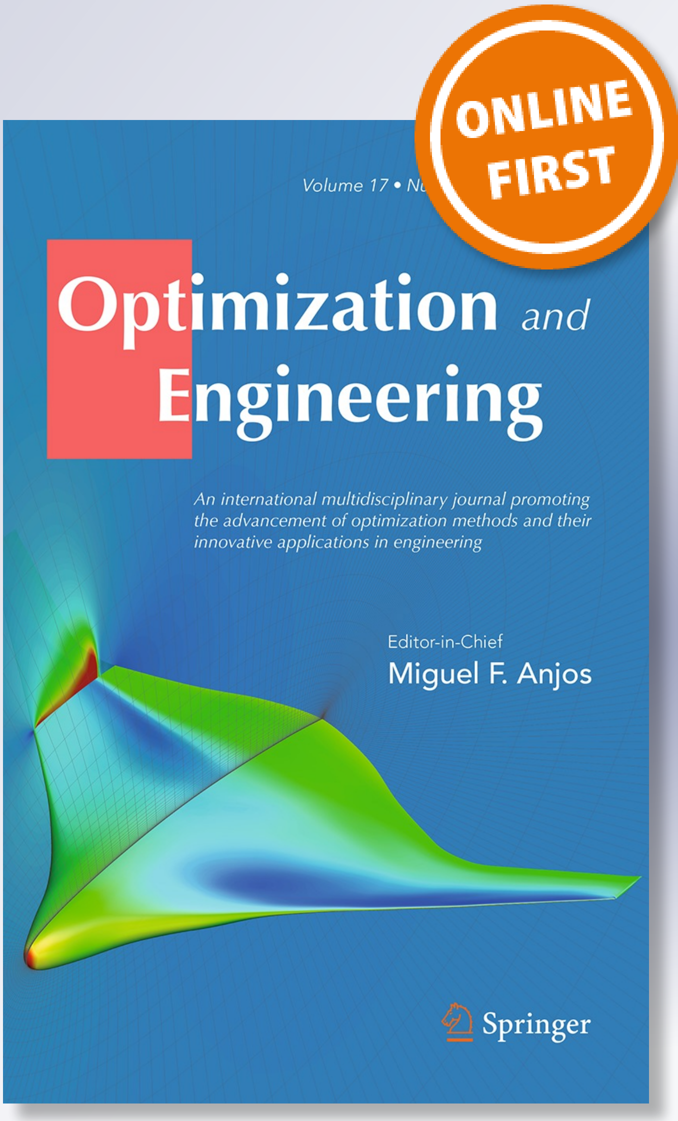
Short-term operational planning of refined products pipelines

Diego C. Cafaro & Jaime Cerdá

Optimization and Engineering
International Multidisciplinary Journal to Promote Optimization Theory & Applications in Engineering Sciences

ISSN 1389-4420

Optim Eng
DOI 10.1007/s11081-016-9330-5



Optim Eng
DOI 10.1007/s11081-016-9330-5

Short-term operational planning of refined products pipelines

Diego C. Cafaro[1] · Jaime Cerdá[1]

Received: 29 April 2015 / Revised: 6 March 2016 / Accepted: 25 June 2016

Abstract Pipelines are the safest and least expensive mode for transporting energy products over long distances. Refined products pipelines convey multiple oil derivatives from refineries to marketing terminals, usually through the same duct. Planning the injection, transportation and delivery of batches moving into pipelines is a very complex industrial problem with many operational constraints. This work synthesizes two innovative optimization tools for the short-term planning of oil product pipelines. The first one is a continuous-time mixed-integer linear programming (MILP) formulation for the short-term planning of pipelines connecting a single source node to multiple terminals over a multiperiod horizon. In the second approach, the MILP formulation is extended to deal with the transportation planning of multi-source pipelines. Common-carrier pipelines often present input facilities at non-origin points, whose operation raises new difficulties. Solutions to real-world case studies illustrate the performance of the proposed optimization tools.

Keywords Oil products pipelines · Operational planning · Optimization · MILP models

1 Introduction

Pipelines are the safest and least expensive way to deliver large quantities of energy products from refineries to distribution terminals. Nearly 70 % of the intercity ton-miles of crude oil and refined products in the US are moved by pipelines (Rabinow 2004). The transportation of refined petroleum products generally combines a long-haul delivery by pipeline from refineries to distribution terminals followed by a

✉ Diego C. Cafaro
dcafaro@fiq.unl.edu.ar

[1] INTEC (UNL - CONICET) - Facultad de Ingeniería Química (UNL), Güemes 3450, 3000 Santa Fe, Argentina

Published online: 08 August 2016

truck journey to local markets. A pipeline network can have several entry and exit points and the interchange of refined products between two common carrier pipelines may occur at shared terminals (Miesner and Leffler 2006). Refined petroleum products are inserted in the pipeline one after another often without any separation device. If two consecutive products are dissimilar, such as gasoline and jet fuel, a hybrid product called *transmix* is created at the interface. The transmix must be separated and stored before sending back to the refinery for reprocessing. Besides, products from different shippers meeting the same specifications can be unified and sent through the pipeline together as a single batch to several, distant receiving points. It is the so-called *fungible* operation mode. In contrast, some pipelines carry smaller lots, each one destined to a single customer, i.e. the *segregated* operation mode. The aim of pipeline logistics is to ensure that the right product will be received by the customer at the right time, at the right depot and at the lowest cost, under the best possible conditions of quality, safety and security.

Planning the injection of new batches in the pipeline and the simultaneous deliveries to depots is a very difficult task with many operational constraints to be satisfied. A good problem representation should be capable of tracking batches while flowing inside the pipeline. Hence, a key issue for efficient terminal operations is the coordination among incoming and outgoing product flows to/from every depot. Product stockouts at input terminals or overloading conditions at intermediate depots oblige the operator to temporarily stop the line. Another important feature of pipeline planning and scheduling problems is the fact that pumping operations must be periodically updated due to the new demand scenario. Late injections should not only be intended for pushing the current pipeline content but also matching product requirements due at future periods.

This work deals, on the one hand, with the short-term planning of a single unidirectional pipeline system involving a unique entry point at the origin and several distribution terminals along the line. The re-routing of shipments, order cancellations and the acceptance of new transportation orders force the scheduler to continuously update the plan. To cope with these issues, we present a dynamic planning tool based on a continuous-time mixed-integer linear programming (MILP) formulation. The approach considers a multi-period rolling horizon with several due dates taking place at period ends.

In common-carrier pipelines, however, several refineries located at different sites use the same trunk line for shipping refined petroleum products to downstream terminals. They are indeed multiple-source pipelines with input facilities at non-origin points. The operation of intermediate sources raises some new difficult issues. Pumping runs taking place at intermediate locations can either insert new lots or increase the size of batches in transit. Batches are no longer arranged along the line in the same order that they are injected, and tracking the batch sequence becomes a more complex task. To face this new challenge, the MILP formulation for single-source pipelines is then extended to deal with the transportation planning of multiple-source pipelines operating on fungible or segregated mode.

1.1 Literature review

Different types of approaches have been proposed to solve pipeline planning and scheduling problems, including rigorous optimization models, knowledge-based heuristic techniques (Hane and Ratliff 1995; Sasikumar et al. 1997), discrete-event simulation tools (Mori et al. 2007; García-Sánchez et al. 2008), and decomposition frameworks (Lopes et al. 2010). Rigorous optimization methods generally consist of solving a single MILP mathematical model and are usually grouped into two classes: discrete and continuous, depending on the way volume and time domains are handled. Discrete MILP-formulations divide both the pipeline volume into a significant number of single-product packs, and the planning horizon into time intervals of equal and fixed duration (Rejowski and Pinto 2003, 2004; Magatão et al. 2004; Zyngier and Kelly 2009). Since discrete models stand for approximate problem representations, they do not provide feasible schedules unless a dense discretization is adopted. Consequently, discrete formulations usually yield large-size models even for rather short time horizons.

In contrast, continuous MILP-formulations in both time and volume were first proposed by Cafaro and Cerdá (2004). Flow-rate variations due to changes in pipeline diameter are easily handled, and the optimal set of pumping and delivery operations is established all at once, in a rigorous manner. In addition, the model is able to track the location and size of product lots over the planning horizon, maintain product inventory levels in refinery and depot tanks within allowable ranges and account for high-energy cost intervals. Another MILP continuous representation is due to Relvas et al. (2006) who studied the scheduling of a single pipeline transporting a variety of oil derivatives from one refinery to a unique distribution center fulfilling daily customer demands over a monthly horizon. However, the computational performance of the approach badly deteriorates when the complete sequence of batch injections is to be selected. A similar pipeline scheduling problem was studied by Cafaro and Cerdá (2008a) who developed a smaller MILP formulation providing better schedules by optimizing the complete sequence of pumping runs in a much shorter computational time.

Recently, more realistic approaches tackling real-world pipeline networks with multiple origins and destinations have been developed. Despite their complex topology, simple principles are usually applied to scheduling pipeline networks. Mori et al. (2007) developed a discrete event simulation model for a detailed study of planned operational activities in real-world pipeline networks. The proposed simulator was used in combination with a short-term optimization package providing the pipeline schedule to be tested. Boschetto et al. (2010) presented a hybrid approach consisting of a decomposition strategy involving three blocks: (i) a resource-allocation block determining candidate product sequences, (ii) a pre-analysis block specifying the precise volumes to be moved, and (iii) a rather simple MILP model determining the exact timing of pumping and delivery operations. Another hybrid approach that combines a randomized constructive heuristic with novel constraint programming (CP) models was reported by Moura et al. (2008). In all the cases, most of the computational burden comes from three difficult tasks: pumping sequencing, batch sizing, and batch allocation to receiving terminals. By heuristically choosing them, the remaining operational decisions can be taken in a short CPU time. However, the final pipeline schedule is greatly influenced by those heuristic-based decisions previously taken (Boschetto et al. 2010).

1.2 Continuous models

To reduce the computational burden and avoid inaccuracies of discrete representations, we present rigorous formulations for the optimal planning of single-source and multiple-source pipelines, using continuous scales, both in time and volume. Figure 1 synthesizes the major elements of continuous models: pipeline segments, sources, destinations, batches, and pumping runs. The rest of the paper is organized as follows. In Sect. 2 we present a continuous-time model to capture single-source pipeline operations. In Sect. 3 we describe an algorithmic approach to dynamically update the transportation plan. In Sect. 4 we extend the single-source model to account for pipeline networks with multiple sources and destinations. In Sect. 5 we present and discuss the results of real-world case studies involving both single-source and multiple-source pipelines. We finally conclude the paper in Sect. 6.

2 Single-source pipeline model

2.1 Model assumptions

We first study a single unidirectional pipeline transporting a number of refined petroleum products from an input terminal to several downstream depots. We assume that pipelines remain completely full of products at any time. By assuming liquid incompressibility, the only way to get a volume of product out of the line at a downstream terminal is by injecting an equal volume at the input station. As a new product batch is injected, a portion of a batch flowing through the pipeline can be diverted to one of the assigned terminals while the remainder will continue moving to more distant points, i.e. the so-called batch “stripping” operation. In fact, a product request at some distribution terminal can be satisfied by diverting material from more than one batch.

Another important assumption is that product batches are pumped into the pipeline at turbulent flow to retard mixing. The transmix between a particular pair of refined products is supposed to be a known constant, independent of the scheduled batch movements. The transmix is kept into the line until it reaches the farthest

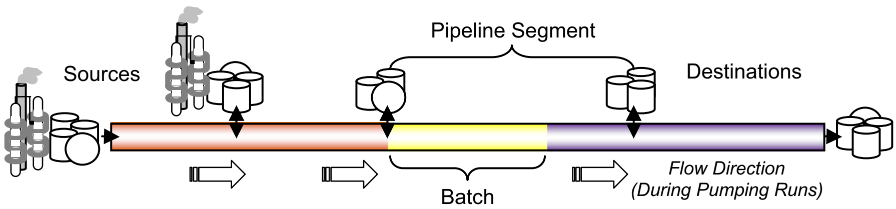


Fig. 1 Overall structure of continuous models for planning pipeline operations

terminal where it is stored and rerouted to the refineries. Besides, the unit pumping cost is a known constant that varies with the product and the final destination, but is independent of the pump rate. Finally, a non-cycling transportation planning strategy is applied. Therefore, the sequence of product shipments to be executed by the operator may change from one to the next time horizon. Since it may take over one or two weeks to move a batch to the assigned terminal, the horizon length must exceed such a large delivery lead-time. Otherwise, batches will be pumped into the pipeline without knowing their exact destinations.

2.2 Model variables and constraints

The mathematical formulation for the short-term planning of multiproduct pipelines with a single source is defined in terms of four major sets: (a) the old and new batches ($i \in I = I^{old} \cup I^{new}$), (b) the pipeline distribution terminals ($j \in J$), (c) the refined petroleum products to be transported ($p \in P$) from the refinery to terminals along the line, and (d) the time periods taking part of the multiperiod horizon ($t \in T$). Old batches $i \in I^{old}$ are those already in transit along the line at the start of the planning horizon, while new fungible batches $i \in I^{new}$ are those to be pumped in the pipeline during future periods. For every element $i \in I^{new}$, the product assigned and its size are model decisions to be optimally taken. Moreover, the problem formulation will assume that the set I has been chronologically arranged, with the old batches $i \in I^{old}$ preceding the new batches $i \in I^{new}$. Since the number of pumping runs to be executed throughout the time horizon is unknown beforehand, some of the later entries of I^{new} may stand for fictitious batches never executed. Some criteria for choosing the number of elements in I^{new} are given in the following sections.

2.2.1 *Batch features*

A new batch $i \in I^{new}$ that is planned to be pumped into the pipeline is characterized by the following properties:

(a) Allocated product (binary $y_{i,p}$).
(b) Initial batch size (Q_i).
(c) Initial injection time (S_i).
(d) Final injection time (C_i).
(e) Pumping run duration (L_i).
(f) Completion time period (binary $w_{i,t}$), i.e. the period at which the pumping of batch i ends.

They can be regarded as static properties since their values do not change with the pipeline activity, i.e. with the injection of new batches. By definition, batch $(i-1)$ precedes batch i. Hence, the interface volume between consecutive batches can be easily determined from the value of the variables $y_{i,p}$ and $y_{i-1,p'}$. The assignment variable $y_{i,p}$ indicates that the new batch $i \in I^{new}$ contains product p whenever $y_{i,p} = 1$. Furthermore, the binary variable $w_{i,t}$ is an assignment variable indicating that the pumping of the new batch $i \in I^{new}$ is completed in period t whenever $w_{i,t} = 1$. Nonetheless, the pumping run may have begun at an earlier period $t' < t$. Note that the model can plan interruptions in the pipeline flow by simply making $S_i > C_{i-1}$. In fact, the duration of the flow interruption is $S_i - C_{i-1}$.

2.2.2 Batch tracking and delivery operations

Some other batch properties are pipeline activity-dependent and their values may change whenever a new batch is pumped into the line. They will be referred to as batch dynamic properties. The pumping completion times stand for the major event points at which the dynamic batch properties are to be determined. To know when a batch will arrive to a stated destination and what amount of product is to be diverted, the batch movement along the pipeline should be followed. The dynamic properties of batch i are monitored with time through the following variables:

(g) Frontal volume coordinate of batch $i \in I$ at time $C_{i'}$ ($F_i^{(i')}$).
(h) Batch size at time $C_{i'}$ ($W_i^{(i')}$).
(i) Amount of material diverted from batch i to depot j during run i' ($D_{i,j}^{(i')}$).
(j) Delivery of product from batch i to depot j during run i' (binary $x_{i,j}^{(i')}$).

Through the set of binary variables $x_{i,j}^{(i')}$ the model evaluates whether diverting batch $i \in I$ to depot j while pumping a new batch $i' \in I^{new}$ ($i' \geq i$) is or is not a feasible action. It will be feasible only if batch i has arrived at (but not surpassed) depot j before or during the time interval $[S_{i'}, C_{i'}]$ and, consequently, $x_{i,j}^{(i')}$ can take value 1.

2.2.3 Monitoring stocks and product deliveries from depots to local markets

The model monitors depot inventory levels to prevent from: (a) batch stripping operations causing tank overloading, and (b) product shipments from depots to neighboring markets that cannot be afforded due to lack of stock. Tracking product inventories over time requires to determine product availabilities at the time points C_i. Moreover, product deliveries to local markets must be scheduled in such a way that the specified demands at the end of each period t are timely satisfied to minimize backorder costs. Such constraints involve the following additional variables:

(k) Inventory level of product p in depot j at time point C_i ($ID_{p,j}^{(i')}$).
(l) Amount of product p diverted to depot j during run i' ($DP_{p,j}^{(i')}$).
(m) Supply of product p from depot j to local markets over the time interval $[C_{i'-1}, C_{i'}]$ ($DM_{p,j}^{(i')}$).
(n) Backorder of product p requested to depot j in time period t ($B_{p,j,t}$).

The overall amount of product p sent from depot j to local markets up to the end of time period t can be computed in terms of the variables $DM_{p,j}^{(i')}$. In turn, the continuous variable $B_{p,j,t}$ represents the demand of product p requested to depot j that remains unsatisfied at the end of period t.

2.3 Mathematical formulation for the short-term planning of single-source multiproduct pipelines

2.3.1 Batch-defining constraints

Product allocation

$$\sum_{p \in P} y_{i,p} \leq 1 \quad \forall i \in I^{new} \tag{1}$$

Batch sequencing

$$L_i \leq C_i \leq h_{\max} \quad \forall i \in I^{new} \tag{2}$$

$$S_i = C_i - L_i \geq C_{i-1} + \tau_{p,p'} \left(y_{i-1,p'} + y_{i,p} - 1 \right) \quad \forall i \in I^{new}; p, p' \in P \tag{3}$$

$\tau_{p,p'}$ is the changeover time between products p and p' when they are successively injected, while $h_{\max}$ is the planning horizon length.

Initial batch size and pumping run duration

$$vb_{\min} L_i \leq Q_i \leq vb_{\max} L_i \quad \forall i \in I^{new} \tag{4}$$

$$\sum_{p \in P} y_{i,p}\, l_{\min,p} \leq L_i \leq \sum_{p \in P} y_{i,p}\, l_{\max,p} \quad \forall i \in I^{new} \tag{5}$$

$$\sum_{p \in P} y_{i,p} \leq \sum_{p \in P} y_{i-1,p} \quad \forall i \in I^{new} \tag{6}$$

$vb_{\min}$ and $vb_{\max}$ are the minimum and maximum pump rates, while $l_{\min,p}$ and $l_{\max,p}$ are the min/max lengths for injections of product p.

Interface volume between consecutive batches

$$WIF_{i,p,p'} \geq ifase_{p,p'} \left(y_{i-1,p'} + y_{i,p} - 1 \right) \quad \forall i \in I, i > 1\ p, p' \in P \tag{7}$$

$ifase_{p,p'}$ is the size of the interface between products p and p'.

Forbidden product sequences p-p''

$$y_{i-1,p} + y_{i,p''} \leq 1 \quad \forall i \in I^{new} \tag{8}$$

Completion time period

$$\sum_{t \in T} w_{i,t} = \sum_{p \in P} y_{i,p} \quad \forall i \in I^{new} \tag{9}$$

If dd_t stands for the end time of period t, we have:

$$C_i \geq dd_{t-1}\, w_{i,t} \tag{10}$$

$$C_i \leq dd_t + (1 - w_{i,t})\,(h_{\max} - dd_t) \qquad \forall i \in I^{new},\ t \in T \tag{11}$$

2.3.2 Batch-tracking constraints

Pipeline coordinates of batch $i \in I$ at time point $C_{i'}$ (see Fig. 2)

$$F_{i+1}^{(i')} + W_i^{(i')} = F_i^{(i')} \qquad \forall i \in I, \forall i' \in I^{new},\ i' \geq i \tag{12}$$

Material diverted from a new batch $i \in I^{new}$ while being injected

$$Q_i = W_i^{(i)} + \sum_{j \in J} D_{i,j}^{(i)}; \quad F_i^{(i)} - W_i^{(i)} = 0 \qquad \forall i \in I^{new} \tag{13}$$

Material diverted from batch i while pumping a later batch $i' \in I^{new}$

$$W_i^{(i')} = W_i^{(i'-1)} - \sum_{j \in J} D_{i,j}^{(i')} \quad \forall i \in I, \forall i' \in I^{new}, i' > i \tag{14}$$

Feasibility conditions for diverting material to depots

$$d_{\min}\, x_{i,j}^{(i')} \leq D_{i,j}^{(i')} \leq d_{\max}\, x_{i,j}^{(i')} \quad \forall i \in I, \forall i' \in I^{new}, i' \geq i, \forall j \in J \tag{15}$$

$d_{\max}$ is an upper bound on the amount of material that can be transferred from a batch into the line to any depot during a batch injection, and $d_{\min}$ is a small positive value standing for the minimum amount to be diverted only if $x_{i,j}^{(i')} = 1$.

$$\begin{aligned} F_i^{(i')} - \sum_{p \in P} \sum_{p' \neq p} WIF_{i,p,p'} \ \geq\ \sigma_j\, x_{i,j}^{(i')} \quad & \forall i \in I, \forall i' \in I^{new}, i' \geq i, \forall j < |J| \\ F_i^{(i')} \ \geq\ \sigma_j\, x_{i,j}^{(i')} \quad & \forall i \in I, \forall i' \in I^{new}, i' \geq i, j = |J| \end{aligned} \tag{16}$$

σ_j is the volumetric coordinate of terminal j (measured from the pipeline origin).

Bound on the amount of material diverted from batch i to depots

$$\begin{aligned} \sum_{j < |J|} D_{i,j}^{(i')} \leq W_i^{(i'-1)} - \sum_{p \in P} \sum_{p' \neq p} WIF_{i,p,p'} \quad & \forall i \in I, \forall i' \in I^{new}, i' > i \\ \sum_{j \in J} D_{i,j}^{(i')} \leq W_i^{(i'-1)} \quad & \forall i \in I, \forall i' \in I^{new}, i' > i \end{aligned} \tag{17}$$

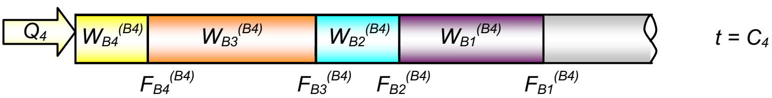


Fig. 2 Positioning of batches in the pipeline

Overall volume balance during the pumping of batch $i' \in I^{new}$

$$\sum_{i \le i'} \sum_{j \in J} D_{i,j}^{(i')} = Q_{i'} \quad \forall i' \in I^{new} \tag{18}$$

Figure 3 describes the pipeline status before and after a single injection. The volumetric balance between input and output flows should be respected. Systematic procedures for generating more detailed schedules are presented by Cafaro et al. (2010, 2011).

2.3.3 Depot inventory management constraints

Product deliveries from distribution terminals to markets

$$DM_{p,j}^{(i)} \le (C_i - C_{i-1})\, vm_{p,j} \quad \forall p \in P, \forall j \in J_p, \forall i \in I^{new} \tag{19}$$

$vm_{p,j}$ stands for the maximum rate for delivering product p from the pipeline terminal j to the markets.

Delivery time requirements

$$\sum_{i \in I^{new}} w_{i,t} \ge 1 \qquad \forall t \in T \tag{20}$$

$$\sum_{\ell \in I^{new} | \ell \le i} DM_{p,j}^{(\ell)} \ge \sum_{k \le t} dem_{p,j,k}\,(w_{i,t} - w_{i+1,t}) - B_{p,j,t} + B_{p,j,(t-1)} \qquad \forall p \in P, j \in J_p, t \in T, i \in I^{new} \tag{21}$$

The parameter $dem_{p,j,k}$ denotes the demand of product p with due date dd_k at terminal j.

Monitoring product inventories in depot tanks

$$DP_{i,p,j}^{(i')} \le d_{\max}\, y_{i,p} \quad \forall i \in I, p \in P, j \in J_p, i' \in I^{new} \tag{22}$$

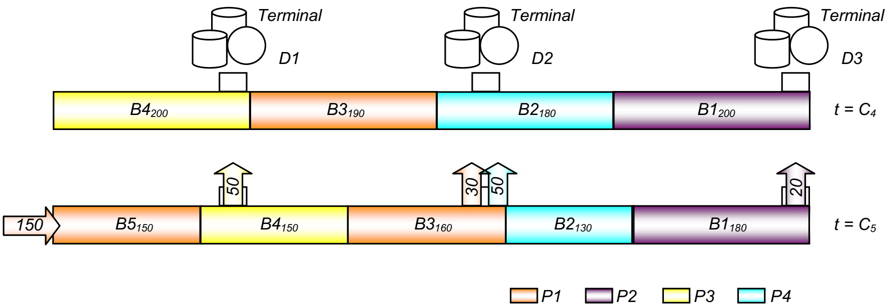


Fig. 3 Overall volume balance while pumping batch *B5*

$$\sum_{p \in P} DP_{i,p,j}^{(i')} = D_{i,j}^{(i')} \qquad \forall i \in I, j \in J_p, i' \in I^{new} \tag{23}$$

$$DP_{i,p,j}^{(i')} = D_{i,j}^{(i')} \quad \forall i \in I^{old}, p = P_i, j \in J_p, i' \in I^{new} \tag{24}$$

where P_i stands for the product contained in the existing lot i.

Inventory feasible range at the depots

$$ID_{p,j}^{(i')} = ID_{p,j}^{(i'-1)} + \sum_{\substack{i \in I \\ i \leq i'}} DP_{i,p,j}^{(i')} - DM_{p,j}^{(i')} \quad \forall p \in P, j \in J_p, \quad i' \in I^{new} \tag{25}$$

$$(id_{\min})_{p,j} \leq ID_{p,j}^{(i')} \leq (id_{\max})_{p,j} \quad \forall p \in P, j \in J_p, i' \in I^{new} \tag{26}$$

Similar constraints can be used to monitor the product inventories in refinery tanks.

2.3.4 Initial conditions

$$F_i^{(i'-1)} = f_i^o;\; W_i^{(i'-1)} = w_i^o \quad \forall i \in I^{old},\; i' = first(I^{new}) \tag{27}$$

2.3.5 Problem objective function

The problem goal is to minimize the total pipeline operating cost including (i) the pumping cost, (ii) the reprocessing cost of the interface material between consecutive batches, (iii) the cost of product backorders being tardily delivered to their destinations, and (iv) the cost of holding product inventory.

$$\begin{aligned} Min\ z = {} & \sum_{p \in P} \sum_{j \in J} \left(cp_{p,j} \sum_{i \in I} \sum_{i' \in I^{new}} DP_{p,i,j}^{(i')} \right) + \sum_{\substack{p' \in P \\ p' \neq p}} \sum_{\substack{i \in I \\ i > 1}} cf_{p,p'}\ WIF_{i,p,p'} \\ & + \sum_{p \in P} \sum_{j \in J} \sum_{t \in T} cb_{p,j,t}\, B_{p,j,t} + \frac{h_{\max}}{|I^{new}|} \sum_{p \in P} \left[\sum_{j \in Jp} cid_{p,j} \left(\sum_{i' \in I^{new}} ID_{p,j}^{(i')} \right) + cir_p \left(\sum_{i' \in I^{new}} IRS_p^{(i')} \right) \right] \end{aligned} \tag{28}$$

The parameter $cp_{p,j}$ stands for the cost of pumping a unit of product p from the oil refinery to destination j, whereas $cf_{p,p'}$ is the cost for reprocessing a unit amount of interface p–p'. In turn, the parameter $cb_{p,j,t}$ is the unit backorder penalty cost. The last term provides an approximate value for the inventory carrying cost at distribution centers and refinery tanks.

3 Updating the transportation plan of single-source pipelines

There are two major reasons for a periodical review of the pipeline transportation plan:

1. New shipper nominations are received during the current period. Such nominations need to be delivered to the stated terminals at later periods of the present planning horizon, and therefore shall be pumped into the pipeline with some anticipation.
2. There is a significant transportation lead-time, especially for shipments destined to farthest distribution terminals. As a result, some batches scheduled for pumping at later periods of the current horizon have the only purpose of pushing the batches already into the pipeline towards their stated destinations. Since they are required to meet yet unknown product demands due at time periods beyond the current horizon, the material pumped into the pipeline may have nothing to do with future terminal requirements. Generally, long pumping runs are last scheduled. As the time horizon rolls, large batches are gradually replaced by a sequence of shorter pumping runs through the periodic rescheduling process. Such smaller batches are mostly aimed at fulfilling new shipper requests due at later periods of the new time horizon.

The algorithm for the periodic update of the pipeline operations schedule comprises five major stages: (a) initialization, (b) problem data update, (c) pipeline transportation plan update, (d) batch dispatching, and (e) horizon rolling and new instance generation.

3.1 Initialization stage

During the initialization stage, some model parameters are set by the scheduler. They include:

(A) The number of time periods (N) into which the rolling horizon T is divided, and the length h_t (=$dd_t - dd_{t-1}$) of every time period t, expressed in hours. In our case studies, $N = 4$ and $h = 168$ h (1 week) for every period t, resembling the typical cyclic horizon used in real world trunk lines, divided into periods of 5 to 10 days (Explorer pipeline 2016). Delivery requests are due at the end of every period. Therefore, the due-dates over the initial horizon are: $\{dd_1 = 168, dd_2 = 336, dd_3 = 504, dd_4 = 672\}$. The value of $N \cdot h$ should never be smaller than the time it takes to move a batch from the pipeline origin to the farthest terminal.

(B) The number of different oil products to be shipped from the refinery to the stated destinations, i.e. $|P|$.

(C) The number of new batches $i \in I^{new}$ to be pumped into the pipeline during the multiperiod time horizon, i.e. the cardinality of the set I^{new}. The value of $|I^{new}|$ is usually set equal to: $|I^{new}| = (N * |P|)/n$, where $n = 2.0$–3.5. If the adopted value for $|I^{new}|$ is not large enough, the problem feasible region may not include the true optimal solution or, at worst, may be empty. Whenever the number of non-fictitious pumping runs at the optimum is equal to $|I^{new}|$ or the problem is infeasible, the value of $|I^{new}|$ must be increased by one. After that, the model is to be solved again until no improvement in the value of the objective function is achieved.

(D) The permissible ranges for product inventories at refinery and depot tanks ($ir_{\min}/ir_{\max}$, $id_{\min}/id_{\max}$), the pipeline pump rate range [$vb_{\min}$; $vb_{\max}$] and delivery rates ($vm_{p,j}$).

(E) The different types of pipeline unit costs arising in the objective function as well as the product–product interface size matrix.

(F) The time interval between two consecutive reviews of the pipeline schedule (t_{RS}). This schedule regeneration frequency is expressed in time periods.

(G) The subset of hard-frozen time periods $T_{HF} \subset T$, usually including the first-period of the new rolling horizon, where the pipeline operations must remain unchanged even during the periodic rescheduling process. In practice, the regeneration frequency is generally equal to the number of hard frozen periods ($t_{RS} = |T_{HF}|$). Typically, $t_{RS} = |T_{HF}| = 1$, and the pipeline rescheduling process is executed at the start of every time period.

(H) The subset of soft-frozen time periods $T_{SF} \subset T$, usually including one or two periods immediately after the first one, over which the sequence of product injections cannot be modified. However, their pumping run lengths may change.

(I) The first period of the current moving horizon (period k). The action period k will be used to identify the corresponding instance of the moving horizon as it rolls over time. Set $k = 1$ for the initial horizon.

3.2 Data updating stage

When the rescheduling process is triggered, or the pipeline schedule for the initial horizon is to be generated, the next step is to update the input data for the current horizon. This stage involves the following steps:

(A) Capture the pipeline current status from the SCADA remote system to establish the sequence of batches in transit (I^{old}), i.e. batch naming (i), product (p_i), size ($w^{\circ}{}_i$) and location ($f^{\circ}{}_i$). The SCADA remote system is usually available in every multiproduct pipeline network.

(B) Pick up product inventory levels at refinery and terminal tank farms ($ir^{\circ}{}_p$, $id^{\circ}{}_{p,j}$) at the start of the current horizon k from the SCADA system.

(C) Import the updated refinery production schedule for periods k to $k + N - 1$, i.e. from $t = dd_{k-1}$ to $t = dd_{k-1} + h_{\max}$, where $h_{\max}$ is the constant length of the rolling horizon. In most cases, the refinery production schedule is previously defined based on crude oil inventories, product expected demands at distribution terminals and available production capacity.

(D) Update product demands at distribution terminals, including old demands not yet satisfied and new shipments received while executing the pipeline schedule for the action period of the previous horizon. To update terminal demands $dem_{p,j,t}$, it must be taken into account: (1) product deliveries to terminals accomplished during period ($k - 1$), in advance of the promised time period $t > k - 1$; and (2) product deliveries due at time $dd_{k\text{-}1}$ that were not satisfied during period $k - 1$ and must be fulfilled as backorders in the next action period k.

3.3 Pipeline rescheduling stage

This stage is the core step of the algorithm. It provides the pipeline master planning over the current rolling horizon k by running the optimization model. Its major goal is to generate the pipeline input and output plans based on the updated information. Just the proposed plan for the first period k is subsequently implemented while the pipeline planning for later periods helps schedulers achieve a better coordination of the entire supply system.

3.4 Detailed scheduling stage

The next step aims to generate the detailed pipeline schedule for the action period k based on the pipeline master planning found in Step 3.3. In particular, the detailed scheduling stage should account for the set of batch injections and batch stripping operations to be carried out from time dd_{k-1} to dd_k. Compared with the pipeline master plan for period k, some additional information is provided by the detailed schedule. For instance, the sequence and timing of the planned delivery operations to be performed during the execution of any pumping run scheduled for period k. The pipeline master planning guarantees the existence of, at least, a feasible sequence of stripping operations for each planned batch injection. Since there are usually several alternative operational schemes, some additional criteria for choosing one of them are to be considered. Algorithmic and heuristic procedures for developing the pipeline schedule at the operational level for the action period k are discussed in Cafaro et al. (2010). In this paper, we are focused on the pipeline master planning. The last planned pumping run i_k to be executed in period k is considered up to time dd_k though it can be extended over period $k + 1$. If the run i_k goes beyond period k in the pipeline master plan, some product deliveries from the line may be decreased or postponed for the next period $k + 1$.

3.5 Horizon rolling and new instance generation

Once the pipeline schedule for the first t_{RS} periods has already been executed (i.e., at time $t = dd_{k-1} + h\, t_{RS}$), the time horizon rolls ahead t_{RS} periods. If $t_{RS} = 1$, the new action period will be $k := k + 1$, and the new instance of the moving horizon is thus generated. To update the pipeline master plan for the new horizon, the rescheduling process is activated, and the execution of Stages 3.2–3.4 is triggered again.

3.6 Comparison with discrete representations

The continuous-time optimization framework presented in this section comprises an iterative procedure to determine the cardinality of the set of batches in the optimal solution, at each rescheduling instance. This could be computationally expensive, but if the initial guess on that number is accurate, no more than two iterations are usually needed. In contrast, discrete formulations would only need to be solved once at each instance. Moreover, discrete models could find the transportation plan and the detailed schedule in a single step. However, due to their large size, discrete formulations usually fail to solve pipeline scheduling problems involving time horizons of more than 5 days (Rejowski and Pinto 2008). Both continuous and discrete models are certainly favored by fixing a subset of decisions, like the ones involved in hard-frozen and soft-frozen periods of the time horizon. A detailed comparison of the performance of both models in dynamic planning environments is left as future work.

4 Multiple-source pipeline model

4.1 Model assumptions

A1. A multiple-source pipeline with unidirectional flow is studied.
A2. Non-interacting batch injections from different pumping terminals can be simultaneously performed.
A3. If individual shipments of the same grade or product from different sources meet common specifications, they can be mixed into a fungible batch.
A4. During a pumping run, the pipeline can receive material from either an adjacent segment or the tank farm, but not both.
A5. The injection rate may change with the source within the allowable range.

4.2 Model variables and constraints

The multiple-source pipeline transportation planning problem involves five major sets: blocks of 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). The problem includes two additional sets with regards to the single-source case, i.e. K and S. The elements of K represent blocks of parallel pumping runs carried out simultaneously at different sources. Parallel runs may not necessarily start or finish at the same time. However, batch injections in block k can start only if the previous block $(k - 1)$ has ended. The length of a block k is the time elapsed from the earliest start to the latest completion of pumping runs belonging to k. Since the required numbers of pumping runs and batches are not precisely known before solving the problem, the values of $|K|$ and $|I|$ should be arbitrarily adopted. A simple expression for the estimation of $|K|$ is given in the following section.

4.2.1 Blocks of pumping runs (K)

Let us define the set $K = \{k_1, k_2, k_3,\ldots, k_m\}$ with the elements $k_1, k_2,\ldots$ chronologically ordered. The cardinality of the set K can be initially set to:

$$|K| = \frac{2}{|S| + 1} \sum_{p \in P} \left(\frac{2}{Q_{\min,p} + Q_{\max,p}} \sum_{j \in J} dem_{p,j} \right)$$

where $Q_{\min,p}$ and $Q_{\max,p}$ represent the minimum/maximum batch sizes for product p, and $dem_{p,j}$ stands for the delivery request of product p at the output terminal j. Three properties characterize every batch injection: (a) the block $k \in K$ to which it belongs, (b) the input node $s \in S$ where it takes place, and (c) either the new batch or the additional portion of a batch $i \in I$ pumped into the line. If lot i is enlarged 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 to divert some amounts of products to output terminals.

4.2.2 Set of batches (I)

By definition, the set I is given by: $I = \{i_1, i_2, i_3,\ldots,i_n\}$ with the elements $i_1, i_2,\ldots$ arranged in the same order that they are sequenced into the pipeline. The separate handling of pumping runs and batches often leads to some reduction in the cardinality of the set I. A good initial choice for the number of new batches ($|I^{new}|$) is $\alpha\ |K|$, with $\alpha = 0.9$ when the pipeline is operated in segregated mode (distinct products for different destinations), and $\alpha = 0.6$ for fungible mode (common products for several destinations). In the optimal single-source pipeline schedule, it is sometimes observed that the pipeline is stopped after injecting batch $(i - 1)$ containing product p, and the following batch i, also transporting product p, is inserted after the idle period. In the multi-source problem formulation, such a batch i will be regarded as an additional portion of batch $(i - 1)$. In this way, some saving in the number of batches is achieved. The major features of a batch are: (a) the product p it contains, (b) the source s from which it is inserted, and (c) the output terminals to which the batch is destined. If a new batch i is shipped through a pumping run k performed at some downstream location s, batch i 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 k (see Fig. 4).

4.2.3 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 input nodes are: (a) the set of products that can be injected, (b) the available product inventories that may change with time through the refinery production runs and the injection of batches into the pipeline, and (c) the terminal volumetric coordinate. Data related to output terminals are: (a′) the set of products that are demanded over the planning horizon, (b′) the initial product stocks in terminal tanks, (c′) the product demands to be satisfied before the end of the current horizon, and (d′) the terminal volumetric coordinate. The overall pipeline volume is denoted by *pv*.

4.2.4 Set of products

The set P comprises all 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 $j \in J$, and P_s denotes the subset of products that can be pumped into the line from the input terminal s.

4.3 Mathematical formulation for the operational planning of multisource, multiproduct pipelines

4.3.1 Pumping run constraints

Sequencing blocks of parallel pumping runs

$$C_k - L_k \geq C_{k-1} \qquad \forall k \in K\ (k > 1) \tag{29}$$

$$C_k \leq h_{\max} \quad \forall k \in K \tag{30}$$

Allocating batches to individual pumping runs

$$\sum_{i \in I} v_{i,s}^{(k)} \leq 1 \qquad \forall k \in K,\ s \in S \tag{31}$$

$$\sum_{s \in S} \sum_{i \in I} v_{i,s}^{(k)} \leq |S| \left(\sum_{s \in S} \sum_{i \in I} v_{i,s}^{(k-1)} \right) \qquad \forall k \in K\ (k > 1) \tag{32}$$

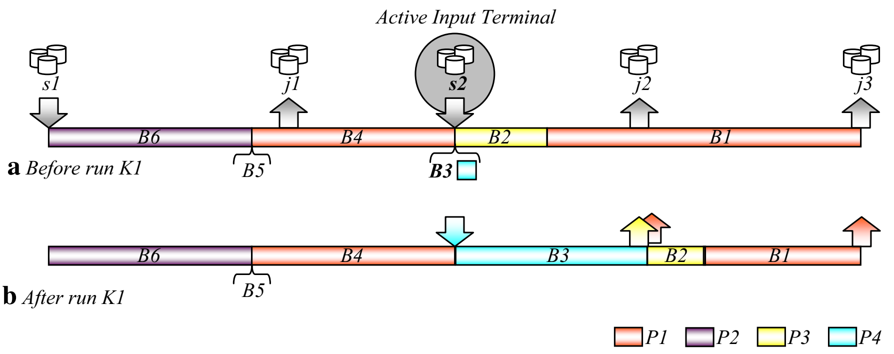


Fig. 4 Product deliveries while injecting batch *B3* from source *s2*

Sizing batch injections

$$q_{\min,s}\, v_{i,s}^{(k)} \leq \; Q_{i,s}^{(k)} \leq \; q_{\max,s}\, v_{i,s}^{(k)} \quad \forall i \in I, s \in S, k \in K \tag{33}$$

where $q_{\min,s}$ and $q_{\max,s}$ stand for the minimum and maximum batch sizes that can be injected through a pumping run from the input terminal *s*.

Choosing lengths for blocks of pumping runs: Let $LL_{k,s}$ be the length of the pumping run (*k, s*). Then,

$$vb_{\min,s}\, LL_{k,s} \leq \sum_{i \in I} Q_{i,s}^{(k)} \leq vb_{\max,s}\, LL_{k,s} \quad \forall k \in K, s \in S \tag{34}$$

$$L_k \; \geq \; LL_{k,s} \qquad \forall k \in K, s \in S \tag{35}$$

The interval $[vb_{\min,s};\, vb_{\max,s}]$ represents the feasible pump rate range at source *s*.

4.3.2 Batch-tracking constraints

Tracking the batch size over time

$$W_{i,k} = W_{i,k-1} + \sum_{s \in S} Q_{i,s}^{(k)} - \sum_{j \in J} D_{i,j}^{(k)} \qquad \forall i \in I\,,\, k \in K \tag{36}$$

$$D_{i,j}^{(k)} \leq \sum_{s/\tau_s < \sigma_j} \sum_{i' \in I} Q_{i',s}^{(k)} \qquad \forall i \in I, j \in J, k \in K \tag{37}$$

τ_s is the volumetric coordinate of source *s*.

Tracking the batch location

$$F_{i,k} - W_{i,k} = F_{i+1,k} \quad \forall i \in I, k \in K \tag{38}$$

$$F_{i,k-1} \; \leq \; F_{i,k} \quad \forall i \in I, k \in K \tag{39}$$

$$F_{i,k} \leq pv; \;\; F_{i,k} - W_{i,k} \geq 0 \quad \forall i \in I, k \in K \tag{40}$$

4.3.3 Pipeline volumetric balance

$$\sum_{i \in I} W_{i,k} = pv \quad \forall k \in K \tag{41}$$

$$\sum_{i \in I} \sum_{s \in S} Q_{i,s}^{(k)} = \sum_{i \in I} \sum_{j \in J} D_{i,j}^{(k)} \quad \forall k \in K \tag{42}$$

4.3.4 Feasibility constraints for batch injections and deliveries

Supplying material from an input node to an existing batch

$$F_{i,k-1} \geq \tau_s \, v_{i,s}^{(k)} \qquad \forall i \in I, s \in S, k \in K \tag{43}$$

$$F_{i,k-1} - W_{i,k-1} \leq \tau_s + (pv - \tau_s)\,\left(1 - v_{i,s}^{(k)}\right) \quad \forall i \in I, s \in S, k \in K \tag{44}$$

Diverting material from in-transit batches to output terminals.

$$F_{i,k} \geq \sigma_j \, x_{i,j}^{(k)} \tag{45}$$

$$d_{\min} \, x_{i,j}^{(k)} \leq D_{i,j}^{(k)} \leq d_{\max} \, x_{i,j}^{(k)} \quad \forall i \in I, j \in J, k \in K \tag{46}$$

$$F_{i,k-1} - W_{i,k-1} \leq \sigma_j + (pv - \sigma_j)\,\left(1 - x_{i,j}^{(k)}\right) \quad \forall i \in I, j \in J, k \in K \tag{47}$$

$$\sum_{j'=1}^{j} D_{i,j'}^{(k)} \leq \sigma_j - (F_{i,k-1} - W_{i,k-1}) + \sum_{\substack{s \in S \\ \tau_s < \sigma_j}} Q_{i,s}^{(k)} + (pv - \sigma_j)\,\left(1 - x_{i,j}^{(k)}\right) \quad \forall i \in I, j \in J, k \in K \tag{48}$$

4.3.5 Non-interacting pumping run constraint

To avoid the generation of a pipeline schedule involving incompatible parallel runs as the ones shown in Fig. 5, the constraint (49) is included in the problem model.

$$\sum_{i \in I} \sum_{\substack{j \in J \\ \sigma_j \leq \tau_s}} D_{i,j}^{(k)} \geq \sum_{i \in I} \sum_{\substack{s' \in S \\ \tau_{s'} < \tau_s}} Q_{i,s'}^{(k)} - q_{\max} \left(1 - \sum_{i \in I} v_{i,s}^{(k)}\right) \quad \forall s \in S, k \in K \tag{49}$$

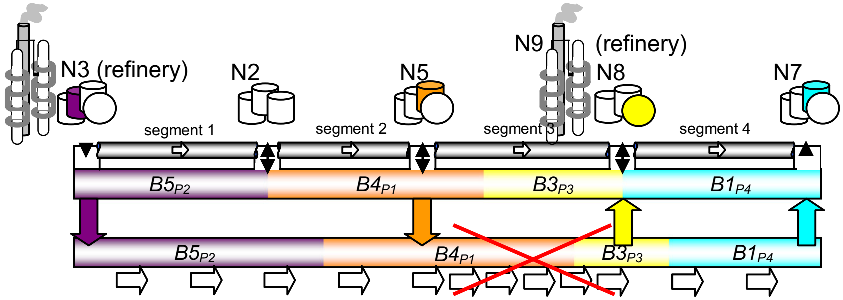


Fig. 5 Incompatible parallel batch injections

4.3.6 Product supply and demand constraints

Assigning products to batches

$$\sum_{p\in P} y_{i,p} \le \quad 1 \qquad \forall i \in I \tag{50}$$

$$\sum_{p\in P} y_{i,p} \le \sum_{s\in S}\sum_{k\in K} v_{i,s}^{(k)} \le \quad |S||K| \sum_{p\in P} y_{i,p} \qquad \forall i \in I^{new} \tag{51}$$

$$\sum_{p\in P} y_{i,p} \le \sum_{p\in P} y_{i-1,p} \quad \forall i \in I^{new}\ (i > 1) \tag{52}$$

If $(p,\ p')$ is forbidden then,

$$y_{i-1,p'} + y_{i,p} \ \le\ 1 \quad \forall i, i-1 \in I^{new} \tag{53}$$

Amount of product p pumped into the line through the run k

$$QP_{i,s,p}^{(k)} \ \le\ q_{\max,\mathrm{p}}\ y_{i,p} \qquad \forall i \in I, s \in S, k \in K, p \in P \tag{54}$$

$$\sum_{p\in P} QP_{i,s,p}^{(k)} \ =\ Q_{i,s}^{(k)} \qquad \forall i \in I,\ s \in S,\ k \in K \tag{55}$$

$q_{\max,p}$ stands for the maximum batch injection size for product p.
Volume of product p delivered from in-transit batches

$$DP_{i,j,p}^{(k)} \ \le\ d_{\max}\ y_{i,p} \qquad \forall i \in I, j \in J, k \in K, p \in P \tag{56}$$

$$\sum_{p\in P} DP_{i,j,p}^{(k)} = D_{i,j}^{(k)} \qquad \forall i \in I, j \in J, k \in K \tag{57}$$

Fulfilling product demands at every output terminal

$$dem_{p,j}\ - B_{p,j} \le \sum_{k\in K}\sum_{i\in I} DP_{i,j,p}^{(k)} \le du_{p,j} \quad \forall p \in P,\ j \in J \tag{58}$$

The model parameter $du_{p,j}$ as the maximum amount of product p that can be delivered and stored in depot j. When the demand of product p at depot j cannot be satisfied before the end of the planning period, a non-zero backorder $B_{p,j} > 0$ will arise. By including the variable $B_{p,j}$ in constraint (58), the pipeline scheduling problem remains feasible even if some demands are unsatisfied at the horizon end. Additional variables and constraints, similar to those presented in Sect. 2, can also be included to optimally plan charging and dispatching operations to/from destination tanks.

Feasible range for the amount of product p shipped from source s

$$sl_{p,s} \leq \sum_{k \in K} \sum_{i \in I} QP_{i,s,p}^{(k)} \leq su_{p,s} \qquad \forall p \in P, s \in S \tag{59}$$

$sl_{p,s}$ is usually large enough to fulfill, in combination with other sources, the specified demands of product p at distribution depots not covered by the initial linefill. Moreover, it may also include the "sweeping" lots pushing the overall pipeline content to the assigned depots. Such filler lots are selected to either getting a suitable final linefill to meet future product demands or providing free tank capacity to receive new production runs from source s. Alternatively, more variables and constraints similar to those presented in Sect. 2 could be included to make a rigorous tracking of product inventories in refineries and other sources over time.

4.3.7 Initial linefill

$$F_{i,k-1} = \sum_{i' \geq i} w_{i'}^{o} \qquad \forall i \in I^{old}, \quad k = 1 \tag{60}$$

4.3.8 Objective function

Two alternative problem goals have been chosen:

1. *The minimum makespan*, assuming a non-fixed horizon length and non-specified delivery due dates.

$$Min \quad z = H, \quad subject\ to \quad H \geq C_k \quad \forall k \in K \tag{61}$$

2. *The minimum total cost*, including transition, pumping and backorder costs. If inventory carrying costs are also to be minimized, more variables and constraints need to be included, as explained in Sect. 4.3.6.

$$Min \quad z = \sum_{i \in I} TC_i + \sum_{k \in K} PC_k + BC \tag{62}$$

where,

$$TC_i \geq cif_{p,p'} \left(y_{i,p} + y_{i+1,p'} - 1 \right) \qquad \forall i \in I; p, p' \in P \tag{63}$$

$$PC_k = \sum_{i \in I} \sum_{s \in S} \sum_{p \in P} cin_{p,s}\, QP_{i,s,p}^{(k)} \quad \forall k \in K \tag{64}$$

$$BC = \sum_{p \in P} \sum_{j \in J} cb_{p,j}\, B_{p,j} \tag{65}$$

The value of $cif_{p,p'}$ is the cost of reprocessing the interface $p - p'$, $cin_{p,s}$ is the cost of pumping a unit volume of product p from s, and $cb_{p,j}$ stands for the cost of failing to provide a single unit of p at depot j on time.

5 Case studies

5.1 Case study I: planning the transportation of a real-world single-source pipeline

The pipeline operational planning approach presented in Sect. 3 is applied to a modified version of the real-world case study introduced by Rejowski and Pinto (2003) to account for a much longer multi-period time horizon and multiple delivery due-dates. It considers the distribution of four refined petroleum products through a single trunk line. The pipeline operational plan over a rolling time horizon steadily comprising four weekly periods (*t1*–*t4*) is to be determined. Neither hard nor soft frozen periods are considered for planning the operations. Product demands initially required for periods *t1*–*t4* are first given. They should be delivered to local markets before the completion of each period. Such terminal requirements may be updated at

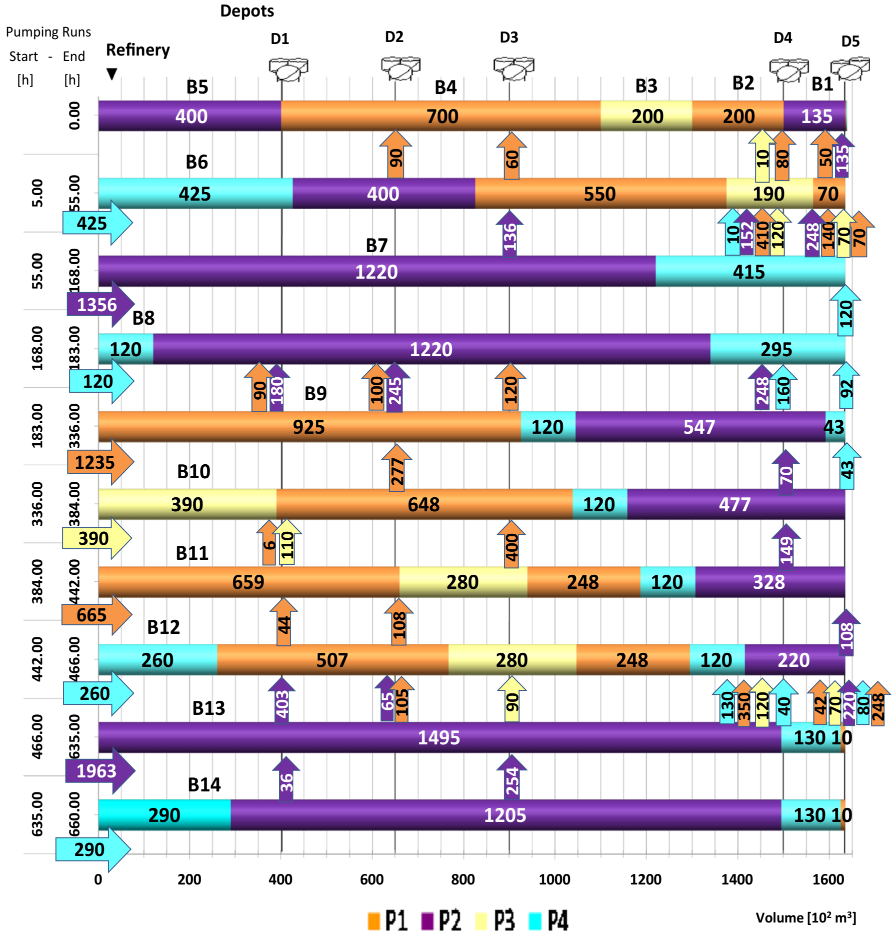


Fig. 6 Optimal pipeline transportation plan for time periods *t1*–*t4*

the start of any new instance of the rolling horizon. Demand data for the subsequent time periods *t5–t7* become gradually available as the horizon rolls with time. All problem data for this case study can be found in the work by Cafaro and Cerdá (2008b). The pipeline status at the start of the planning horizon is given in the first line of Fig. 6. Five batches {*B5, B4, B3, B2, B1*} containing products $P2_{400}$, $P1_{700}$, $P3_{200}$, $P1_{200}$, $P2_{135}$, respectively, are inside the pipeline at $t = 0$ (the subscripts indicate their initial volumes in 10^2 m^3). If the pipeline is activated, the pumping rate must be set between 800 and 1200 m^3/h.

5.1.1 Planning pipeline operations for {t1–t4}

At time $t = 0$, the first transportation plan for the initial horizon {*t1–t4*} is to be determined. The proposed pumping run schedule includes a sequence of five batches {*B6*, *B7*, *B8*, *B9*, *B10*} involving the following products and volumes (given as subscripts): $P4_{425}$, $P2_{1720}$, $P1_{1282}$, $P3_{430}$, $P1_{1180}$. But only the operations planned for the first week are executed. The pumping run of batch *B6* will be executed as originally planned. In contrast, the injection of *B7* within period *t1* will end at time 168 h. Therefore, it will last $(168 - 55) = 113$ h.

The mathematical model for every instance of the rolling time horizon was solved on a Pentium IV 2 GHz processor with CPLEX by using ILOG OPL Studio 3.7 (ILOG 2004). A relative MIP gap of 10^{-4} was adopted. After solving the MILP, the cardinality of I^{new} is increased by one and the model is solved again until no further decrease in the operation costs is achieved. The size of the model and the required CPU time to find the best pipeline schedule for the horizon {*t1–t4*} are both summarized in Table 1. This table also shows the numerical results for a 3-days time horizon proposed by previous authors (Rejowski and Pinto, 2003). Compared to discrete models, the computational requirements are reduced by almost two orders of magnitude. In contrast to discrete approaches, the solution is exact (no volume approximation), and the computational effort is independent of the horizon length (for a fixed number of batches).

5.1.2 Updated pipeline planning for the next horizon {t2–t5}

After solving the new problem instance, it can be observed that the injection of *P2* last shipped in period *t1* is interrupted to start pumping product *P4*. In addition,

Table 1 Model sizes and results for every instance of the single-source pipeline problem

Horizon	$\|I^{old}\|$	$\|I^{new}\|$	$\|I\|$	Binary variables	Continuous variables	Equations	CPU time (s)	Optimal solution (10^2 \$)
3-days	5	4	9	214	2135	3000	14.80	19373.84
t1–t4	5	5	10	240	2223	3380	15.63	175951.68
t2–t5	2	6	8	213	1958	3228	124.41	164681.95
t3–t6	4	6	10	273	2660	3882	216.33	181538.22
t4–t7	7	6	13	363	3418	4757	330.30	189873.39

there are quite significant changes in the pumping run sequence and the batch sizes. At the initial horizon {*t1-t4*}, the last two batches containing products *P3* and *P1* are pumped just to push forward the batches flowing into the line towards their stated terminals. However, their own destinations were still undefined since terminal requirements at period *t5* were unknown. As new demands for period *t5* arise at the more distant terminals *D4* and *D5*, batches of *P1* and *P3* are reduced just to the required volumes and a new batch of *P4* is next inserted. Similar to the previous horizon, the operational schedule for {*t2–t5*} includes the injection of two large batches to "sweep" previous shipments.

5.1.3 Pipeline transportation plan finally executed during the first month

At $t = 336$ h, only the pumping runs planned for the action period *t2* have been performed. The next step is to update the information and refresh the plan again. As the four-period horizon has rolled from {*t1–t4*} to {*t4–t7*} the pipeline transportation plan undergoes significant changes. The sequence of batches finally pumped and the amounts delivered to the terminals present some major differences with regards to the initial schedule for *t1–t4*. To meet customer demands, the pipeline remains operative from time 0 to 660 h. The injection of product *P3* is delayed until the start of the action period *t3* when new production of *P3* becomes available at the refinery tankage. Despite that, the pipeline system features a total idle time of 12 h over a time horizon length of 672 h. Variations of product inventories in refinery tanks are illustrated in Fig. 7. Further numerical experiments demonstrate that this model can also help planners to adjust the refinery production schedule to future demands (for instance, anticipating the availability of *P3*). A better integration of refinery and pipeline operations yields significant benefits.

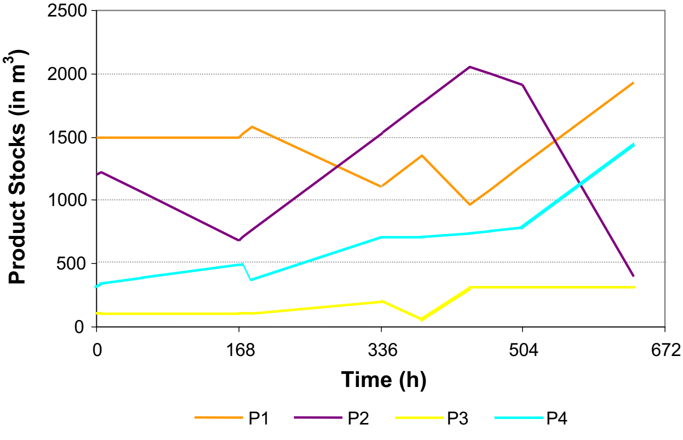


Fig. 7 Projected inventory levels in refinery tanks for time periods *t1–t4*

5.1.4 Single-source pipelines: conclusions

Results show that the sequence of pumping runs finally executed by the pipeline operator along the time horizon looks quite different from the one originally proposed. Pumping runs become shorter and its number is significantly increased. Such changes arise because in a static time horizon the pumping runs of later periods have the only purpose of pushing in-transit batches to their destinations. In the dynamic solution, no batch is dispatched only due to interface compatibility but mostly to satisfy specific terminal requests due at future periods. In fact, the pipeline idle time practically vanishes. Computational requirements grow as the time horizon rolls and the number of pumping runs increases, but in any case it remains quite reasonable varying from 16 to 330 CPU s. The model is flexible enough to dynamically adjust the transportation plan to changes in demands, refinery production runs or new batch destinations.

5.2 Case study II: planning the operation of a multi-source pipeline network

The second example deals with a real-world pipeline composed of 3 segments transporting 5 oil derivatives (*P1*–*P5*) from 2 sources (*N1*, *N3*) to 3 receiving depots (*N2*, *N3*, *N4*). Note that the pipeline network includes a dual-purpose node (*N3*) where pumping and delivery operations can take place simultaneously (see Fig. 8).

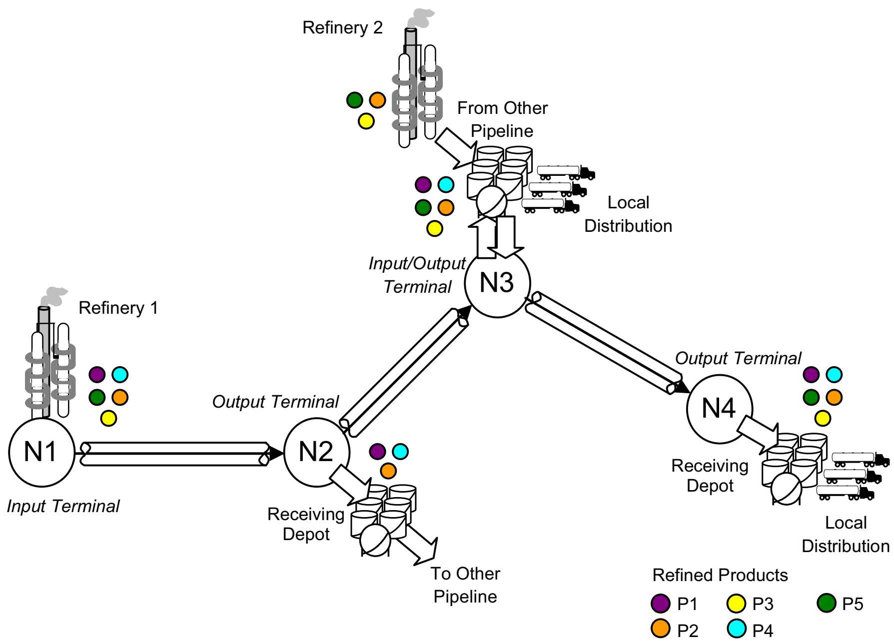


Fig. 8 The pipeline network for the second case study

The overall length of the pipeline system is over 1000 km and the injection rate at the two input nodes should be set between 310 and 580 m^3/h. Product demands for the next 10 days, product inventories at source nodes, unit pumping costs and interface reprocessing costs for this case study can be found in the work of Cafaro and Cerdá (2010). The initial linefill is shown in the first line of Fig. 9. The model assumes the existence of an empty batch *B3* to consider the possibility of injecting a new lot just in the interface between the initial lots *B4* and *B2*. The pumping time and the size of the new batch *B3* are both model decisions. The problem goal is to develop the pipeline operational plan for the next 10 days (240 h) in order to satisfy the specified depot needs at minimum total cost.

The best pipeline schedule shown in Fig. 9 includes a total of 12 batch injections (6 from each source) that are grouped into 8 blocks of parallel pumping runs. Since some runs just add further amounts of products to existing batches, 10 lots are moving throughout the system over the time horizon. The first operation (*k1*) running from time $t = 0.00$ to $t = 51.41$ h involves the pumping of a large lot $B7_{29820}$ of product *P2* at node *N1*. While doing so, the interface *B2–B4* moves forward just to reach the location of the intermediate source *N3*. The next runs *k2* and *k3* both involve simultaneous pumping operations at the two sources *N1* and *N3*. In each of the next four runs (*k4–k7*) a single batch injection is performed. Two runs (*k4*, *k6*) are made at source *N3* while the others take place at source *N1*. Finally, the last block *k8* comprises simultaneous, non-interacting batch injections. The best multisource pipeline schedule depicted in Fig. 9 was found in 333.0 s of CPU time on a 64 bits 4-processors (3.0 GHz) Pentium IV PC, with GAMS/GUROBI 3.0 as the MILP solver (Brooke et al. 2006) and a relative tolerance gap of 10^{-4}. By increasing the cardinality of the set *K* or the cardinality of the set *I*, no improvement in the optimal schedule is achieved. From Table 2 it can also be observed the sharp

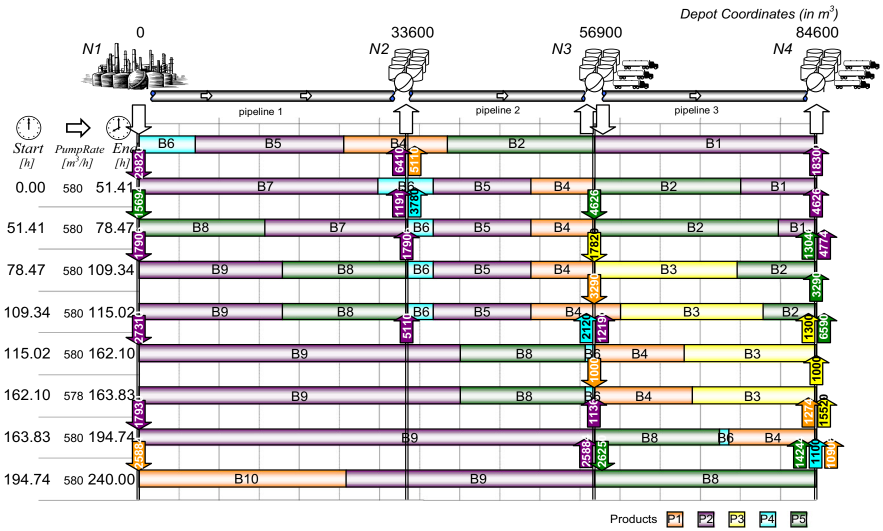


Fig. 9 Optimal schedule for the multi-source pipeline system

increase of the computational effort when the number of runs and batches is slightly changed.

5.2.1 Multiple-source pipelines: conclusions

Through the accurate coordination of product injections yielded by the solution of the proposed MILP model, product demands at distribution depots are fully satisfied within the time horizon. Further numerical experiments were conducted to assess the benefits of allowing parallel batch injections in multiple-source pipelines. By comparing the results, several interesting findings can be drawn. When blocks of parallel runs are permitted at large-size problems, the set of runs (*K*), the model size, and especially the required CPU time all usually decrease with regards to the sequential injection mode. The computational cost substantially diminishes because a large number of batch injections are represented using a single element of the set *K*. For case studies comprising up to 3 sources, 4 depots, 5 products, 10 batches and 8 runs, during a 10-days time horizon, the model size remains quite reasonable (3000 equations, 5000 continuous variables and 500 binaries) and the CPU time rarely exceeds 1000 s. The effects of including inventory carrying costs in the objective function of multiple-source pipeline models is left as future work.

6 Final conclusions

Computationally-efficient mixed-integer linear formulations for the short-term planning of pipeline networks with one and multiple sources have been synthesized. The proposed mathematical models are restricted to pipeline networks with unidirectional flow and a single duct between every pair of adjacent terminals. Dual-purpose stations performing input and/or output tasks are also managed. The optimization tools permit to determine the most convenient sequence of batch injections at every source node, the lot sizes, the start/end times of every pumping run, and the optimal batch allocation to receiving terminals, all at once. In addition, the models are able to track the location and size of product lots in different pipeline segments over the time horizon. Flow-rate variations due to changes in pipeline diameters are easily handled. In particular, the multisource formulation includes

Table 2 Model sizes and computational results for the multisource case

Runs |K|	Lots |I|	Eqs.	Cont. var.	Bin. var.	CPU time (s)	Iter. (10^6)	Opt. gap (%)	Pump cost (10^3 \$)	Interf. cost (10^3 \$)	Back orders (%)
8	10	4605	2504	450	333.0	6.8	0.0	1421.0	33.7	0.0
9	10	5139	2808	500	10,295.3	213.0	0.0	1421.0	33.7	0.0
8	11	5053	2750	495	20,000[a]	340.3	1.0	1421.0	33.7	0.0

[a] Optimality is not proved after 20,000 CPU s

simple mathematical constraints for planning blocks of non-interacting, simultaneous batch injections. A single pipeline segment can receive material from either an adjacent upstream segment or the tank farm at the source node. One of the major advantages of the proposed approach is the better utilization of the pipeline transportation capacity. In fact, more efficient operational plans are more rapidly discovered. A better coordination of input and output flows yields substantial savings in the pipeline operation costs.

Acknowledgments Financial support received from FONCYT-ANPCyT under Grant PICT 1763, from CONICET under Grant PIP 2221 and from Universidad Nacional del Litoral under CAI + D program is fully appreciated.

References

Boschetto SN, Magatão L, Brondani WM, Neves-Jr F, Arruda LVR, Barbosa-Póvoa APFD, Relvas S (2010) An operational scheduling model to product distribution through a pipeline network. Ind Eng Chem Res 49:5661–5682

Brooke A, Kendrick D, Meeraus A, Raman R (2006) GAMS: a user's guide. GAMS Development Corporation, Washington, DC

Cafaro DC, Cerdá J (2004) Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation. Comput Chem Eng 28:2053–2068

Cafaro DC, Cerdá J (2008a) Efficient tool for the scheduling of multiproduct pipelines and terminal operations. Ind Eng Chem Res 47:9941–9956

Cafaro DC, Cerdá J (2008b) Dynamic scheduling of multiproduct pipelines with multiple delivery due dates. Comput Chem Eng 32:728–753

Cafaro DC, Cerdá J (2010) Operational scheduling of refined products pipeline networks with simultaneous batch injections. Comput Chem Eng 34:1687–1704

Cafaro VG, Cafaro DC, Méndez CA, Cerdá J (2010) Oil-derivatives pipeline logistics using discrete-event simulation. In: Proceedings of the winter simulation conference, pp 2101–2113

Cafaro VG, Cafaro DC, Méndez CA, Cerdá J (2011) Detailed scheduling of operations in single-source refined products pipelines. Ind Eng Chem Res 50:6240–6259

Explorer Pipeline (2016) http://www.expl.com/Media/manuals/Section6NOMINATIONINSTRUCTIONS.pdf

García-Sánchez A, Arreche LM, Ortega-Mier M (2008) Combining simulation and tabu search for oil-derivatives pipeline scheduling. Stud Comput Intell 128:301–325

Hane CA, Ratliff HD (1995) Sequencing inputs to multi-commodity pipelines. Ann Oper Res 57:73–101

ILOG OPL Studio 3.7 (2004) A user's manual. ILOG S.A

Lopes TMT, Ciré AA, de Souza CC, Moura AV (2010) A hybrid model for a multiproduct pipeline planning and scheduling problem. Constraints 15:151–189

Magatão L, Arruda LVR, Neves-Jr FA (2004) Mixed integer programming approach for scheduling commodities in a pipeline. Comput Chem Eng 28:171–185

Miesner TO, Leffler WL (2006) Oil & gas pipelines in nontechnical language. Pennwell, Tulsa

Mori FM, Lüders R, Arruda LVR, Yamamoto L, Bonacin MV, Polli HL, Aires MC, Bernardo LFJ (2007) Simulating the operational scheduling of a realworld pipeline network. Comput Aided Chem Eng 24:691–696

Moura AV, de Souza CC, Cire AA, Lopes TM (2008) Planning and scheduling the operation of a very large oil pipeline network. Lect Notes Comput Sci 5202:36–51

Rabinow RA (2004) The liquid pipeline industry in the United States, where it's been, where it's going. Association of Oil Pipelines, Washington

Rejowski R, Pinto JM (2003) Scheduling of a multiproduct pipeline system. Comput Chem Eng 27:1229–1246

Rejowski R, Pinto JM (2004) Efficient MILP formulations and valid cuts for multiproduct pipeline scheduling. Comput Chem Eng 28:1511–1528

Rejowski R, Pinto JM (2008) A novel continuous time representation for the scheduling of pipeline systems with pumping yield rate constraints. Comput Chem Eng 32:1042–1066

Relvas S, Matos HA, Barbosa-Póvoa APFD, Fialho J, Pinheiro AS (2006) Pipeline scheduling and inventory management of a multiproduct distribution oil system. Ind Eng Chem Res 45:7841–7855

Sasikumar M, Prakash PR, Patil SM, Ramani S (1997) PIPES: a heuristic search model for pipeline schedule generation. Knowl Based Syst 10:169–175

Zyngier D, Kelly JD (2009) Multi-product inventory logistics modeling in the process industries. Optim Logist Chall Enterp 30:61–95