# Optimal Scheduling of Refined Products Pipelines with Multiple Sources

**Diego C. Cafaro and Jaime Cerdá***

*INTEC (UNL—CONICET), Güemes 3450, 3000 Santa Fe, Argentina*

Most contributions on short-term planning of multiproduct pipeline operations deal with pipelines featuring a single input terminal. In common-carrier pipelines, however, several refineries located at different sites use the same trunk line for shipping refined petroleum products to downstream output terminals. They can be regarded as multiple-source pipelines with input facilities at nonorigin points. The operation of intermediate sources raises some new difficult issues. Pumping runs taking place at intermediate locations can either insert a new lot or increase the size of a batch in transit. Batches are no longer arranged in the line in the same order that they are injected, and tracking the batch sequence becomes a more complex task. This paper introduces a novel continuous formulation for the scheduling of multiple-source pipelines operating on fungible or segregated mode. A case study involving a single pipeline that transports three distillates from two input to three output terminals was successfully solved over a 10-day time horizon.



## 1. Introduction

Shipments of crude oil and refined petroleum products through pipelines represent a great portion of the inland transportation. The main purpose of refined products pipelines is to supply large amounts of petroleum derivatives to marketing terminals at the right time with the least cost of transportation, and the minimum interfacial losses due to product mixing. After the refiners contact the pipeline carrier to place their transport orders or "nominations" for the next month, the scheduler usually develops a plan of the pipeline activities over a monthly horizon. Planning the operation of a multiproduct pipeline is an important industrial problem that was first formalized in the pioneer work of Hane and Ratliff.[1]

The pipeline schedule indicates the sequencing, timing, and location of pumping and stripping operations, that is, the input and delivery schedules. The input schedule establishes the sequence of batch injections, the entering products, the batch lengths, the injection rates, and the input terminals at which batches are inserted. Finding the optimal batch ordering and sizing along trunk lines is a combinatorial problem aimed at minimizing the transmix generated between consecutive product batches. To this end, batches are made as large as possible to reduce the number of interfaces. This is achieved by merging shipments of the same standard refined product from different shippers into a joint, longer batch, that is, a fungible operation mode. On the other hand, the delivery schedule indicates the products leaving the pipeline, the stripping operations, and the amounts diverted to the assigned destinations during every pumping run. Input and delivery schedules constitute the so-called batch schedule. Refined products pipelines can operate in two different ways: segregated or fungible modes. In segregated mode (also called batch mode), the identity of the product shipped is maintained throughout the transportation process, and the same material that was accepted for shipment in the origin is delivered at the destination. But in fungible operations, the carrier does not necessarily deliver the same batch of product injected at the specified input terminal. Instead, the discharged material will match the same product specifications but may not be the original lot.

In the final step, the pump optimization phase for the selected batch plan is performed using a detailed hydraulic model to schedule pump operations. This phase provides the times at which pumps should be turned on/off to run the batch plan at the specified flow rates. Its goal is to minimize the number of pipeline stoppages and pump switchings so as to get savings on the energy cost consumed for restarting flow in idle segments, and on pump maintenance costs. Pump optimization then requires a choice of the best sequence and timing of stripping operations during each batch injection, that is, a detailed delivery schedule. Flow is assumed to stop downstream of the delivery point and upstream of the input terminal. This paper is just focused on the optimal batch schedule for refined products pipelines with multiple sources. Pump optimization using discrete-event simulation will be studied in a next paper.

**Multiple-Source Pipelines.** In common-carrier pipelines, several oil refineries located at different sites use the same trunk line for shipping batches of distinct oil derivatives to downstream distribution terminals near large consumer markets (see Figure 1). Previous work on multiproduct pipeline scheduling assumed that the pipeline carries oil products from only one oil refinery (or a single input terminal) located at the pipeline origin, to several depots along the line, that is, the single-source multiple-destination case. Multiple-source pipelines involve additional input locations at nonorigin points collecting batches of oil products from several downstream refineries to move them along the line to farther output terminals. In other words, a multisource pipeline transports batches of oil products from various sources to many destinations.

As remarked by Hane and Ratliff,[1] multiple-source pipelines raise some difficult issues that can be ignored in the treatment of single-source pipeline systems. Let us assume that a multiple-source trunk line is operated on fungible mode. Therefore, individual batches of the same grade or the same product featuring common specifications, though provided by different shippers, can be joined into a single consolidated batch with several destinations. In this way, it may happen that a batch of product *A* coming from refinery *S1* and destined for depot *D1* can be finally delivered to terminal *D2* also requiring *A*. The operation of multiple-source systems implies the execution of a sequence of pumping runs each one injecting some amount of a certain product in the pipeline from the assigned input node. A major difference with regard to the single-source case is the need of additionally specifying the input terminal where the next pumping run will be driven.

* To whom correspondence should be addressed. Tel.: +54 342 4559175. Fax: +54 342 4550944. E-mail: jcerda@intec.unl.edu.ar.

10.1021/ie900015b CCC: $40.75 
Published on Web 06/08/2009

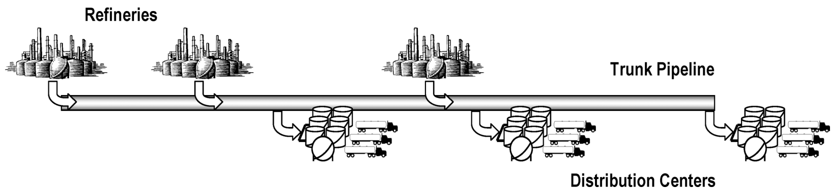


**Figure 1.** A multiple-source multiple-destination pipeline.

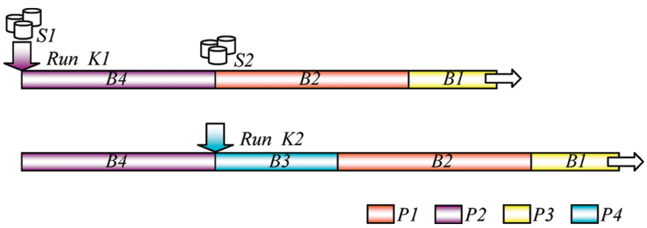


**Figure 2.** The batch sequence pattern in multisource pipelines.



Another important feature of multisource trunk pipelines is the possibility of injecting a new batch from a nonorigin point. Therefore, batches flowing along the line will no longer be chronologically arranged. In single-source pipelines, batches are sequenced on the line in the same order that they are injected. If the chronological batch sequence holds, batch *B3* will be immediately succeeded by batch *B4*, if *B4* is pumped right after batch *B3*. Then, there is a one-to-one relationship between batches and pumping runs, and a common mathematical entity can be used to represent both of them in the problem formulation. In multisource pipelines, a batch is not necessarily preceded by those previously pumped into the line (see Figure 2). Batch *B4* injected at the origin may be preceded by batch *B3* pumped in the line from a downstream input terminal, even though *B4* has been inserted earlier. In previous work, batch $(i + 1)$ not only travels through the line right behind batch $i$ but it has also been pumped right after the injection of batch $i$ at the pipeline origin. Things are different in multisource systems. A batch $(i + 1)$ flowing immediately after batch $i$ is not necessarily inserted in the line right after pumping batch $i$. Since batches and pumping runs are not always similarly sequenced, they should be handled as independent entities in the multiple-source pipeline scheduling problem formulation. The chronological batch sequence, a common assumption of single-source pipeline scheduling approaches, can no longer be used. This surely complicates the evaluation of the interface reprocessing cost because the tracking of the product batch sequence may be even more difficult.

In Figure 2, batch *B4* has been injected in the line from the input terminal *S1* during pumping run *K1* and travels right behind batch *B3*, even though *B3* is inserted in the next run *K2*. This happens because run *K2* takes place at a downstream source *S2*.

One of the most important issues in pipeline operation is the so-called mixing cost. Most mixing that occurs in the pipeline is due to the displacement of a lighter fluid by a heavier one, because the heavy fluid tends to settle to the bottom of the pipeline (Hane and Ratliff[1]). To reduce settling effects, it is important that batches move on turbulent flow and the pipeline almost never remains idle. In this regard, it must be remarked that the injection of a batch in the line from an intermediate input terminal $T_a$ implies that the segment of batches from the origin to location $T_a$ will stay idle, while the rest is moving along the line. The movement of such a first segment of batches or a part of it will be restarted at the expense of a penalty cost, when a new batch is pumped from an input terminal closer to the origin. This operational feature can only be found in multiple-source pipelines. Similarly, the segment of the pipeline beyond terminal $T_b$ remains idle during a pumping run if $T_b$ is the farthest depot to which some product flows are delivered from the line. This latter type of flow pattern can be found in multiple-destination pipelines.

An additional critical matter in the operation of multisource pipelines closely related to the mixing cost is the batch integrity. The pipeline operator is usually forbidden from inputting another product in the pipeline at some nonorigin point if, by so doing, a batch in transit becomes split into a pair of nonconsecutive smaller lots. New interfaces will be generated and consequently, mixing costs rise. In multiple-source pipelines running on fungible mode, pumping runs taking place at downstream sources can either insert a new batch of product or increase the size of a batch in transit. In the former case, the new batch should be injected just at the interface of two consecutive lots to mostly avoid the splitting of in-transit batches. In the latter case, the flowing batch to be enlarged should be accessed from the intermediate input terminal. In both situations, the primary goal is to keep the mixing costs as low as possible. The chance of increasing the size of a batch in transit through product injections at downstream input terminals is an exclusive feature of multiple-source pipelines running on fungible mode. In short, a pumping run taking place at an intermediate input node not always generates new interfaces as assumed in the single-source case. Sometimes, it may only increase the size of a batch inserted before from an upstream input terminal and therefore, no additional interfaces are created.

This paper presents a new mixed-integer linear programming (MILP) formulation for the planning and scheduling of oil products pipelines operating on either fungible or segregated mode and featuring multiple input and output terminals. The proposed approach uses a continuous volume-and-time domain representation.

**Literature Review.** Most publications on short-term planning of pipeline operations deal with single-source pipeline systems. Different types of representations and solution techniques for the pipeline batch scheduling problem have been proposed. Among them, knowledge-based heuristic techniques (Sasikumar et al.[2]), discrete-event simulation tools (Maruyama Mori et al.,[3] García-Sánchez et al.[4]) and rigorous optimization models. Nonrigorous search techniques often yield costly solutions if they start from a poor initial point and require a large CPU time to even find a feasible schedule. On the other hand, optimization models can be grouped into two classes: (a) discrete representations (Rejowski and Pinto,[5] Magatao et al.,[6] Zyngier and Kelly[7]), and (b) continuous formulations (Cafaro and Cerdá,[8,9] Relvas et al.[10]). In particular, Zyngier and Kelly[7] proposed a unified approach for the scheduling of pipelines and other transport and inventory systems operations. Formulations of type (a) generally use uniform time and volume discretization. However, a recent paper of Rejowski and Pinto[5] assumes that each pipeline segment is composed by packs with equal or different prespecified capacities to account for reductions in the pipeline diameter, and the horizon length comprises time intervals of adjustable duration to allow changes in the pump injection rate. In turn, continuous time-and-volume representations were introduced by Cafaro and Cerdá[8,9] and Relvas et al.[10] to get more efficient problem formulations that can consider longer time horizons with less computational cost.

For any pipeline problem, the last batch input can never be completely delivered to the assigned destinations because there is no way to push it forward from its source. When the planning horizon is a few days long, things are worse because the initial linefill is mostly enough to meet the terminal demands. Therefore, product inputs may have nothing to do with future requirements. According to Hane and Ratliff,[1] there are two ways to overcome this problem. The first one is to be only interested in inputting all the planned batches to the pipeline and ignore the issue of delivering all of them to output terminals. The second alternative is to extend the batch sequence by either (i) assuming the existence of an infinite amount of some filler product to push the last batch input out of the line (the preferred option in the cited literature), or (ii) considering future demands that are assumed to closely mimic the current order set. The filler product option often generates an inappropriate initial linefill for the next horizon. A third alternative was presented by Cafaro and Cerdá[9] by developing an efficient MILP continuous-time framework for the scheduling of single-source pipeline operation over a multiperiod rolling horizon. At the completion of the current period, the fixed-length planning horizon moves forward and the rescheduling process based on updated problem data is triggered again over the new horizon instance. In this way, a new period is recursively added to the end of the planning horizon and the related product requirements are further considered. The approach was successfully applied to a real-world pipeline scheduling problem over a four-week rolling horizon. Results show that the sequence of pumping runs finally executed by the pipeline dispatcher looks quite different from the one found by assuming the existence of some filler product. As the pumping runs become shorter, its number rises and the pipeline utilization level shows a substantial increase.

Only a few publications have either made some useful discussion or introduced a solution methodology for the scheduling of pipelines with multiple input terminals. A recent paper by Boschetto et al.[11] introduced an integrated heuristic framework for operational scheduling of multipipe fuel distribution systems that has been applied to a real-world case study. Hane and Ratliff[1] studied the problem of sequencing the input of petroleum products to a pipeline, from a single source at the origin to multiple destinations. Their approach assumed that the batch sequence follows a cyclic pattern, with cycle time and batch sizes adopted by the scheduler in a convenient manner before solving the problem. Batch sequencing is selected so that an objective function accounting for pumping and maintenance costs is minimized. The method used a discrete framework to handle the sequencing choices and applied a decomposition scheme to partition the problem into subproblems that can be easily priced out in a branch-and-bound algorithm. In the future work section, the authors raised some new issues that should be considered to schedule the operation of multiple-source pipelines.

On the other hand, Jittamai[12] modeled the distribution of multiple products through a single-source pipeline subject to delivery time-windows as a multicommodity network flow problem. Since the problem is NP-complete, the author also developed a heuristic reversed-flow solution algorithm to find the input schedule yielding the minimum total time-window violation. Similar to Hane and Ratliff,[1] the sequence of batch inputs follows a cyclic pattern with predefined cycle time. Besides, the reversed-flow algorithm was modified to account for pipeline systems with multiple input terminals. However, it was assumed that each product can at most be injected from just one input terminal, that is, a single source for each product. Moreover, every batch has a unique destination and the related batch size exactly meets the specified product demand at the assigned output terminal. As a result, the pipeline works on strict segregated mode and the enlargement of batches at intermediate sources cannot be considered. The selected problem goal was to minimize the number of pump starts in every cycle in order to reduce pipeline operating costs as much as possible. By using the reversed-flow algorithm, the optimal input sequence was found in approximately 40% of the tested problems.

## 2. Problem Elements

The multiple-source pipeline scheduling problem involves five major sets: pumping runs (the set $K$), batches (the set $I$), oil derivatives (the set $P$), oil refinery sources or input nodes (the set $S$) and output terminals (the set $J$). Therefore, the problem includes two additional sets with regards to the single-source case, $K$ and $S$. Since the number of pumping runs and batches to be injected is not precisely known before solving the problem, it should be arbitrarily adopted. The common rule is to choose their values as low as possible to decrease the problem size, but large enough to be at least equal to the ones required at the optimal pipeline schedule. The best choices usually depend on both the number of refining products to be transported and the extent of the scheduling horizon.

**2.1. Set of Pumping Runs *K*.** Let us define the set $K = \{k_1, k_2, k_3, ..., k_m\}$ with the pumping runs $k_1$, $k_2$,... chronologically ordered. Therefore, the $r$th-element of $K$, if performed, must be executed right after the run $k_{r-1}$. Moreover, their completion times should satisfy the condition: $C_r \geq C_{r-1}$, where $C_r$ is the end time of run $r$. When the number of pumping runs to be executed is lower than the one proposed ($||K||$) by some positive value $kf$, some $kf$ predefined runs are never performed. They are regarded as fictitious runs. Some model constraints will force the fictitious runs to be the last $kf$ elements of $K$. If the optimal solution features $kf = 0$, the cardinality of $K$ is increased by one and the modified problem formulation must be solved again. The procedure is repeated until no better optimum is discovered. A good initial choice for the number of pumping runs $||K||$ is given by

$$|K| = \left\lceil \sum_{p\in P} \left( \frac{2}{Q_{\min,p} + Q_{\max,p}} \sum_{j\in J} DL_{p,j} \right) \right\rceil$$

where ($Q_{\min,p}$, $Q_{\max,p}$) represent the minimum/maximum batch injection sizes for product $p$, and $DL_{p,j}$ stands for the delivery request of product $p$ at the output terminal $j$. Though product injections from multiple input terminals can be made, pumping runs are performed one at a time for operational reasons. Two properties characterize every active run: (a) the input node $s \in S$ where it takes place, and (b) either the new batch or the additional portion of a batch $i \in I$ injected in the line. If lot $i$ is enlarged by a run at the intermediate source $s$, then it should be well positioned to receive product from $s$. Every time a run is performed, some segments of the pipeline are activated and batches in those segments will move forward so as to divert some amounts of products to output terminals.

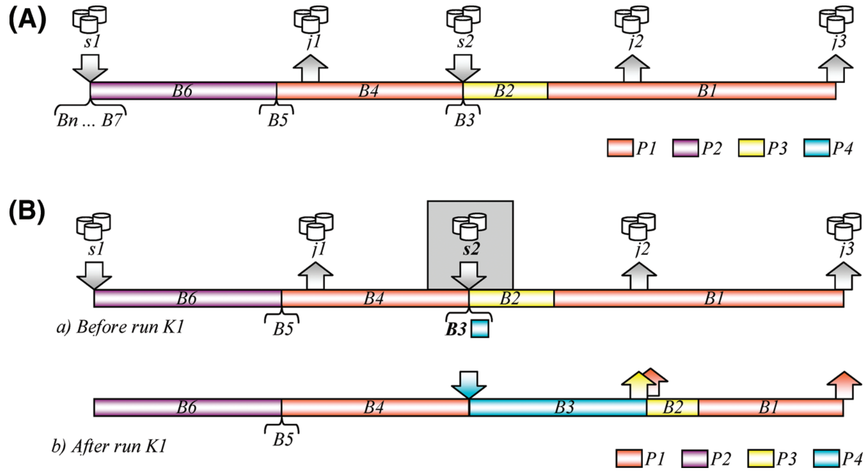


**Figure 3.** (a) Positioning batches *B3* and *B5* to be injected from a downstream location. (b) Product deliveries while injecting batch *B3* from source *s2*.



**2.2. Set of Batches *I*.** By definition, the set $I$ is given by $I = \{i_1, i_2, i_3, ..., i_n\}$ with the elements $i_1, i_2,...$ arranged in the same order that they are sequenced into the pipeline. Then, batch $i_1$ occupies the farthest position from the pipeline origin. Old batches ($i \in I^{\text{old}}$) already in the linefill at the start of the scheduling horizon are the first elements of the set $I$. The next ones stand for future batches ($i \in I^{\text{new}}$) to be injected through the planned pumping runs. In any case, the batch $i_b$ will be flowing immediately after the batch $i_{b-1}$ along the line. The separate handling of pumping runs and batches often leads to some reduction in the cardinality of set $I$. A good initial choice for $||I^{\text{new}}||$ is $||I^{\text{new}}|| = \alpha||K||$, with $\alpha = 0.9$ for segregated mode and $\alpha = 0.6$ for fungible mode. In optimal single-source pipeline schedules, it is sometimes observed that the pipeline remains idle after injecting batch $(b - 1)$ containing product $p$. The idle period usuallly arises because shipper's nominations do not make full use of the piepeline transport capacity. Sometimes, the new batch $b$ inserted after the idle period also transports product $p$. In the new problem formulation, batch $b$ will be regarded as an additional portion of batch $(b - 1)$. In this way, some saving in the number of batches is achieved. The major features of a batch are (a) the product $p$ that it contains, (b) the source $s$ from which it is inserted, and (c) the output terminals for which the batch is destined. If a new batch $b$ is shipped through a pumping run $k$ performed at some downstream location $s$, then it should be flowing immediately behind batch $(b - 1)$. To meet such a condition, the mathematical model must book batch $b$ for run $k$ at the intermediate source $s$. Therefore, batch $b$ will be traveling with a null size from the pipeline origin to source $s$ so that it can be accessed from that input terminal at the initial time of run $k$. In Figure 3a, the initial linefill includes two empty lots *B3* and *B5*, with *B3* just at the interface between batches *B2* and *B4*. They have been reserved for pumping runs to be performed at the intermediate input terminal *s2* during the current horizon. Batch *B3* just reaches the coordinate of the source *s2*.

Figure 3b shows the pipeline state at the end of run *K1* injecting batch *B3* from *s2*. No transport activity in the pipeline segment going from the origin to the input terminal *s2* is observed. In turn, the shipment of batch *B3* containing product *P4* pushes batches *B2* and *B1* forward to deliver product *P3* from *B2* to the output terminal *j2*, and product *P1* from *B1* to depots *j2* and *j3*.

**2.3. Sets of Input and Output Terminals.** Even though some pipeline stations may have a dual purpose working as both an input node and a receiving depot, the terminal sets $S$ and $J$ will just comprise "pure" input terminals and "pure" output terminals, respectively. Dual-purpose stations will be regarded as composed by a single source belonging to $S$ and a single output terminal in set $J$, both elements featuring the same location. The most important data related to the input nodes are (a) the set of products that can be injected, (b) the available product inventories that may change with time through additional refinery production runs, and (c) the terminal volumetric coordinate. Data related to output terminals are (a′) the set of products that are demanded over the scheduling horizon, (b′) the initial product inventories in terminal tanks, (c′) the product demands to be satisfied before the end of the current horizon, and (d′) the terminal volumetric coordinate.

**2.4. Set of Products.** The set $P$ comprises all the oil refinery products to be transported from input to output terminals closer to the consumer markets. In turn, $P_j$ stands for the group of products demanded by the output terminal $j \in J$, while $P_s$ denotes the subset of products that can be injected in the line from the input terminal $s$.

## 3. Problem Variables

Three different sets of binary variables are to be incorporated in the problem formulation to stand for the following pipeline scheduling decisions:

(a) The allocation of the oil refined product $p$ to batch $i$, $y_{i,p}$.

(b) The assignment of batch $i \in I$ and the input node $s \in S$ to the (nonfictitious) pumping run $k \in K$, $w_{i,s}^{(k)}$. For run $k$, it indicates the batch $i$ that is injected in the line and the source $s$ where it is accomplished. If $i$ is a new batch and $s$ is a downstream location, choosing $w_{i,s}^{(k)} = 1$ implies that an empty batch $i$ just at the interface between batches $(i - 1)$ and $(i + 1)$ has traveled from the origin to reach the location of terminal $s$ just at the time run $k$ begins. If instead $i$ is an existent batch flowing along the pipeline, $w_{i,s}^{(k)} = 1$ implies that a batch $i$ containing some product $p$ has already arrived at terminal $s$ to receive a further amount of product $p$ when run $k$ has started.

(c) The destinations $j \in J$ that receive some amount of product from the existent batch $i$ during run $k$, $x_{i,j}^{(k)}$. Choosing $x_{i,j}^{(k)} = 1$ implies that batch $i$ has reached the output terminal $j$ at the start of or during run $k$.

On the other hand, the model also includes the following sets of continuous variables:

(d) The problem time events $C_k$, that is, the end time of any pumping run $k$.

(e) The length of run $k$, $L_k$.

(f) The size of the flowing batch $i$ at time event $C_k$, $W_{i,k}$.

(g) The volume of the new (or the additional portion of the existing) batch $i$ injected in the line from source $s$ during run $k$, $Q_{i,s}{}^{(k)}$. In case batch $i$ was injected by an earlier pumping run $k' < k$, then it should reach the downstream location $s$ ($w_{i,s}{}^{(k)} = 1$) at time $[C_k - L_k]$, and the value of $Q_{i,s}{}^{(k)}$ would represent the additional volume of product supplied to batch $i$ from source $s$ during run $k$.

(h) The upper coordinate of the existing batch $i$ at time event $C_k$, that is, the pipeline volume from the origin to the farthest extreme section of batch $i$ at the end of run $k$, $F_{i,k}$.

(i) The amount of product diverted from batch $i$ to the output terminal $j$ during the pumping run $k$, $D_{i,j}{}^{(k)}$. Obviously, a nonzero $D_{i,j}{}^{(k)}$ implies that terminal $j$ is accessed from batch $i$ during run $k$.

Other continuous variables like $QP_{i,s,p}{}^{(k)}$ and $DP_{i,j,p}{}^{(k)}$ are incorporated in the problem formulation, but their values are provided by the related variables $Q_{i,s}{}^{(k)}$ and $D_{i,j}{}^{(k)}$ just in case product $p$ has been allocated to batch $i$ ($y_{i,p} = 1$).

## 4. Model Assumptions

(1) A single multisource pipeline with unidirectional flow is considered.

(2) The pipeline remains completely full of incompressible liquid products at any time. The only way to get a volume of product out of the line is by injecting an equal volume at some upstream input terminal.

(3) Consecutive product batches travel along the pipeline with no physical barrier between them, at turbulent flow to retard mixing.

(4) The "transmix" or contamination volume between a particular pair of refined products is a known constant, regardless of the pumping rate and the travel distance.

(5) The product injection rate should belong to the specified feasible range $[vb_{\min,s}; vb_{\max,s}]$ that can vary with the source $s$.

(6) Pumping runs are performed one at a time. Therefore, a single input facility can at most be injecting a batch of product at any time.

(7) Product demands at output terminals to be satisfied before the end of the scheduling horizon are deterministic data.

(8) Initial product inventories available in output terminals and the pipeline linefill at $t = 0$ are known.

(9) Product inventories available in refinery tanks to meet product demands at output terminals are also given.

Though it leads to use approximate interface volumes, assumption 4 still allows to fairly account for interface cost contributions in the objective function through using interface size estimations that roughly follow the same pattern of the exact values. In this way, pipeline schedules with a lower number of cheaper batch transitions can be generated. In the pump optimization phase, with many planning decisions already adopted, more exact interface costs can be computed. On the other hand, assumption 6 should be relaxed if several batch injections instead of a single one can be run simultaneously. This new problem feature will be considered in a next paper.

## 5. Mathematical Formulation

**5.1. Problem Constraints.** The proposed problem formulation comprises four blocks of equations. The pumping run constraints deal with the sizing, timing, and content of batches shipped through the line by the pumping runs. Batch-tracking restraints monitor changes in size and position of in-transit batches over time. Feasibility constraints ensure that (empty, nonempty) batches have arrived at downstream locations before starting product injections from intermediate refineries or deliveries of products to destinations. The final block of constraints guarantees that product inventories in depot tanks remain within feasible ranges and product demands at output terminals are all satisfied.

**5.1.1. Pumping Run Constraints. Pumping Run Sequencing.** A pumping run $k \in K$ must be started after the completion of the preceding run $(k - 1)$. Let $C_k$ denote the completion time of run $k$ and $L_k$ represent its duration. Then,

$$C_k - L_k \geq C_{k-1} \qquad \forall k \in K(k > 1) \tag{1}$$

For simplicity, transition times between consecutive pumping runs have been neglected. Nonetheless, they can be handled in a straightforward manner. Though arbitrarily chosen, the cardinality of the set $K$ should be at least as large as the number of performed pumping runs in the optimal pipeline schedule. If $h_{\max}$ stands for the overall length of the time horizon, then the completion time of any pumping run should not exceed $h_{\max}$.

$$C_k \leq h_{\max} \qquad \forall k \in K \tag{2}$$

**Allocation of Pumping Runs to Batches and Input Terminals.** By assumption 6, a pumping run $k$ can at most be performed from only one source $s \in S$ to either inject a new batch $i \in I^{\text{new}}$ in the line, or enlarge an existing batch $i \in I$ already in transit. Let us define the binary variable $w_{i,s}{}^{(k)}$ to denote that run $k$ is planned to inject a new batch $i \in I^{\text{new}}$ (or some amount of product to an existing batch $i \in I$) from input node $s$, whenever $w_{i,s}{}^{(k)} = 1$. Hence,

$$\sum_{s \in S} \sum_{i \in I} w_{i,s}^{(k)} \leq 1 \qquad \forall k \in K \tag{3}$$

In other words, there is at most a single source and only one batch associated to any pumping run. Moreover, a run $k^{\#}$ is never performed if all related variables $w_{i,s}{}^{(k\#)}$ are equal to zero.

**Sizing Batch Injections.** Let $Q_{i,s}{}^{(k)}$ denote the size of batch $i$ injected in the pipeline from the input facility $s$ through the pumping run $k$. Batch $i$ may be either a new batch or an additional portion of an existing batch $i$ flowing along the line. $Q_{i,s}{}^{(k)}$ will be positive only if run $k$ is really performed and a new (or an additional portion to an existent) batch $i$ is inserted ($w_{i,s}{}^{(k)} = 1$). Therefore,

$$Q_{\min} w_{i,s}^{(k)} \leq Q_{i,s}^{(k)} \leq Q_{\max} w_{i,s}^{(k)} \qquad \forall i \in I,\ s \in S,\ k \in K \tag{4}$$

where $(Q_{\min,}\ Q_{\max})$ stand for the minimum and maximum batch sizes that can be injected through a pumping run. Therefore, the volume of product inserted in the pipeline from source $s$ through run $k$ will be given by: $\sum_{i \in I} Q_{i,s}^{(k)}$.

**Pumping Run Length.** Let $L_{k,s}$ be the length of a pumping run $k$ taking place at the input terminal $s$. Then,

$$vb_{\min,s} L_{k,s} \leq \sum_{i \in I} Q_{i,s}^{(k)} \leq vb_{\max,s} L_{k,s} \qquad \forall k \in K,\ s \in S \tag{5}$$

where $L_k = \sum_{s \in S} L_{k,s}$ and the interval $[vb_{\min,s}; vb_{\max,s}]$ represents the feasible pumping rate range for the source $s$. As a result, a fictitious pumping run $k$ never executed must feature $L_k = 0$, and $\sum_{s \in S} \sum_{i \in I} Q_{i,s}^{(k)} = 0$.

**5.1.2. Batch Tracking Constraints. Tracking Batch Size over Time.** Let $W_{i,k}$ be the size of a batch $i$ at the end time

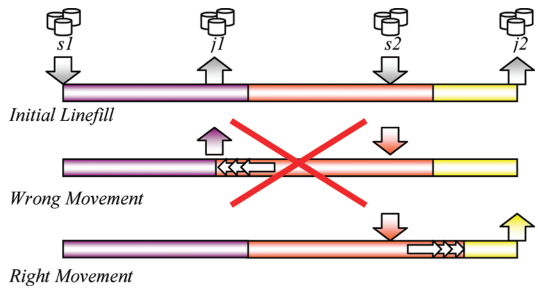


**Figure 4.** Batch forward movement condition.

of pumping run $k$, that is, at time $C_k$. During run $k$, the volume of batch $i$ can change for two reasons: (a) it may increase by receiving additional product from an intermediate location $s$, or (b) it may decrease by delivering some amount of product to depots $j \in J$.

$$W_{i,k} = W_{i,k-1} + \sum_{s \in S} Q_{i,s}^{(k)} - \sum_{j \in J} D_{i,j}^{(k)} \qquad \forall i \in I,\ k \in K \tag{6}$$

If batch $i$ is a new lot inserted in the line through run $k$, then $W_{i,k-1}$ will be zero. Otherwise, $W_{i,k-1}$ is the size of batch $i$ at the end of run $(k-1)$. If batch $i$ is already in the pipeline at time $t = 0$, $W_{i,k-1} = W_i^o$ for $k = 1$, with $W_i^o$ denoting the content of batch $i$ in the initial linefill.

**Tracking Batch Location over Time.** Let $F_{i,k}$ be the upper volumetric coordinate of batch $i$ at the end of run $k$, that is, at time $C_k$. In other words, $F_{i,k}$ is the total volume between the pipeline origin and the farthest extreme of batch $i$ after completing run $k$. In turn, $F_{i+1,k}$ represents the upper coordinate of batch $(i+1)$ immediately chasing batch $i$ in the pipeline at the end of run $k$. $F_{i+1,k}$ can also be regarded as the lower coordinate of batch $i$, that is, the pipeline volume between the origin and the closest-to-origin extreme of batch $i$ at time $C_k$. Then,

$$F_{i,k} - W_{i,k} = F_{i+1,k} \qquad \forall i \in I,\ k \in K \tag{7}$$

Since a batch can only move forward when the pipeline is active (see Figure 4), the upper coordinate of batch $i$ at the end of two consecutive runs $(k-1)$ and $k$ must satisfy the following condition,

$$F_{i,k-1} \le F_{i,k} \qquad \forall i \in I,\ k \in K \tag{8}$$

Similarly, no material can be delivered to depot $j$ while executing a pumping run at a downstream input terminal ($\sigma_j < \tau_s$). Then,

$$D_{i,j}^{(k)} \le \sum_{s/\tau_s < \sigma_j} \sum_{i' \in I} Q_{i',s}^{(k)} \qquad \forall i \in I,\ j \in J,\ k \in K \tag{9}$$

If $PV$ stands for the total pipeline content from the origin to the farthest depot, then the upper coordinate of any batch $i$ traveling along the pipeline must never be greater than $PV$.

$$F_{i,k} \le PV \qquad \forall i \in I,\ k \in K \tag{10}$$

Moreover, the lower coordinate of any batch $i$ in transit must be non-negative. In other words, it should stay at the downstream side of the origin or at most at the origin.

$$F_{i,k} - W_{i,k} \ge 0 \qquad \forall i \in I,\ k \in K \tag{11}$$

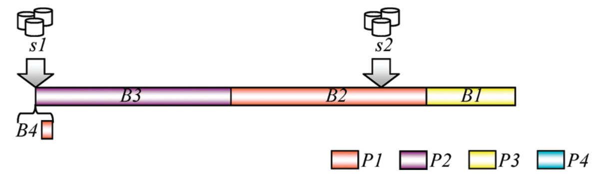


**Figure 5.** Feasible injection of oil products to existing batches.

**Pipeline Volumetric Balance.** At any time, a refined products pipeline remains full of products. Therefore, the total content of batches flowing inside the pipeline must be equal to $PV$ at the end time of any run $k$.

$$\sum_{i \in I} W_{i,k} = PV \qquad \forall k \in K \tag{12}$$

Because of the liquid incompressibility property, the total volume of products diverted from batches in pipeline transit to output terminal tanks must equal the size of the batch injected from any source, during any run $k$.

$$\sum_{i \in I} \sum_{s \in S} Q_{i,s}^{(k)} = \sum_{i \in I} \sum_{j \in J} D_{i,j}^{(k)} \qquad \forall k \in K \tag{13}$$

According to eqs 3 and 4, at most one of the variables $Q_{i,s}{}^{(k)}$ can take a nonzero value for any run $k$.

**5.1.3. Feasibility Constraints for Batch Injections and Product Deliveries.** When a pumping run $k$ takes place at some downstream location $s$, the proposed problem representation assumes that the injected material is supplied to an existing batch $i$. The batch size $W_{i,k-1}$ at the start of run $k$ will be either zero (if batch $i$ is a new one) or positive (if batch $i$ has been previously injected at an upstream terminal).

**Supplying Material from an Input Node to an Existing Batch.** An existing batch $i$ can receive material during run $k$ taking place at source $s$ only if the following two conditions are satisfied:

(a) Before starting run $k$, batch $i$ has already reached the location of the input facility $s$ ($\tau_s$). Then, the upper coordinate of batch $i$ ($F_{i,k-1}$) should never be lower than $\tau_s$.

$$F_{i,k-1} \ge \tau_s w_{i,s}^{(k)} \qquad \forall i \in I,\ s \in S,\ k \in K \tag{14}$$

(b) Before starting run $k$, the lower coordinate of batch $i$ ($F_{i,k-1} - W_{i,k-1}$) should have not surpassed the location of source $s$ ($\tau_s$). Then, ($F_{i,k-1} - W_{i,k-1}$) must never be greater than $\tau_s$.

$$F_{i,k-1} - W_{i,k-1} \le \tau_s + (PV - \tau_s)(1 - w_{i,s}^{(k)}) \qquad \forall i \in I,\ s \in S,\ k \in K \tag{15}$$

From eqs 14−15, it follows that $F_{i,k-1} - W_{i,k-1} \le \tau_s \le F_{i,k-1}$ whenever $w_{i,s}{}^{(k)} = 1$ and some product is supplied from input node $s$ to the existing batch $i$ flowing along the line. In case batch $i$ is a new one, $W_{i,k-1} = 0$ and consequently $F_{i,k-1} \le \tau_s \le F_{i,k-1}$, that is, $\tau_s = F_{i,k-1}$. To reduce the search computational cost, some very small tolerance $\varepsilon > 0$ is allowed and the practical feasibility condition for the insertion of a new batch at an intermediate source is given by: $F_{i,k-1} - \varepsilon \le \tau_s \le F_{i,k-1} + \varepsilon$.

Figure 5 shows the pipeline state before starting a new pumping run. Refined products can be injected from one of the sources *s1* or *s2*. Then, there are three alternatives: (a) pumping a new batch *B4* from the input node *s1*, (b) adding some amount of product *P2* from source *s1* to the existing batch *B3*, or (c) injecting more product *P1* from input terminal *s2* into the existing batch *B2*.

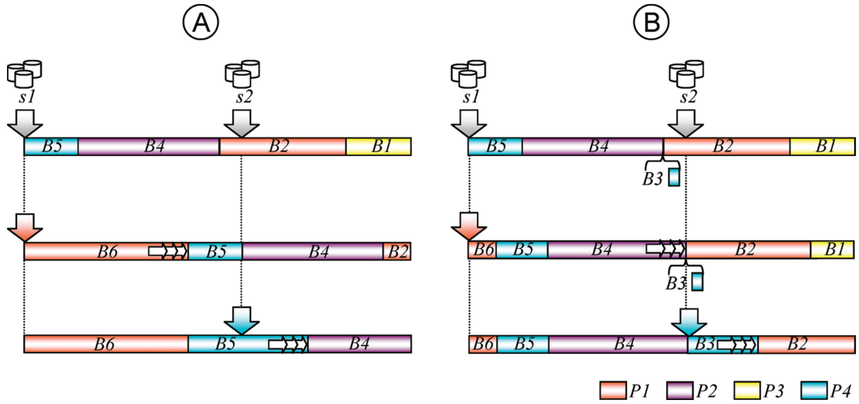


**Figure 6.** Feasible pumping runs at the intermediate source node *s2*.



Figure 6 illustrates two alternative input operations that can be performed at the intermediate source *s2*. The initial linefill is depicted at the top of Figure 6A. In the next line of Figure 6A, a new lot *B6* containing product *P1* is pumped into the line from source *s1* until the upper coordinate of *B5* reaches the location of *s2*. When this event occurs, the injection of *B6* is interrupted and an input operation at the input node *s2* is started to increase the size of the existing batch *B5* (see bottom of Figure 6A). In Figure 6B, the initial linefill includes an empty batch *B3* between lots *B4* and *B2*. As before, a new batch *B6* containing *P1* is injected in the line from source *s1*. In this case, the pumping run is stopped when the empty batch *B3* between batches *B4* and *B2* reaches the location of node *s2*. The next input operation takes place at source *s2* to add some amount of product *P4* to the empty batch *B3*, that is, to inject a new batch *B3* between lots *B4* and *B2* (see bottom of Figure 6B).

**Diverting Material from in-Transit Lots to Terminal Tanks.** The transfer of material from batch $i \in I$ to depot $j \in J$ during run $k \in K$ is feasible only if the physical connection to depot $j$ is reachable from batch $i$. The fulfillment of such a feasibility condition implies that

(a) the upper coordinate of batch $i$ at the end of run $k$ ($F_{i,k}$) should never be lower than the $j$th-depot coordinate $\sigma_j$, that is, $F_{i,k} \geq \sigma_j$, and

(b) the lower coordinate of batch $i$ at the end of run $(k-1)$ must be less than the depot coordinate $\sigma_j$, that is, $F_{i,k-1} - W_{i,k-1} \leq \sigma_j$.

Let $x_{i,j}^{(k)}$ be a binary variable denoting that the $j$th-terminal tankage is reachable from batch $i$ during pumping run $k$ ($x_{i,j}^{(k)} = 1$). Otherwise, $x_{i,j}^{(k)} = 0$ and no material can be transferred from batch $i$ to depot $j$ ($D_{i,j}^{(k)} = 0$). Therefore,

$$D_{\min} x_{i,j}^{(k)} \leq D_{i,j}^{(k)} \leq D_{\max} x_{i,j}^{(k)} \qquad \forall i \in I,\ j \in J,\ k \in K \tag{16}$$

where $D_{\max}$ is an upper bound on the amount of material that can be transferred from batch $i$ to depot $j$. Moreover, constraints 17 and 18 stand for the feasibility conditions (a) and (b) for diverting material from in-transit lots to depots, respectively.

$$F_{i,k} \geq \sigma_j x_{i,j}^{(k)} \qquad \forall i \in I,\ j \in J,\ k \in K \tag{17}$$

$$F_{i,k-1} - W_{i,k-1} \leq \sigma_j + (PV - \sigma_j)(1 - x_{i,j}^{(k)}) \qquad \forall i \in I,\ j \in J,\ k \in K \tag{18}$$

Though the delivery of batch $i$ to an output terminal $j$ is feasible, it may happen that the batch $i$ is destined for other depots. In such a case, the variable $x_{i,j}^{(k)}$ is driven to zero.

If pumping run $k$ injects a new batch $i'$ and the coordinates ($F_{i,k-1}$, $F_{i,k}$) of in-transit batch $i < i'$ at the end of runs $(k-1)$ and $k$ satisfy the following condition: $F_{i,k-1} - W_{i,k-1} < \sigma_j \leq F_{i,k}$, then an upper bound on the volume of product that can be transferred from batch $i$ to the output terminal $j$ during run $k$ is given by $[\sigma_j - (F_{i,k-1} - W_{i,k-1})]$. However, a different situation may arise for a pumping run $k$ enlarging the size of a batch $i$ already in the line. It may occur that run $k$ pumps an additional amount of material to batch $i$ from source $s$ ($Q_{i,s}^{(k)} > 0$) while some portion of batch $i$ is diverted to accessible downstream depots. As a result, the maximum volume that can be delivered from batch $i$ to accessible downstream terminals up to depot $j$ during run $k$ is given by

$$\sum_{j'=1}^{j} D_{i,j'}^{(k)} \leq \sigma_j - (F_{i,k-1} - W_{i,k-1}) + \sum_{\tau_s < \sigma_j} Q_{i,s}^{(k)} + (PV - \sigma_j)(1 - x_{i,j}^{(k)}) \qquad \forall i \in I,\ j \in J,\ k \in K \tag{19}$$

Given the linefill shown at the top of Figure 7, two alternative input operations can be executed: (A) the insertion of a new batch *B3* at the origin *s1*, or (B) the injection of additional material from *s2* to the existing batch *B1*. In Figure 7A, it is shown the maximum portion of batch *B1* that can be diverted to depot *j1* during the pumping of batch *B3*, that is, a bound on $D_{B1,j1}^{(k)}$ given by eq 19. The other feasible input operation is shown at the bottom of Figure 7B. In the latter case, the bound on $D_{B1,j1}^{(k)}$ is relaxed by $Q_{B1,s2}^{(k)}$.

**5.1.4. Product Supply and Demand Constraints. Assigning Products to Batches.** Every batch can at most contain a single product. Let $y_{i,p}$ be a binary variable denoting that batch $i$ contains product $p$ whenever $y_{i,p} = 1$. Then,

$$\sum_{p \in P} y_{i,p} \leq 1 \qquad \forall i \in I \tag{20}$$

If a predefined batch $i$ does not contain any product ($y_{i,p} = 0$ $\forall p \in P$), it is never injected into the pipeline. In other words, $\sum_{p \in P} y_{i,p} = 0$ implies that there is no run $k$ performed at any input terminal $s$ inserting batch $i$ in the line. Therefore, batch $i$ is a fictitious lot and $\sum_{s \in S} \sum_{k \in K} w_{i,s}^{(k)} = 0$. If instead product $p$ has been allocated to batch $i$, then there is at least one pumping run $k$ inserting batch $i$ in the line carrying some volume of product $p$. This conditions can be mathematically written as follows,

$$\sum_{p \in P} y_{i,p} \leq \sum_{s \in S} \sum_{k \in K} w_{i,s}^{(k)} \leq |K| \sum_{p \in P} y_{i,p} \qquad \forall i \in I^{\text{new}} \tag{21}$$

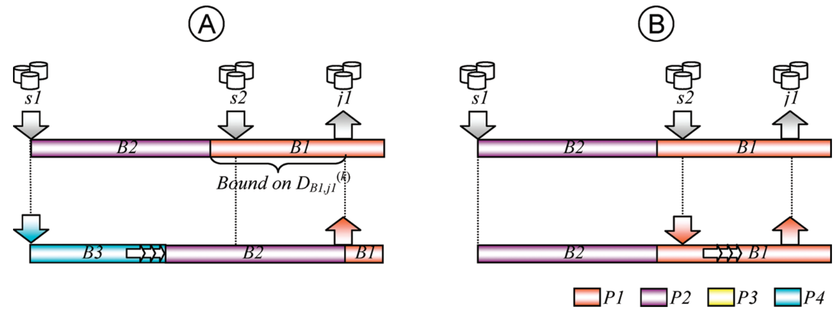


**Figure 7.** Upper bound on the extent of product delivery operations given by eq 19.

To reduce the size of the problem feasible region without cutting off the optimal solution, fictitious batches are confined to the end of the pumping run sequence through the following constraint,

$$\sum_{p\in P} y_{i,p} \le \sum_{p\in P} y_{i-1,p} \qquad \forall i \in I^{new}(i > 1) \qquad (22)$$

**Amount of Product *p* Injected in the Line through Run *k*.** If $y_{i,p} = 0$, the amount of product *p* in batch *i* inserted in the line from any source *s* through any run *k* will be equal to zero. Otherwise, $y_{i,p} = 1$ and the volume of product *p* pumped in the pipeline will be equal to $Q_{i,s}^{(k)}$, i.e. the volume of product inserted in the line assuming that run *k* takes place at the input terminal *s*. Both conditions are stated through eqs 23 and 24.

$$Q_{\min,p} y_{i,p} \le \sum_{s\in S}\sum_{k\in K} QP_{i,s,p}^{(k)} \le Q_{\max,p} y_{i,p} \qquad \forall i \in I^{new}, p \in P \qquad (23)$$

$$\sum_{p\in P} QP_{i,s,p}^{(k)} = Q_{i,s}^{(k)} \qquad \forall i \in I, s \in S, k \in K \qquad (24)$$

$Q_{\max,p}$ stands for the maximum admissible batch injection size for product *p*.

**Volume of Product *p* Diverted from in-Transit Batches to Depots.** No product *p* can be delivered from batch *i* to output terminal *j* through any run *k* if lot *i* does not contain *p*. Otherwise, $y_{i,p} = 1$ and the amount of product *p* supplied by batch *i* to depot *j* during run *k* is given by $D_{i,j}^{(k)}$. Both conditions are given by eqs 25 and 26, respectively.

$$\sum_{k\in K} DP_{i,j,p}^{(k)} \le D_{\max} y_{i,p} \qquad \forall i \in I, j \in J, k \in K, p \in P \qquad (25)$$

$$\sum_{p\in P} DP_{i,j,p}^{(k)} = D_{i,j}^{(k)} \qquad \forall i \in I, j \in J, k \in K \qquad (26)$$

**Feasible Range for the Amount of Product *p* Shipped from Source *s*.** Let us assume that $SU_{p,s}$ stands for the total amount of product *p* available in source *s* during the planning horizon. Besides, $SL_{p,s}$ stands for a lower bound on the amount of product *p* that should be pumped into the line from source *s* during the time horizon. Hence,

$$SL_{p,s} \le \sum_{k\in K}\sum_{i\in I} QP_{i,s,p}^{(k)} \le SU_{p,s} \qquad \forall p \in P, s \in S \qquad (27)$$

$SL_{p,s}$ is usually large enough to fulfill, in combination with other sources, the specified demands of product *p* not covered by the initial linefill. In some cases, however, it should be even greater to either get a suitable final linefill to meet future product demands or provide free capacity to receive new production runs from refineries feeding source *s*.

**Fulfilling Product Demands at Every Output Terminal.** Let $DL_{p,j}$ be the demand of product *p* at the output terminal *j* to be satisfied before the end of the time horizon. Since the storage capacity at any output terminal is finite, the total amount of product *p* diverted from the pipeline to depot *j* during the planning horizon should be bounded. Let us define the model parameter $DU_{p,j}$ as the maximum amount of product *p* that can be delivered and stored in tanks of depot *j*. Then, the total amount of product *p* delivered to terminal *j* from any batch containing *p* is constrained as follows,

$$DL_{p,j} - B_{p,j} \le \sum_{k\in K}\sum_{i\in I} DP_{i,j,p}^{(k)} \le DU_{p,j} \qquad \forall p \in P, j \in J \qquad (28)$$

When the demand of product *p* at depot *j* cannot be satisfied before the end of the planning period, a nonzero backorder $B_{p,j} > 0$ will arise. By including the continuous variable $B_{p,j}$ in eq 28, the pipeline scheduling problem will remain feasible even though some demands are unsatisfied at the horizon end.

**Initial Linefill.** Let $W_i^o$ be the volume of the old batch $i \in I^{old}$ already in the pipeline at the start of the scheduling horizon. Then, the upper coordinate of batch $i \in I^{old}$ can be obtained by summing the volume of every old batch $i' \in I^{old}$ succeeding batch *i*, plus the initial volume of batch *i*:

$$F_{i,k-1} = \sum_{\substack{i'\ge i\\ i'\in I^{old}}} W_{i'}^o \qquad \forall i \in I^{old}, k = 1 \qquad (29)$$

Moreover, it is known the product $P_i$ that is contained in every old batch $i \in I^{old}$.

$$y_{i,p} = 1 \qquad \text{for} \qquad p = P_i, \forall i \in I^{old} \qquad (30)$$

**5.2. Problem Objective Function.** The problem goal is to minimize the total pipeline operating cost including (i) the cost of underutilizing pipeline transportation capacity (*UC*), (ii) transition costs associated to the reprocessing of interface material between consecutive batches containing products *p* and *p′* ($cif_{p,p'}$), (iii) pumping costs, and (iv) backorder costs.

The pipeline remains active while pumping runs are being executed, that is, $\Sigma_k L_k$. If $h_{\max}$ stands for the length of the time horizon, and $\rho$ denotes the penalty cost per unit idle time, then the cost of underutilizing the available pipeline transport capacity is given by

$$UC = \rho(h_{\max} - \sum_{k\in K} L_k) \qquad (31)$$

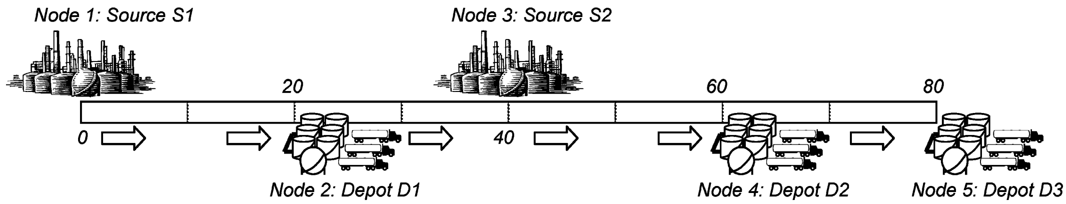


**Figure 8.** A pipeline example involving two sources and three destination points.

Let us assume that the reprocessing cost of the interface volume between consecutive batches containing products $p$ and $p'$ is a known problem datum ($cif_{p,p'}$) which does not depend on neither the pumping flow nor the travel distance. Then, a loss of $cif_{p,p'}$ on the amount of product contained in batch $i$ will arise if batch $i$ contains product $p$ and is succeeded by another batch $i'$ transporting product $p'$.

$$TC_i \geq cif_{p,p'}(y_{i,p} + y_{i+1,p'} - 1) \qquad \forall i \in I;\, p, p' \in P \tag{32}$$

If $cin_{p,s}$ represents the average cost of injecting a unit volume of product $p$ from source $s$ to downstream terminals demanding $p$ ($j \in J_{s,p} \subseteq J_p$), then the total pipeline pumping costs are given by

$$PC_k = \sum_{i \in I} \sum_{s \in S} \sum_{p \in P} cin_{p,s} QP^{(k)}_{i,s,p} \qquad \forall k \in K \tag{33}$$

Weighting coefficients $\eta_{p,s,j}$ based on the $p$th-product demands at downstream terminals given by $\eta_{p,s,j} = DL_{p,j}/(\Sigma_{j' \in Js,p} DL_{p,j'})$ $\forall p \in P,\ s \in S,\ j \in J_{s,p}$ are used to evaluate the average unit pumping cost $cin_{p,s} = \Sigma_{j \in Js,p}\ \eta_{p,s,j}\ cin_{p,s,j}$. The parameter $cin_{p,s,j}$ stands for the cost of pumping a single unit of product $p$ from source $s$ to destination $j$.

On the other hand, backorder costs $BC$ are proportional to the unsatisfied product demands at the horizon end. If $cb_{p,j}$ stands for the cost of failing to provide a single unit of product $p$ at depot $j$ on time, then,

$$BC = \sum_{p \in P} \sum_{j \in J} cb_{p,j} B_{p,j} \tag{34}$$

Therefore, the problem goal will be given by

$$\min z = UC + \sum_{i \in I} TC_i + \sum_{k \in K} PC_k + BC \tag{35}$$

that is equivalent to

$$\min z = \rho(h_{\max} - \sum_{k \in K} L_k) + \sum_{i \in I} TC_i + \sum_{k \in K} \sum_{i \in I} \sum_{s \in S} \sum_{p \in P} cin_{p,s} QP^{(k)}_{i,s,p} + \sum_{p \in P} \sum_{j \in J} cb_{p,j} B_{p,j} \tag{36}$$

## 6. Results and Discussion

The proposed MILP formulation for the scheduling of multisource pipelines was applied to a pipeline system involving two inputs and three output terminals introduced by Jittamai.[12] It consists of a unidirectional transmission line composed by four pipeline segments each one featuring a capacity of 20 volumetric units (see Figure 8). The first segment connects the input node 1 at the origin (source *S1*) with the output node 2 (depot *D1*). The second one goes from the output node 2 to the input node 3, while the last two segments convey products to the destination nodes 4 and 5. Therefore, the total pipeline capacity is equal to 80 units.

**Table 1. Product Demands and Transition Costs**

| | product demands (in units) | | | interface costs ($10^2$ \$) | | | |
|---|---|---|---|---|---|---|---|
| | destinations | | | | successor | | |
| | depot 1 | depot 2 | depot 3 | predecessor | A | B | C |
| A | 30 | 30 | | A | | 22.0 | 35.0 |
| B | | | 50 | B | 24.0 | | 21.0 |
| C | | 30 | | C | 30.0 | 32.0 | |

Three types of liquid fuels called *A*, *B*, and *C* are transported by the pipeline from refineries *S1* and *S2* to meet product demands at depots *D1*, *D2*, and *D3*, before the end of the planning horizon. The horizon length is 120 h and the flow rate at every pipeline segment should belong to the range 0.80−1.20 units per hour. Besides, product demands are shown in Table 1. The problem goal is to schedule input and delivery operations over the planning horizon in such a way that the pipeline works at full capacity and depot demands are satisfied at minimum pumping and transmix reprocessing costs. In contrast to Jittamai's approach, the cyclic schedule assumption no longer holds. Instead, the best sequence and timing of pipeline operations are found by solving an optimization model that accounts for acyclic schedules.

The initial linefill includes four in-transit batches (*B5*, *B4*, *B2*, and *B1*) containing products *A*, *B*, *A*, and *B*, respectively, with *B5* being the nearest lot to the origin (source *S1*). Their volume sizes are 20, 10, 30, and 20, in that order. An initial batch *B3* with zero volume comes from the previous scheduling horizon. It has been reserved for a planned injection of product *C* from the intermediate refinery *S2*. Such a pumping run will start just as batch *B3* reaches the location of *S2* (see Figure 9).

The transition cost between two different products is related to either the transmix reprocessing or the downgrading quality cost due to batch mixing at the interfaces. Thus, the mixing cost between a pair of consecutive shipments strongly depends on the products carried by them. For simplicity, it will be assumed that the transition cost is independent of both the flowrate and the distance between input and output nodes. Let us suppose that a lot of product *B* is directly preceded by a lot of *A*, that is, a transition *A*−*B*. The generated transmix at the interface has a reprocessing cost equal to 22.0 ($10^2$ \$) (see Table 1). However, the interface cost when a lot of *A* is directly preceded by a lot of *B* amounts to 24.0 ($10^2$ \$). Therefore, the number of transitions should be minimized and the best batch input sequence must be selected in order to get substantial savings in interface costs. In Jittamai's approach, transitions costs were ignored.

On the other hand, the pumping cost is a function of the number of pumps on service. We assume that the unit pumping cost, expressed in dollars per unit volume pumped in the line, depends on the product being injected and the refinery from which it is inputted (see Table 2).

The selected objective function does not explicitly consider pump activating/deactivating costs. Instead, it has been specified that there is a minimum size on batch injections ($Q_{\min}$) and deliveries ($D_{\min}$) equal to 10 volumetric units to reduce the

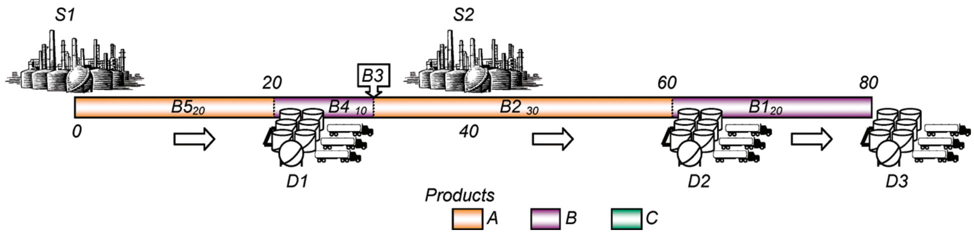


**Figure 9.** Initial linefill for the multisource pipeline case study.

**Table 2. Product-Dependent Pumping Costs at Every Refinery**

| product ($p$) | pumping costs ($10^2$ \$ per unit volume) | |
|---|---|---|
| | refinery source 1 | refinery source 2 |
| A | 29.0 | 14.5 |
| B | 34.0 | 17.0 |
| C | 49.0 | 24.5 |

**Table 3. Product Inventories at Every Refinery (Example 1)**

| product ($p$) | product supplies (in units) | |
|---|---|---|
| | refinery source 1 | refinery source 2 |
| *A* | 30 | |
| *B* | 70 | |
| *C* | | 40 |

number of pump and line segment switchings from active to idle condition and vice versa. Besides, the model assumes that the refined products will be available in the source nodes at the times they will be injected. To avoid large batch inputs leading to inadequate linefills for satisfying future demands, a maximum lot injection size ($Q_{max}$) equal to 30 units is also specified.

In the next sections 6.1 and 6.2, two variants of Jittamai's case study have been tackled. They differ in the selected mode of pipeline operation: segregated mode (example 1) and fungible mode (example 2). In example 1, each product can be supplied just by a single source, while both refineries can provide lots of the same standard fuel in example 2. In both examples, however, the amounts of products *A*, *B*, and *C* injected in the line throughout the planning horizon will be the same. Longer time horizons are considered in section 6.3 to determine its impact on the required computational effort.

**6.1. Example 1: Pipeline Operation on Segregated Mode.** In example 1, the pipeline system is assumed to operate on segregated mode. The two refineries have been specialized in the production of a narrow range of products. Then, the product sets supplied by them have no common chemical species. Source node or refinery *S1* provides the refined products *A* and *B*, while *S2* supplies only product *C*. Product inventories available in nodes *S1* and *S2* are shown in Table 3.

Therefore, a total volume of 140 units of products is available. Contrarily to Jittamai's example, each product batch can have several destinations, that is, the one-to-many segregated mode. Nonetheless, the enlargement of lots in transit at the intermediate source *S2* is still not possible.

The optimal schedule of input/delivery operations that was obtained by solving the proposed MILP continuous-time formulation is depicted in Figure 10. It includes a sequence of six batch inputs, with four shipments (*B6*, *B6*, *B8*, and *B9*) being pumped from source node *S1* and two batches (*B3* and *B7*) from *S2*. Batch *B6* containing 50 units of product *B* is inserted by performing two nonconsecutive pumping runs. The first run is interrupted at the time the interface between batches *B2* and *B4*, containing products *A* and *B*, arrives at node *S2*. Stopping the insertion of *B6* after pumping 30 units of fuel *B* makes it possible to inject batch *B3* transporting 30 units of product *C* to meet a similar demand at depot *D2*. As said before, it was assumed that the linefill at the end of the previous horizon includes an empty batch *B3* traveling to the intermediate source *S2*.

The batch plan provided by the proposed MILP approach includes the following sequence of input and delivery operations:

(i) From time $t = 0.00$ h to time $t = 25.00$ h, a batch *B6* containing 30 volumetric units of product *B* is injected from refinery *S1* at the maximum flow-rate of 1.20 units/h. During the pumping of batch *B6*, the following set of delivery operations is performed: 10 units of product *A* coming from batch *B2* are loaded in a tank of depot *D2*, and 20 units of product *A* are

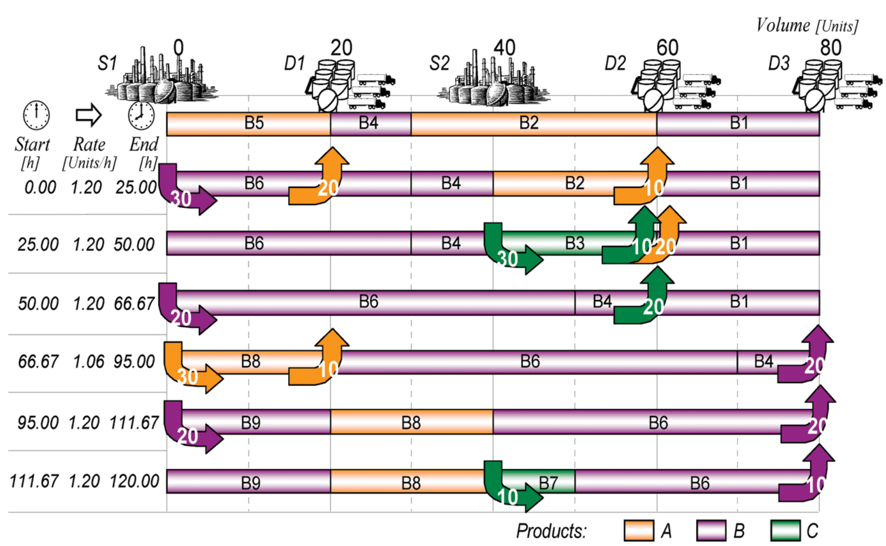


**Figure 10.** Optimal input and delivery schedule for example 1.

**Table 4. Model Statistics and Computational Results for Examples 1 and 2**

| example | eqs. | cont. variables | binary variables | CPU time (s) | number of iterations | optimal solution ($10^2$ \$) | pipeline usage (%) | interface cost ($10^2$ \$) |
|---|---|---|---|---|---|---|---|---|
| 1 | 2180 | 1076 | 297 | 12.5 | 54188 | 4440.0 | 100 | 210.0 |
| 2 | 2421 | 1280 | 304 | 45.3 | 180168 | 4267.0 | 100 | 202.0 |

**Table 5. Product Inventories Available at the Refineries (Example 2)**

| | product supplies (in units) | |
|---|---|---|
| product (*p*) | refinery source 1 | refinery source 2 |
| *A* | 20 | 10 |
| *B* | 40 | 30 |
| *C* | 20 | 20 |

diverted from lot *B5* to depot *D1*. While performing the stripping operation at depot *D2*, there is no flow in the pipeline segment connecting output terminals *D2* and *D3*. On the other hand, the pipeline segments between *D1* and *D3* remain idle during the execution of the delivery operation at depot *D1*.

(ii) From time $t = 25.00$ h to time $t = 50.00$ h, an input operation is scheduled at the intermediate source node *S2*. As mentioned before, the empty batch *B3* located at the interface between lots *B2* and *B4* has moved forward during the previous pumping run up to the volumetric coordinate of depot *S2* ($V = 40$). In this way, the injection of batch *B3* featuring a volume of 30 units of product *C* from refinery *S2* can be executed. During the pumping of *B3* at a flow-rate of 1.20 units/h, two delivery operations at depot *D2* are scheduled. Tanks at *D2* will first receive 20 units of product *A* from batch *B2*, and then 10 units of product *C* from *B3*. While injecting the last 10 units of *B3* into the line, the same amount is diverted from *B3* to depot *D2*.

(iii) From time $t = 50.00$ h to time $t = 66.67$ h, the injection of batch *B6* at refinery *S1* is restarted, and 20 additional units of product *B* are inserted in the line. The delivery operation to execute while injecting the last portion of *B6* consists of diverting the remaining 20 units of batch *B3* containing product *C* to depot *D2*. At time $t = 66.67$ h, batches *B5*, *B3*, and *B2* have all vanished. One of the advantages of the new formulation is that the same batch can be inputted in the pipeline through two nonconsecutive pumping runs. Previous continuous approaches assume that every input operation is associated to a different batch. Therefore, a lower number of potential batches can be postulated by using the proposed formulation to obtain the best pipeline schedule at lower computational cost.

(iv) At time $t = 66.67$, the pipeline is completely filled with product *B*. From $t = 66.67$ h to $t = 95.00$ h, it is scheduled to input 30 units of product *A* by injecting batch *B8* from refinery *S1* at a flow-rate of 1.06 units/h. While performing this input operation, two product deliveries are to be made: 20 units of product *B* are diverted from batch *B1* to depot *D3*, and 10 units of *A* coming from the inputted batch *B8* are loaded in a tank of terminal *D1*. The entire pipeline is active during the stripping operation at *D3*, while a nonzero flow just occurs through the first segment *S1*−*D1* when product *A* is received at depot *D1*. It is important to remark that there is an empty batch *B7* between lots *B8* and *B6* traveling to source node *S2*. The MILP solution has postponed the injection of *B7* until it reaches the *S2*-coordinate because it has been reserved for a batch of product *C* only available at source *S2*.

(v) Without any interruption, another input operation at refinery *S1* involving the insertion of batch *B9* with 20 units of product *B* is scheduled from time $t = 95.00$ h to $t = 111.67$ h, at a flow-rate of 1.20 units/h. While performing this pumping run, 20 units of product *B* are stripped from batches *B4* and *B6*, and stored at the last depot *D3*. During the whole run, there is a finite flow all along the pipeline.

(vi) The last pumping run takes place at the intermediate source node *S2*. At time $t = 111.67$ h, the empty batch *B7* arrives at node *S2* and the injection of product *C* becomes feasible. From $t = 111.67$ h to $t = 120.00$ h, batch *B7* containing 10 units of *C* is then pumped from refinery *S2* at a flow-rate of 1.20 units/h. During this run, it is scheduled to deliver 10 units of product *B* from batch *B6* to depot *D3*. Hence, just the pipeline segments between nodes *S1* and *S2* will remain idle. The final linefill comprises batches *B9*, *B8*, *B7*, and *B6* transporting 20, 20, 10, and 30 units of products *B*, *A*, *C*, and *B*, respectively.

Model statistics and computational results are summarized in Table 4. The MILP formulation includes 297 binary variables, 1076 continuous variables, and 2180 linear constraints. The optimal schedule makes full use of the pipeline capacity and was found in 12.5 CPU s using GAMS/CPLEX 11.0[13] on an Intel 2.80 GHz processor. The minimum pipeline operating costs amount to 4440.0 ($10^2$ \$), from which 210.0 ($10^2$ \$) correspond to transmix reprocessing costs. From Figure 10, it follows that the batch sequence *A*−*B*−*C*−*A*−*B* resulting from the optimal batch input plan favors the transitions *A*−*B*, *B*−*C*, and *C*−*A* with the lowest interface costs. In his heuristic approach, Jittamai[12] considered delivery time windows for product requests at output terminals, and a pipeline scheduling goal aimed at minimizing time-window constraint violations. The resulting pipeline schedule includes a larger number of batch injections to timely start dispatching products. As a result, shippers requests are often satisfied through several partial deliveries, and transition costs substantially increase. To mitigate these operational inconveniences, Jittamai's procedure includes a batch consolidation stage where close batches of the same product are merged into a single one. This remedial action produces a reduction in pumping runs but at the same time, it causes to miss some delivery time windows. Two batch merging options were considered. One of them leads to a batch sequence of the type [*B*−*C*−*A*−*B*−*C*−*A*...], quite similar to the one generated by our approach. Because Jittamai[12] ignored mixing costs, however, it does not seem fair to make a further comparison with his results.

**6.2. Example 2: Pipeline Operation on Fungible Mode.** The case study introduced by Jittamai[12] is tackled again in example 2, but this time the pipeline is operated on fungible mode. Besides, both sources *S1* and *S2* can supply the whole set of products (*A*, *B*, and *C*) demanded by the output terminals. Therefore, product injections from refinery *S2* can either insert new batches at intermediate linefill positions or provide additional amounts of product to increase the size of batches previously pumped at node *S1*. Since the pipeline is working on fungible mode, refineries *S1* and *S2* are assumed to provide similar refined products fulfilling the same standard specifications.

One of the advantages of pipeline operation on fungible mode is the reduction in the number of batches and interfaces, which produces important savings in transmix reprocessing and quality downgrading costs. However, such savings can only be achieved through an effective coordination of product shipments from different refiners. Merging lots of the same refined product pumped from different source nodes and, at the same time, meeting product demands at output terminals require a very

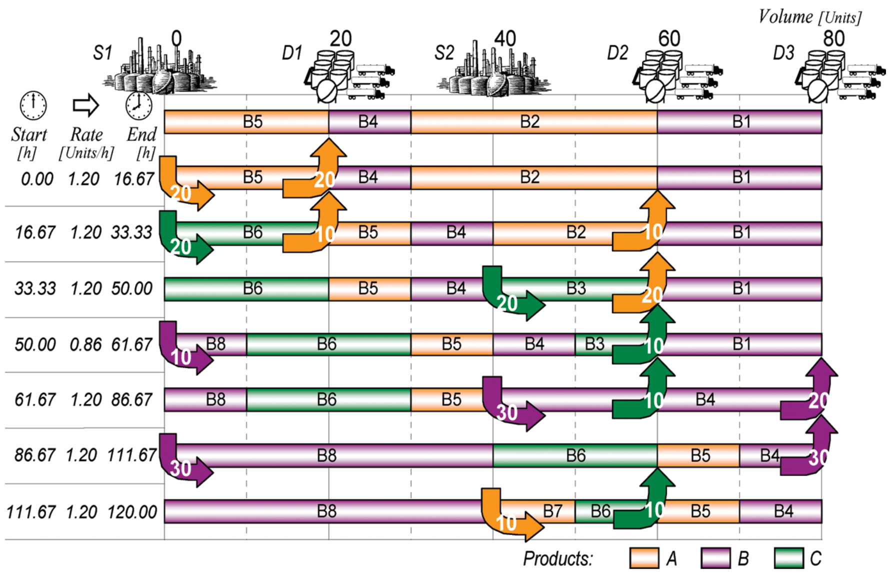


**Figure 11.** Optimal schedule of input and delivery operations for example 2.

**Table 6. Product Demands for Longer Time Horizons**

| | horizon length = 168 h | | | | horizon length = 240 h | | |
|---|---|---|---|---|---|---|---|
| | destinations | | | | destinations | | |
| | depot 1 | depot 2 | depot 3 | | depot 1 | depot 2 | depot 3 |
| A | 42 | 42 | | A | 60 | 60 | |
| B | | | 70 | B | | | 100 |
| C | | 42 | | C | | 60 | |

careful selection of the sequencing, sizing, timing, and location of the batch inputs. Therefore, the development of the optimal input and delivery schedules for pipeline operation on fungible mode is a very complex industrial problem.

Table 5 reports the inventories of products *A*, *B*, and *C* available in refineries *S1* and *S2* for the planning horizon. Similar to example 1, there is a total inventory of 140 units, but its distribution between the two refineries is rather different. In this case, not only stocks of products *A* and *B* but also of product *C* are contained in tanks of refinery *S1*. Again, product demands at output terminals to meet not later than time $t = 120$ h, and the initial linefill including an empty batch (*B3*) at the interface between lots *B4* and *B2*, are those shown in Table 1 and Figure 9, respectively.

The optimal batch schedule for example 2 is depicted in Figure 11. Because product demands can be fulfilled from two alternative source nodes (*S1* and *S2*), the number of planned injection runs rises to seven. The possibility of injecting products *A*, *B*, and *C* from refineries *S1* and *S2* produces an increase in the number of postulated runs and a larger size of the problem formulation. However, the resulting number of batches moving along the line over the planning horizon is even lower than the one required at the optimal solution of example 1. Despite seven pumping runs being scheduled, the number of new batches will just rise by four because some batch injections at the intermediate source *S2* merely make in-transit batches larger. This produces an extra saving in mixing costs.

The optimal input plan specifies the following sequence of batch injections from refineries *S1* and *S2*:

(i) From $t = 0.00$ h to $t = 16.67$ h, 20 additional units of product *A* in batch *B5* are injected from refinery *S1* at a flowrate of 1.20 units/h. The delivery schedule indicates that the first output terminal *D1* should receive a similar volume of fuel *A* from the inputted batch *B5* during the same period.

(ii) The next pumping run will also take place at source node *S1* from $t = 16.67$ h to $t = 33.33$ h, to insert a new batch *B6* containing 20 volumetric units of product *C*. During the pumping of batch *B6*, depot *D2* should receive 10 units of product *A* coming from batch *B2* and a volume of 10 units of the same product *A* will be diverted from batch *B5* and stored in depot *D1*. To minimize the number of idle segments, pipeline operators are capable of siphoning fluid out of the pipeline at depot *D1* as the batches move forward through the line and the delivery operation at depot *D2* is performed (Hane and Ratliff[1]).

(iii) At the end of the second run, the empty batch *B3* located at the interface between lots *B2* and *B4* reaches the location of refinery *S2*. From $t = 33.33$ h to $t = 50.00$ h it is then scheduled an input operation at the intermediate source *S2*. A new batch *B3* transporting 20 units of product *C* is inserted and, at the same time, the remaining 20 units of product *A* in batch *B2* are delivered to depot *D2*.

(iv) Input operations at refinery *S1* are restarted at time 50.00 h. From $t = 50.00$ h to $t = 61.67$ h, a batch consisting of 10 units of product *B* labeled *B8* is injected, while 10 units of product *C* will be stripped from *B3* and loaded in a tank of depot *D2*. Again, there will be no flow in the last pipeline segment *D2*−*D3*. As before, batch *B7* has been reserved to be pumped at the intermediate source node *S2*. Therefore, it will travel as an empty batch from the origin to node *S2*.

(v) At time $t = 61.67$ h, the injection of product *B* from refinery *S1* is stopped and pumping operations start at refinery *S2*. A 30-unit lot of the same fuel *B* is inserted to increase the size of one of the initial linefill batches, that is, the batch *B4*. Two delivery operations are planned to be executed during this pumping run. The remaining 10 units of product *C* contained in lot *B3* are stored in depot *D2*, and a volume of 20 units of product *B* are diverted from lot *B1* to depot *D3*.

(vi) The next pumping run will return to the head terminal *S1* to complete the injection of batch *B8*. An additional volume of 30 units of product *B* is added to *B8*, and a similar volume of fuel *B* is stripped from the farthest batch *B4* and delivered to depot *D3*.

(vii) At time $t = 111.67$ h, the empty batch *B7* located just at the interface between batches *B6* and *B8* reaches the coordinate of refinery *S2*. Therefore, a new product injection can start at the intermediate source *S2* to insert batch *B7* transporting 10 units of fuel *A*. At the same time, a volume of

**Table 7. Effect of the Horizon Length on the Optimal Pipeline Schedule**

| operational mode | horizon length (h) | batches $\|\|I\|\|$ | pumping runs $\|\|K\|\|$ | optimal solution ($10^2$ \$) | interface cost ($10^2$ \$) | pumping cost ($10^2$ \$) | CPU time (s) |
|---|---|---|---|---|---|---|---|
| segregated | 120 | 9 | 6 | 4440.0 | 210.0 | 4230.0 | 12.5 |
| | 168 | 9 | 8 | 6132.0 | 210.0 | 5922.0 | 47.1 |
| | 240 | 10 | 8 | 8638.0 | 228.0 | 8410.0 | 97.9 |
| fungible | 120 | 8 | 7 | 4267.0 | 202.0 | 4065.0 | 45.3 |
| | 168 | 8 | 8 | 5982.0 | 182.0 | 5800.0 | 134.5 |
| | 240 | 9 | 10 | 8165.0 | 235.0 | 7930.0 | 341.5 |

10 units of product *C* is stripped from batch *B6* and stored in depot *D2*. The end linefill involves five batches (*B8*, *B7*, *B6*, *B5*, and *B4*) containing 40, 10, 10, 10, and 10 volumetric units of products *B*, *A*, *C*, *A*, and *B*, respectively.

Model statistics and computational results for example 2 are summarized in Table 4. By adopting a fungible operation mode, the best schedule features a total cost of 4267.0 ($10^2$ \$) what represents a weekly saving of 173.0 ($10^2$ \$) with regards to the segregated mode. This implies a 3.90% reduction of the total operating cost. Some savings come from lower transition costs which drop from 210.0 ($10^2$ \$) to 202.0 ($10^2$ \$). Despite the same volume of products being injected throughout the planning horizon, the number of pumping runs rise by one in the fungible mode. In both examples, three shipments of product *B* and two of product *C* are injected. However, two batches of *A* (one from each source) rather than a single lot are inserted on fungible mode. Surprisingly, the number of new batches flowing along the line is only four because some batch injections at source *S2* just increase the size of in-transit batches, thus reducing the mixing costs. A brief analysis of Figure 11 shows that a great portion of product demands at downstream depots *D2* and *D3* are satisfied by stripping operations on batches injected at the intermediate source *S2*. As a result, the total pumping cost reduces from 4230.0 to 4065.0 despite a batch of the heaviest fuel *C* with the highest pumping cost being now injected at the head input terminal *S1*. In the segregated mode, the injections of product *C* both occur at the intermediate refinery *S2*.

The MILP mathematical formulation for example 2 comprises 304 binary variables, 1280 continuous variables, and 2421 linear constraints. The model size shows a limited increase with regards to example 1 because the three products (*A*, *B*, and *C*) are now available at both refineries, and the cardinality of the run set $||K||$ must be slightly increased. Similar to example 1, a full use of the pipeline transport capacity is planned. The optimal schedule has been found in 45.3 CPU s and 180168 iterations. As explained before, the larger CPU time comes from the fact that the three products can now be injected from both refineries *S1* and *S2*.

To develop a detailed delivery schedule, the sequence and timing of the stripping operations to be carried out during each pumping run must be first established. Besides, stripping operations at some output terminals may be performed in two or more nonconsecutive steps. As discussed before, the goal of this new problem aims to minimizing the total number of pipeline segment stoppages over the time horizon. This planning phase is addressed on a next paper. Another major point to remark is the fact that intermediate input terminals may be injecting in the line batches of products coming from other interconnected trunk lines. Therefore, the approach represents a major step toward planning the operation of pipeline networks.

**6.3. Scheduling the Pipeline Operation over Longer Time Horizons.** To study the impact of the horizon length on the model solution time, examples 1 and 2 were solved again

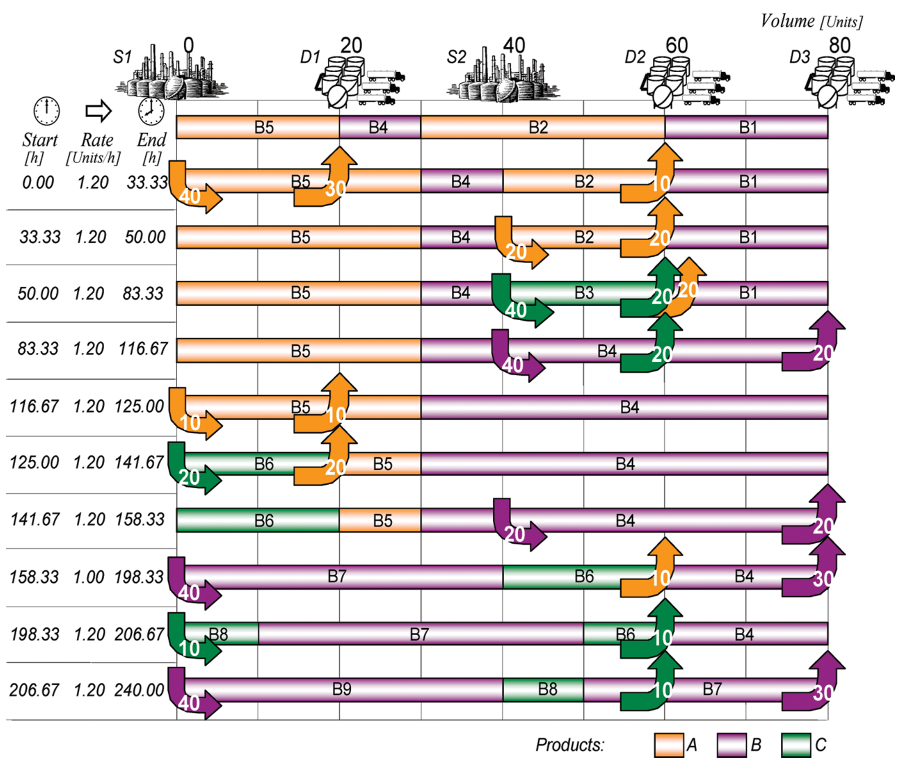


**Figure 12.** Optimal schedule for example 2 over a 10-day time horizon.

but now considering time horizons of 168 and 240 h, respectively. Since the time-horizon length increases, new product demands at distribution terminals should be considered. We will assume that the additional product requirements mimic the demand pattern for the first 120 h-interval of the new time horizon, that is, the demand data for examples 1 and 2 (see Table 1). The product demands for the extended time horizons of 168 and 240 h are shown in Table 6. The maximum input size ($Q_{max}$) has been increased in both cases to 40 units to allow the injection of larger batches, thus limiting the rise of the interface cost.

Computational results including the optimal operational costs, the interface and pumping cost contributions, the required CPU times, and the amounts of batches and batch injections are all shown in Table 7. For longer time horizons, the quantities of batches and pumping runs both increase. Consequently, the model size becomes larger and more CPU time is needed to find the best pipeline schedule. Nonetheless, the computational cost still remains reasonable.

Regardless of both the horizon length and the operational mode, there is always a full usage of the pipeline capacity and the average flow rate steadily remains at 1.167 [units/h] in all cases. This is the minimum flow rate that prevents from product backorders at distribution terminals. Figure 12 depicts the best pipeline schedule for fungible mode over a 10-day time horizon.

## 7. Conclusions

A new MILP mathematical formulation for the multisource pipeline scheduling problem has been developed. The problem goal is to schedule input and delivery operations over the planning horizon in such a way that the pipeline works at full capacity and depot demands are timely satisfied at minimum pumping and transmix reprocessing costs. The proposed MILP model has been obtained by generalizing the approach of Cafaro and Cerdá[8] for the scheduling of pipelines with a single input node at the origin, and multiple distribution terminals. Therefore, the approach still uses a continuous mathematical representation in both volume and time domains.

To convey batches of different refined products from various sources to several depots, the operation of multisource pipeline systems involves the execution of a sequence of pumping runs each one injecting a refined product from just a single source. A major difference with regard to the single-source case is the need of specifying the input terminal where the pumping run is driven. Another important feature of multisource pipelines is the injection of new batches from intermediate sources at nonorigin points. As a result, batches in the line will no longer be sequenced in the same order that they are injected. Moreover, some product injections from intermediate sources do not generate new interfaces but only increase the size of in-transit lots. These facts may greatly complicate the evaluation of interface reprocessing costs. Since they are not arranged in the same order, the approach handles pumping runs and product batches as separate mathematical entities. In this way, the proposed formulation is able to still manage a predefined batch sequence by booking batches to be inserted at intermediate sources. A reserved empty lot travels from the origin to the assigned input terminal. When it reaches the assigned source location, the pumping run occurs and the batch now containing a finite volume of a certain product is inserted into the line. By assigning products to batches, the product sequence is defined and the evaluation of mixing costs becomes a much simpler task.

The definition of separate sets of pumping runs and product batches produces a limited growth in both the model size and the computational cost. Since the required number of pumping runs and batches to be injected is not precisely known before solving the problem, they should be arbitrarily adopted. A simple rule has been given to properly guess the number of pumping runs and new batches required. The proposed MILP formulation for the scheduling of multisource pipelines was successfully applied to a pipeline system involving two inputs and three output terminals, and transporting three different oil refined products. Two alternative pipeline operation modes were considered. Compared with the segregated mode, the fungible mode permits an increase in the size of in-transit batches through product injections at intermediate sources. In this way, some savings in mixing costs can be achieved.

## Acknowledgment

Financial support received from FONCYT-ANPCyT under Grant PICT 01857, from CONICET under Grant PIP-5729, and from Universidad Nacional del Litoral under CAI+D 003-13 is fully appreciated.

## Nomenclature

*Sets*

$K$ = chronologically ordered pumping runs

$I$ = product batches arranged in the same order that they move along the pipeline ($I = I^{old} \cup I^{new}$)

$I^{new}$ = new batches to be injected during the planning horizon

$I^{old}$ = old batches in the initial linefill

$P$ = refined petroleum products

$P_i$ = refined petroleum product contained in old batch $i \in I^{old}$

$S$ = refinery sources or input terminals along the pipeline

$J$ = destination depots or output terminals along the pipeline

$J_p$ = destination depots demanding product $p$

$J_{s,p}$ = destination depots demanding product $p$ to be supplied by an upstream source $s$

*Parameters*

$cb_{p,j}$ = unit backorder penalty cost for tardily satisfying a requirement of product $p$ at output terminal $j$

$cif_{p,p'}$ = total reprocessing cost of interface material involving products $p$ and $p'$

$cin_{p,p'}$ = pumping cost per unit of product $p$ injected from input terminal $s$

$D_{min}$, $D_{max}$ = minimum/maximum delivery size to output terminals

$DL_{p,j}$ = minimum request of product $p$ from the output terminal $j$

$DU_{p,j}$ = maximum amount of product $p$ that can be delivered to output terminal $j$

$h_{max}$ = planning horizon length

$PV$ = total pipeline volume from the origin to the farthest depot

$Q_{min}$, $Q_{max}$ = minimum/maximum batch injection size

$Q_{min,p}$, $Q_{max,p}$ = minimum/maximum batch injection size for product $p$

$SL_{p,s}$ = lowest amount of product $p$ to be shipped from input terminal $s$

$SU_{p,s}$ = maximum amount of product $p$ that can be shipped from input terminal $s$

$vb_{min,s}$, $vb_{max,s}$ = minimum/maximum product flow-rates at source $s$

$W_i^o$ = initial volume of old batch $i$

$\eta_{p,s,j}$ = weighting coefficient for pumping product $p$ from source $s$ to depot $j$

$\rho$ = unit-time penalty cost for underutilizing pipeline transport capacity

$\sigma_j$ = volume coordinate of output terminal $j$ from the pipeline origin

$\tau_s$ = volume coordinate of input terminal $s$ from the pipeline origin

*Continuous Variables*

$BC$ = backorder penalty cost for tardily meeting product demands

$B_{p,j}$ = backorder of product $p$ at the output terminal $j$

$C_k$, $L_k$ = completion time/length of pumping run $k$
$D_{i,j}^{(k)}$ = volume of batch $i$ diverted to output terminal $j$ during run $k$
$DP_{i,j\ p}^{(k)}$ = volume of product $p$ diverted from batch $i$ to terminal $j$ during run $k$
$F_{i,k}$ = upper coordinate of batch $i$ from the origin, at time $C_k$
$L_k$ = length of pumping run k
$L_{k,s}$ = length of pumping run $k$ taking place at source s
$PC_k$ = total pumping cost during run $k$
$Q_{i,s}^{(k)}$ = size of batch $i$ shipped from input terminal $s$ during run $k$
$QP_{i,s,p}^{(k)}$ = size of batch $i$ containing product $p$ inputted from input terminal $s$ through run $k$
$TC_i$ = total reprocessing cost for the interface between batch $i$ and the following $(i + 1)$
$UC$ = total cost of underutilizing the pipeline transport capacity
$W_{i,k}$ = size of batch $i$ at the end of run $k$

*Binary Variables*

$w_{i,s}^{(k)}$ = variable denoting that a new (or a portion of the existent) batch $i$ is injected from source $s$ through run $k$
$x_{i,j}^{(k)}$ = variable denoting that a portion of batch $i$ is diverted to the output terminal $j$ during the pumping run $k$
$y_{i,p}$ = variable denoting that batch $i$ contains product $p$



## Literature Cited

(1) Hane, C. A.; Ratliff, H. D. Sequencing Inputs to Multi-Commodity Pipelines. *Ann. Operat. Res.* **1995**, *57*, 73–101.

(2) Sasikumar, M.; Prakash, P. R.; Patil, S. M.; Ramani, S. PIPES: A Heuristic Search Model for Pipeline Schedule Generation. *Knowledge-Based Syst.* **1997**, *10*, 169–175.

(3) Maruyama Mori, F.; Luders, R.; Arruda, L. V. R.; Yamamoto, L.; Bonacin, M. V.; Polli, H. L.; Aires, M. C.; Bernardo, L. F. J. Simulating the Operational Scheduling of a Realworld Pipeline Network. *Comput.-Aided Chem. Eng.* **2007**, *24*, 691–696.

(4) García-Sánchez, A.; Arreche, L. M.; Ortega-Mier, M. Combining Simulation and Tabu Search for Oil-Derivatives Pipeline Scheduling. *Stud. Comput. Intell.* **2008**, *128*, 301–325.

(5) Rejowski, R.; Pinto, J. M. A Novel Continuous Time Representation for the Scheduling of Pipeline Systems with Pumping Yield Rate Constraints. *Comput. Chem. Eng.* **2008**, *32*, 1042–1066.

(6) Magatão, L.; Arruda, L. V. R.; Neves, F. A Mixed Integer Programming Approach for Scheduling Commodities in a Pipeline. *Comput. Chem. Eng.* **2004**, *28*, 171–185.

(7) Zyngier, D.; Kelly, J. D. Multi-Product Inventory Logistics Modeling in the Process Industries, In *Proceedings of DIMACS and ExxonMobil Workshop on Computational Optimization and Logistics Challenges in the Enterprise (COLCE)*; Furman, K. C., Grossmann, I. E., Eds.; Springer, USA, 2006.

(8) Cafaro, D. C.; Cerdá, J. Optimal Scheduling of Multiproduct Pipeline Systems Using a Non-Discrete MILP Formulation. *Comput. Chem. Eng.* **2004**, *28*, 2053–2068.

(9) Cafaro, D. C.; Cerdá, J. Dynamic Scheduling of Multiproduct Pipelines with Multiple Delivery Due Dates. *Comput. Chem. Eng.* **2008**, *32*, 728–753.

(10) Relvas, S.; Matos, H. A.; Barbosa-Póvoa, A. P. F. D.; Fialho, J.; Pinheiro, A. S. Pipeline Scheduling and Inventory Management of a Multiproduct Distribution Oil System. *Ind. Eng. Chem. Res.* **2006**, *45*, 7841–7855.

(11) Boschetto, S. N.; Felizari, L. C.; Yamamoto, L.; Magatão, L.; Stebel, S. L.; Neves, F., Jr.; Arruda, L. V. R.; Lüders, R.; Ribas, P. C.; Bernardo, L. F. J. An Integrated Framework for Operational Scheduling of a Real-World Pipeline Network. *Comput.-Aided Chem. Eng.* **2008**, *25*, 259–264.

(12) Jittamai, P. Analysis of Oil-Pipeline Distribution of Multiple Products Subject to Delivery Time-Windows. Ph.D. Dissertation. Department of Industrial Engineering, Texas A&M University, College Station: Texas, 2004.

(13) Brooke, A.; Kendrick, D.; Meeraus, A.; Raman, R. *GAMS—A User's Guide*. GAMS Development Corporation: Washington, DC, 2006.

*Received for review* January 5, 2009
*Revised manuscript received* April 28, 2009
*Accepted* May 20, 2009

IE900015B