4	0.2
2	0.7	0.5	0.6	0.5	0.8	0.2

Table 22: Demand tank specifications

	Bounds on C (% mass)		F_d (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	β_d **($/kg)**
7	0.9-1.0	0.0-0.5	0.08	0.17	0.83	8.1
8	0.5-0.9	0.2-0.6	0.13	0.39	0.80	5.8

Table 23: Initial conditions and costs of flows

		C_0 (% mass)		Cost Coefficients α (10^3 $), β ($/ kg) to Tank					
Tank	I_0	**Qual. 1**	**Qual. 2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0.1	--	--	--	0.18, 0.13	1.00, 0.17	0.03, 0.56	0.88, 0.67	0.19, 0.37
2	1.5	--	--	0.46, 0.98	0.16, 0.86	--	0.64, 0.38	0.19, 0.43	0.48, 0.12
3	1.0	0.1	0.7	--	0.59, 0.23	0.58, 0.58	0.25, 0.29	0.62, 0.27	0.82, 0.98
4	1.0	0.0	0.1	0.73, 0.34	--	--	0.58, 0.11	0.91, 0.88	0.82, 0.26
5	1.8	0.5	0.1	0.59, 0.02	0.43, 0.31	--	0.16, 0.18	0.42, 0.09	0.60, 0.47
6	1.2	0.8	0.8	0.70, 0.70	0.64, 0.03	0.07, 0.32	--	0.53, 0.65	0.41, 0.82
7	1.2	--	--	--	--	--	--	--	--
8	1.7	--	--	--	--	--	--	--	--

Globally optimal objective function value: 7.3936

B.4. Problem 8T-3P-2Q-721

Table 24: Supply tank specifications

	C_s (% mass)		F_s (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	β_s **(\$/kg)**
1	0.4	0.1	1.0	0.1	0.4	0.1
2	0.1	0.9	0.6	0.2	0.8	0.2

Table 25: Demand tank specifications

	Bounds on C (% mass)		F_d (10^3 kg) in time			
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	β_d **(\$/kg)**
7	0.1-0.4	0.2-1.0	0.02	0.17	0.73	5.1
8	0.2-0.9	0.4-0.7	0.04	0.65	0.65	4.0

Table 26: Initial conditions and costs of flows

		C_0 (% mass)		Cost Coefficients α (10^3 \$), β (\$/ kg) to Tank					
Tank	I_0	**Qual. 1**	**Qual. 2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0.3	--	--	0.30, 0.47	0.23, 0.84	0.19, 0.23	0.17, 0.23	--	0.44, 0.31
2	1.7	--	--	--	0.92, 0.43	0.18, 0.90	0.98, 0.44	--	0.11, 0.26
3	1.2	0.5	0.3	--	0.41, 0.59	0.26, 0.60	0.71, 0.22	0.12, 0.30	0.32, 0.42
4	1.1	0.9	0.4	0.51, 0.09	--	0.26, 0.80	0.03, 0.93	0.73, 0.49	0.58, 0.24
5	0.3	0.1	0.8	0.46, 0.96	0.55, 0.52	--	0.23, 0.49	0.62, 0.68	0.40, 0.37
6	1.7	0.4	0.2	0.99, 0.04	0.89, 0.91	0.80, 0.10	--	0.26, 0.34	0.68, 0.14
7	1.2	--	--	--	--	--	--	--	--
8	0.7	--	--	--	--	--	--	--	--

Globally optimal objective function value: 13.3527

B.5. Problem 8T-4P-2Q-480

Table 27: Supply tank specifications

	C_s (% mass)		F_s (10^3 kg) in time				
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	**4**	β_s **(\$/kg)**
1	0.1	0.2	0.0	0.2	0.7	0.5	0.0
2	0.9	0.8	0.0	0.6	0.6	0.5	0.9

Table 28: Demand tank specifications

	Bounds on C (% mass)		F_d (10^3 kg) in time				
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	**4**	β_d **(\$/kg)**
7	0.7-1.0	0.3-0.6	0.30	0.19	0.18	0.63	2.4
8	0.5-0.8	0.3-0.8	0.34	0.69	0.37	0.78	4.2

Table 29: Initial conditions and costs of flows

		C_0 (% mass)		Cost Coefficients α (10^3 \$), β (\$/ kg) to Tank					
Tank	I_0	**Qual. 1**	**Qual. 2**	**3**	**4**	**5**	**6**	**7**	**8**
1	1.2	--	--	--	0.92, 0.43	0.18, 0.90	0.98, 0.44	0.11, 0.26	0.41, 0.59
2	0.7	--	--	0.26, 0.60	0.71, 0.22	0.12, 0.30	0.32, 0.42	0.51, 0.09	0.26, 0.80
3	1.0	0.4	0.2	--	0.03, 0.93	0.73, 0.49	0.58, 0.24	0.46, 0.96	0.55, 0.52
4	0.8	0.4	0.1	0.23, 0.49	--	0.62, 0.68	0.40, 0.37	0.99, 0.04	0.89, 0.91
5	0.2	0.1	0.9	0.80, 0.10	0.26, 0.34	--	0.68, 0.14	0.72, 0.11	0.65, 0.49
6	0.5	1.0	0.6	0.78, 0.72	0.90, 0.89	0.33, 0.70	--	0.20, 0.03	0.74, 0.50
7	0.2	--	--	--	--	--	--	--	--
8	0.4	--	--	--	--	--	--	--	--

Globally optimal objective function value: 9.2266

B.6. Problem 8T-4P-2Q-531

Table 30: Supply tank specifications

	C_s (% mass)		F_s (10^3 kg) in time				
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	**4**	β_s **($/kg)**
1	1.0	0.6	0.4	0.1	0.2	0.8	0.5
2	0.6	0.9	0.8	0.1	0.4	0.8	0.2

Table 31: Demand tank specifications

	Bounds on C (% mass)		F_d (10^3 kg) in time				
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	**4**	β_d **($/kg)**
7	0.6-1.0	0.7-1.0	0.06	0.53	0.66	0.29	8.9
8	0.5-0.9	0.2-0.6	0.40	0.42	0.63	0.43	0.3

Table 32: Initial conditions and costs of flows

		C_0 (% mass)		Cost Coefficients α (10^3 $), β ($/ kg) to Tank					
Tank	I_0	**Qual. 1**	**Qual. 2**	**3**	**4**	**5**	**6**	**7**	**8**
1	1.8	--	--	--	0.19, 0.37	--	--	0.46, 0.98	--
2	1.2	--	--	0.16, 0.16	0.64, 0.38	0.19, 0.43	0.48, 0.12	--	0.59, 0.23
3	1.2	0.5	0.1	--	0.38, 0.58	0.25, 0.29	0.62, 0.27	0.82, 0.98	0.73, 0.34
4	1.7	0.8	0.8	0.11, 0.58	--	0.91, 0.88	0.82, 0.26	0.59, 0.02	0.43, 0.31
5	1.6	0.7	0.1	0.18, 0.16	0.42, 0.09	--	0.60, 0.47	0.70, 0.70	0.64, 0.03
6	1.2	0.7	0.5	0.32, 0.07	--	0.53, 0.65	--	0.41, 0.82	0.72, 0.97
7	0.4	--	--	--	--	--	--	--	--
8	0.5	--	--	--	--	--	--	--	--

Globally optimal objective function value: 20.02668

B.7. Problem 8T-4P-2Q-852

Table 33: Supply tank specifications

	C_s (% mass)		F_s (10^3 kg) in time				
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	**4**	β_s **($/kg)**
1	0.7	0.2	0.3	0.1	0.7	1.0	0.2
2	0.9	0.8	0.0	0.8	0.3	0.0	1.0

Table 34: Demand tank specifications

	Bounds on C (% mass)		F_d (10^3 kg) in time				
Tank	**Qual. 1**	**Qual. 2**	**1**	**2**	**3**	**4**	β_d **($/kg)**
7	0.6-1.0	0.5-0.9	0.44	0.77	0.19	0.45	9.6
8	0.8-1.0	0.1-1.0	0.38	0.80	0.49	0.65	9.6

Table 35: Initial conditions and costs of flows

		C_0 (% mass)		Cost Coefficients α (10^3 $), β ($/ kg) to Tank					
Tank	I_0	**Qual. 1**	**Qual. 2**	**3**	**4**	**5**	**6**	**7**	**8**
1	1.6	--	--	0.35, 0.83	0.59, 0.55	0.92, 0.29	0.76, 0.75	0.38, 0.57	--
2	1.8	--	--	--	0.08, 0.05	0.53, 0.78	0.93, 0.13	0.57, 0.47	0.01, 0.34
3	0.3	0.7	0.0	--	0.16, 0.79	0.31, 0.53	0.17, 0.60	0.26, 0.65	0.69, 0.75
4	1.8	0.8	0.9	0.45, 0.08	--	0.23, 0.91	0.15, 0.83	0.54, 1.00	0.08, 0.44
5	1.3	0.7	0.8	0.11, 0.96	0.00, 0.77	--	0.82, 0.87	0.08, 0.40	0.26, 0.80
6	0.2	0.7	0.4	0.43, 0.91	0.18, 0.26	0.15, 0.14	--	0.87, 0.58	0.55, 0.14
7	0.6	--	--	--	--	--	--	--	--
8	1.1	--	--	--	--	--	--	--	--

Globally optimal objective function value: 53.9627