Computers and Chemical Engineering 61 (2014) 59–76

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Rigorous formulation for the scheduling of reversible-flow multiproduct pipelines

Diego C. Cafaro, Jaime Cerdá *

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

ARTICLE INFO

Article history:
Received 4 January 2013
Received in revised form 8 October 2013
Accepted 11 October 2013
Available online xxx

Keywords:
Multiproduct pipelines
Reversible-flow
Operational scheduling
Monolithic model
Continuous approach

ABSTRACT

Pipelines play a major role in the petroleum industry by providing a safe, reliable and economical transportation mode over land. Frequently, they connect a pair of refineries or harbors with the purpose of sharing oil products. As the construction of twin pipelines transporting products in opposite directions demands large capital investments, reversible-flow pipelines arise as a promising alternative. This paper introduces a novel continuous-time formulation for the short-term operational planning of reversible multiproduct pipelines. The proposed model allows to change the flow direction as many times as needed to meet terminal demands, determining precise time instants for flow reversals. It provides the input and output schedules in a single step, and the most convenient product used as filler to push current batches out of the line. Three examples are successfully solved with much less computational effort than previous approaches.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Oil pipelines are extensively used by petroleum refiners for carrying large quantities of crude oil and refined petroleum products over long distances. Crude oil pipelines gather oil flows coming from onshore and offshore well pads to send them to seaports or refineries. On the other hand, refined products pipelines transport multiple products such as gasoline, diesel oil and jet fuel, from refineries to distribution centers or harbors. Then, minor pipelines deliver oil products from distribution centers to major consumers like petrochemical plants or airports.

Regarding the flow direction, pipelines can be roughly classified into two types: unidirectional (non-reversible) and bidirectional (reversible) lines. Non-reversible pipelines are the common choice in the oil supply chain. Lots of crude oil are transported from oil fields to refineries, while refined product batches flow from refineries to distribution centers near dense population areas. Sometimes, however, refined products pipelines are used to exchange products between refineries, distribution centers or harbors in both directions. In those cases, the construction of a pair of unidirectional lines working in opposite directions is economically unjustified due to the high investment cost, and reversible pipelines appear as a less expensive alternative. This type of pipelines should be designed to reverse the flow in order to operate in both directions. If batches move on the most frequent direction, the bidirectional pipeline is said to operate in "direct flow". Pumping products in the opposite direction is called a "reverse flow" operation.

Besides the dual-purpose (input/output) stations at both extremes, reversible pipelines may include intermediate terminals receiving lots of products coming from any of both sources, as shown in Fig. 1.

Before reversing the flow direction, a key issue in the operation of bidirectional pipelines is the need of pushing all the batches in the current linefill to their assigned destinations. Hence, a large batch of a filler product should be pumped to drive the whole batch sequence into the receiving tank farms. Once the inputted batches have reached their destinations, the flow direction can be switched. At that moment, the farthest receiving terminal turns to be the input terminal and begins to pump batches in the opposite direction. In this way, the lot of filler returns to the assigned tanks at the origin. This procedure is repeated each time the flow is reversed, thus resulting in time delays and additional operation costs. Moreover, the large quantity of product used as filler should be available at the active source to sweeping off the pipeline linefill before the flow reversal.

Scheduling pumping and delivery operations in multiproduct pipeline systems is a complex logistic problem requiring advanced computational tools. Limited tank capacities, delivery dates and refinery production plans are problem constraints to be satisfied. Because different products are usually shipped through the same line without separation devices, a product mixture is formed in the interface of two consecutive batches. Such an undefined product (called "transmix") implies quality downgrading and/or reprocessing, with the resulting increase in operation costs. The scheduling

* Corresponding author. Tel.: +54 342 4559175; fax: +54 342 4550944.
E-mail address: jcerda@intec.unl.edu.ar (J. Cerdá).

0098-1354/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compchemeng.2013.10.006

Nomenclature

Sets

I	batches
J	terminals
K	pumping runs (chronologically arranged)
P	refined oil products
E	peak energy cost periods (chronologically arranged)

Subsets

$I^{new} \subset I$	new batches to be pumped in future runs
$I^{old} \subset I$	old batches initially in the pipeline
$ID \subset I^{new}$	batches to be pumped in direct flow (from j_D to j_R)
$IR \subset I^{new}$	batches to be pumped in reverse flow (from j_R to j_D)
$P_j \subset P$	products that can be pumped at source j

Parameters

$bc_{p,j}$	unit backorder penalty cost for product p at terminal j
c_{DR}/c_{RD}	unitary costs for reversing the pipeline flow direction
$dem_{p,j}$	demand for product p at terminal j
$d_{\min}/d_{\max}$	minimum/maximum size for a single delivery
$dr_{p,j}$	maximum delivery rate of product p dispatched to market from terminal j
$fr_{p,j}$	constant feed rate of product p to terminal j
$h_{\max}$	maximum length of the planning horizon
$ic_{p,j}$	unit inventory carrying cost for product p at terminal j
$id_{p,j}^{\min}/id_{p,j}^{\max}$	minimum/maximum levels for the inventory of product p at terminal j
$iface_{p,p'}$	size of the interface generated between products p and p'
$ifc_{p,p'}$	unit reprocessing/degrading cost for interface p–p'
$l_{\max}$	maximum run length
$pc_{p,j}$	unit pumping cost for product p injected at source j
pk	unit pipeline operating cost in daily peak hours
$pr_{\min}^{j}/pr_{\max}^{j}$	minimum/maximum pumping rates for injections at source j
pv	pipeline volume
$q_{\min}^{j}/q_{\max}^{j}$	minimum/maximum size for a batch injection at input terminal j
s_e/f_e	starting/ending time of peak period e
wo_i	initial size of the old batch i
ρ_{DR}/ρ_{RD}	setup times for reversing the pipeline flow direction
σ_j	volumetric coordinate of terminal j
$\tau_{p,p'}$	transition time between products p and p' successively pumped at the same source

Non-negative variables

$AT_{i,j}^{(k)}$	arrival time of lot i at terminal j during run k
$B_{p,j}$	backorder of product p at the customers supplied by terminal j
C_k/L_k	completion time/length of pumping run k
$D_{i,j}^{(k)}$	volume of batch i delivered to terminal j during run k
$DA_{p,j,i}^{(k)}$	amount of product p dispatched to the customers of terminal j during the time interval $[AT_{i-1,j}^{(k)}, AT_{i,j}^{(k)}]$
$DM_{p,j}^{(k)}$	amount of product p dispatched to the customers from depot j at run k
$DP_{i,j,p}^{(k)}$	volume of product p delivered to terminal j from batch i during run k
$F_i^{(k)}$	frontal coordinate of batch i at time C_k (measured from the origin)
H	optimal makespan
$ID_{p,j}^{(k)}$	inventory level of product p in tanks of terminal j at time C_k
$IA_{p,j,i}^{(k)}$	inventory of product p in terminal j at the arrival time of lot i during run k
$Q_i^{(k)}$	volume of batch i injected during run k
$QP_{i,p}^{(k)}$	volume of product p into batch i injected during run k
rd_k	variable denoting that run k changes from reverse to direct flow
rr_k	variable denoting that run k changes from direct to reverse flow
$SS_{p,j,i}^{(k)}$	use of safety stock of product p in terminal j at the arrival time of lot i during run k
ST_k	transition time between runs $(k-1)$ and k
$TK_{k,e}$	length of run k executed within peak period e
$WIF_{p,p'}^{(k)}$	interface volume between products p and p' generated by pumping run k
$W_i^{(k)}$	size of batch i at time C_k
$wd_{k,p}$	variable denoting that product p is pumped in direct flow during run k
$wr_{k,p}$	variable denoting that product p is pumped in reverse flow during run k

Binary variables

$u_i^{(k)}$	denotes that pumping run k injects lot $i \in ID$ (in direct flow) at j_D
$v_i^{(k)}$	denotes that pumping run k injects lot $i \in IR$ (in reverse flow) at j_R
$wf_{k,e}$	denotes that the completion time of run k is greater or equal to the ending time of peak period e whenever $wf_{k,e} = 1$
$ws_{k,e}$	denotes that the initial time of run k is lesser or equal to the starting time of peak period e, whenever $ws_{k,e} = 1$
$x_{i,j}^{(k)}$	denotes that a portion of lot i is discharged into terminal j during run k
$y_{i,p}$	denotes that product p is assigned to lot i

problem becomes even more difficult for reversible lines because the pipeline planner should decide on the time intervals during which the pipeline operates in forward or backward direction, and on what product to use as filler before a flow reversal, among other issues. Planning the size and the sequence of new batches to pump into a reversible pipeline and the simultaneous product deliveries to receiving depots along the line is a difficult task demanding accurate and efficient models. This paper introduces a novel mixed-integer linear programming (MILP) formulation for the reversible pipeline scheduling problem, whose solution provides both the input and output operational plans in a single step. The proposed model can be regarded as a generalization of the continuous-time representation introduced by Cafaro and Cerdá (2004, 2008a) for the scheduling of unidirectional pipelines.

1.1. Literature review

In recent years, a considerable amount of work has been done on multiproduct pipeline scheduling. A broad class of pipeline scheduling problems, from simpler cases involving a unidirectional pipeline with a single origin and a single destination (Cafaro & Cerdá, 2008b; Relvas, Matos, Barbosa-Póvoa, Fialho, & Pinheiro, 2006), or a sequence of major trunk lines with several receiving terminals (Cafaro & Cerdá, 2004, 2009; Hane & Ratliff, 1995; Rejowski & Pinto, 2003, 2008), to more complex instances like the scheduling of tree-structure (Cafaro & Cerdá, 2011; Castro, 2010; MirHassani & Jahromi, 2011) and mesh-structure pipeline networks connecting refineries, distribution centers and customers' facilities (Cafaro & Cerdá, 2012; Herrán, de la Cruz, & de Andrés, 2010, 2012) have already been tackled.

Besides the pipeline network topology, scheduling approaches can be classified into two categories: detailed mathematical models and decomposition procedures. The most detailed representations intend to precisely follow the movement of product lots along the ducts throughout the planning horizon, while the other approaches no longer trace the batches into the pipelines but are essentially focused on the evolution of the product inventories at every terminal (Moura, de Souza, Cire, & Lopes, 2008). These techniques are usually based on decomposition strategies that initially determine the size, origin, destination and route of each lot, and then find out the most convenient batch sequence and pumping schedule across the pipeline network. The first stage is usually accomplished through heuristic-based procedures whereas the second step alternatively applies rigorous MILP models (Boschetto et al., 2010), constraint programming (Lopes, Ciré, de Souza, & Moura, 2010) and discrete-event simulation tools (García-Sánchez, Arreche, & Ortega-Mier, 2008).

Regarding detailed scheduling models, both discrete-time and continuous-time mathematical formulations have been reported in the literature during the last decade. Discrete models divide both the pipeline volume into single-product packs and the planning horizon into time intervals of fixed duration. They are based on the fact that pipelines are always full of products, and due to the liquid incompressibility assumption, every time a new pack enters the pipeline another pack of the same size is discharged into the tanks of some receiving terminal. Discrete models were initially proposed for pipeline systems with a single entry and several destinations (Hane & Ratliff, 1995; Rejowski & Pinto, 2003; Sasikumar, Prakash, Patil, & Ramani, 1997) and then extended to manage bidirectional pipeline systems (Magatão, Arruda, & Neves-Jr, 2004) and mesh-like pipeline networks involving segments working in both directions (Herrán et al., 2010). However, no feasible solution will be found unless a fine discretization of time and volume domains is made. Moreover, the CPU time needed to reach the optimal solution could become extremely high. On the contrary, continuous-time models appear as the most efficient alternative in terms of the computational cost. Nevertheless, up to now, the problem of sizing and sequencing oil product batches moving through reversible pipelines has not been solved in a rigorous and comprehensive way, using a continuous-time formulation.

Magatão et al. (2004) are the first to present an optimization approach based on a mixed-integer programming model for scheduling batch injections in a reversible pipeline. It is applied to a real world Brazilian pipeline connecting a refinery to an important harbor with no intermediate depots. As the proposed model relies on the uniform discretization of time and volume domains, it assumes a fixed pumping rate for batches moving in any of both directions. The objective function tends to minimize transition costs that are significantly high in case "plugs" of convenient products are required to separate pairs of successive, non-compatible batch injections. Because pumps are electricity-driven, daily peak periods of high cost are also taken into account. Regarding the computational burden, the authors argue that the integer search space is excessively large if the discrete MILP formulation is solved as a monolithic model, even for small problem instances. As a result, they propose an alternative optimization structure that decomposes the problem into three phases: (a) the tank bound procedure, (b) an auxiliary routine, and (c) the main model. Step (a) consists of the pre-selection of the tanks used at both extremes of the reversible pipeline to satisfy pumping activities, while step (b) is a pre-processing module that reduces the search space by roughly bounding the time intervals during which the pipeline should operate on direct or reverse flow. Here it comes one of the most critical assumptions of the model stating that the pipeline operates only once in direct flow, and only once in reverse flow throughout the planning horizon. Moreover, the pipeline planner should choose the initial flow direction before solving the model. Another drawback of this approach is the need of solving the main model for different values of the problem makespan (i.e. the length of the scheduling horizon). In other words, the optimal makespan cannot be determined at once but through an iterative procedure.

More recently, the same authors (Magatão, Arruda, & Neves-Jr, 2011) refine their previous approach by combining constraint logic programming (CLP) and MILP constraints to model the main stage of the decomposition strategy. The new version presents a more efficient, hybrid model that permits to reduce the computational burden of the previous methodology by one or two orders of magnitude. The time domain is handled in a continuous scale, but the pipeline volume is still discretized into fixed-sized packs of 1800 m^3. Besides the discrete nature of the model, the initial flow direction is also predefined and it cannot be reversed more than once over the planning horizon. Similar to the previous approach, the scheduling horizon is iteratively extended until the minimum cost solution is achieved. Another critical feature is that the evolution of the total cost with the horizon length may show more than one local minimum.

Herrán et al. (2010) propose the first detailed pipeline scheduling model for the transportation of multiple petroleum products in a mesh-structure multi-pipeline system, including a reversible segment between a pair of terminals. As the model assumes that every pipeline segment has a single origin and a single destination, no pipeline in the network can have intermediate output terminals. The approach is based on a monolithic MILP model totally discretized in time and volume domains. Every pipeline segment in the network has a single origin and a single destination and is uniformly divided into equal-sized packages of 5000 m^3, while the time horizon involves 20 fixed intervals of 5 h. The authors recognize that solving rather small problems, with 4 products, 7 terminals, 7 pipelines, even using a rough time and volume discretization, may require more than 20,000 CPUs to reach the optimal solution. For that reason they derive a simplified version of the original model by assuming that all the pipelines in the network are always active to avoid pipeline start/stop cost contributions. In this way, the computational burden is reduced by one order of magnitude but the assumption often excludes the optimal solution from the feasible region. In fact, such a simplification can lead to non-optimal solutions featuring operating costs 20% above the minimum value, especially for low-demand scenarios. A particular feature of the reported solution is that the reversible segment is operated only once at each direction, although it can be reversed as many times as necessary. Contrarily to previous approaches, the initial pumping direction is conveniently determined by the model.

In a more recent work, the same authors (Herrán et al., 2012) develop several metaheuristic algorithms to efficiently solve the discrete models previously published. The algorithms use monotonous searches (multi-start search, variable neighborhood search), non-monotonous searches (tabu search, simulated annealing), and multiple Markov chain models. Given the NP-hard combinatorial nature of the proposed pipeline network models, some metaheuristics (particularly the simulated annealing) show certain capability for reducing the computational burden by up to two orders of magnitude with regards to the rigorous MILP

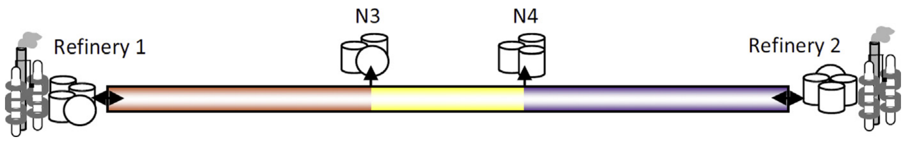

Fig. 1. A reversible pipeline system.

model, for rather small problems. Nonetheless, the optimality of the solution cannot be ensured unless the rigorous formulation is applied. One major drawback of the metaheuristic techniques is clearly the very low number of feasible solutions that are found. In fact, no more than 25 feasible solutions are discovered by any algorithm after 500 CPUs.

In contrast to previous methodologies, this work introduces the first rigorous approach for the operational planning of reversible pipelines. It consists of an MILP continuous-time formulation that accounts for the possibility of inverting the flow direction as many times as necessary at the proper time instants. Batch movements are rigorously traced along the line in both directions, and inventory levels are efficiently monitored in a continuous way. The model can optimally select pump rates and choose the most convenient product to be pumped as the filler batch. The problem goal is to find the optimal sequence of pumping and delivery operations that satisfy terminal requirements at minimum total cost, including pumping, mixing, backordered demand, and inventory carrying costs.

2. Problem statement and major model elements

2.1. Problem statement

Let us consider a pipeline system transporting batches of petroleum derivatives from oil refineries located at both extremes of the line either in direct or reverse flow, as depicted in Fig. 2. When the pumping station at the upstream origin is active, batches move forward to downstream depots. As soon as the input activity shifts to the other source, the flow direction is reversed and there is an upstream batch movement. Both extreme terminals do not only inject new batches into the line but also receive products coming from the other source. In other words, they are dual-purpose (input/output) stations. Some products transported through the pipeline are supplied by only one of the input stations. Between the two entry points, there are additional intermediate depots receiving lots of products from both sources. At each pipeline terminal, fuel tankers transporting lots of refined products are dispatched to meet the demand of nearby customers. Product inventories at sources are continually increased by new refinery supplies, and decreased by both the injection of new batches into the pipeline and the delivery of products to neighboring customers. Initial inventories, product supply rates to refinery tanks, and delivery rates from terminals to customers are all problem data. The problem goal is to develop the input schedule for the two extreme sources, and the output schedule for every depot using a continuous-time model-based approach.

Handling flow reversals poses new challenges in the modeling of multiproduct pipeline scheduling problems. Previous continuous-time formulations can no longer be applied without introducing some suitable changes for managing batch movements in both directions. This new problem feature adds further difficulties especially for tracking the batches and interfaces generated by new product inputs, and the need of filling lots to push out the pipeline content to receiving depots before reversing the flow direction.

2.2. Major model sets

Batches of refined products can be gathered into two groups: old (I^{old}) and new batches (I^{new}). Old batches reside in the pipeline at the initial time, while new batches are those pumped into the line from either input station over the time horizon. Hence, $I^{new} = IR \cup ID$, where IR is the subset of "reverse" batches to inject from the input station at the pipeline downstream end, and ID represents the "forward" or direct batches to pump at the upstream origin. To track the location of every batch moving through the line, a special arrangement of the elements in the set I given by: $I = IR \cup I^{old} \cup ID$ is required (see Fig. 2). Let us assume that four batches are initially contained in the pipeline and three lots are at most inserted in the line from either input station. Then, $IR = \{b1, b2, b3\}$, $I^{old} = \{b4, b5, b6, b7\}$, and $ID = \{b8, b9, b10\}$. Batches pumped into the line at the direct input terminal will be chronologically injected in the same order that they arise in the set ID. As a result, batch i will be injected before lot $i+1$ at the origin. In contrast, reverse batches will be pumped in the opposite order in which they appear in the set IR, i.e. lot $i+1$ is injected in the line before lot i.

Products transported by the pipeline can also be gathered into two subsets, i.e. $P = P_D \cup P_R$. The set P_D includes all the refined products available at the upstream origin and moving forward, while P_R comprises the fuels on hand at the downstream source to be pumped in the reverse direction. The sets P_D and P_R may share some products but generally contain non-common species. Before reversing the flow, a lot of a filler product should be inserted from the current active terminal to push out the whole pipeline linefill to receiving depots. The filler can be any of the products available at the current active source and should be properly chosen.

Pipeline terminals are of two types: dual purpose stations at the extreme sections, and pure receiving depots at intermediate locations. They are listed in the set J in the same order they appear along the pipeline, starting from the upstream origin, i.e. $J = \{j_D, j_1, j_2, \ldots, j_R\}$, as depicted in Fig. 2.

In contrast to single-source, unidirectional pipeline scheduling models, the batches in the line are not chronologically arranged and two separate sets are defined to represent the lots (I) and the pumping runs (K). The elements of the set K are listed in the same order that they are performed at either pipeline source. As the number of batches to inject at each source and the number of pumping

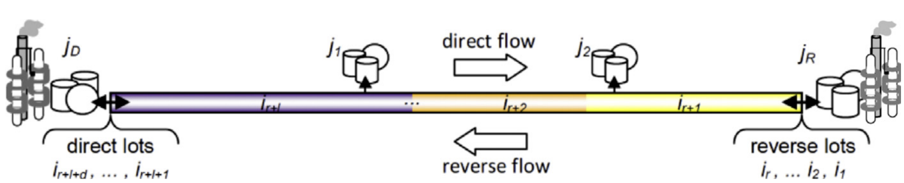

Fig. 2. A bidirectional pipeline and the definition of the set of batches.

runs to carry out over the planning horizon are not known before solving the model, the values of $|I^{new}|$ and $|K|$ must be properly estimated. They should be rather low but large enough to avoid excluding the optimal schedule. A good estimation for $|I^{new}|$ is the number of products to be injected in the line ($|P|$) to meet the specified terminal demands. This is so for two reasons: (1) a batch $i \in I^{new}$ of product p can be reused and injected again at the same terminal after a flow reversal, as explained in Section 4.5.3; and (2) the size of a lot of product p recently injected can be enlarged by subsequently performing an additional injection of that product at the same input terminal. In other words, two consecutive injections separated by an idle period may involve the same batch. On the other hand, an acceptable guess for $|K|$ can be obtained by multiplying the estimate of $|I^{new}|$ by a low integer m = 1, 2, . . ., whose value depends on the required number of flow reversals. Multiple changes in the flow direction may be needed because the demanded amount of a product cannot be inputted in the line through a single pumping run given the limited inventories at the input terminal.

2.3. New model variables

Compared with continuous-time formulations for the operational scheduling of unidirectional pipelines, three groups of variables are defined for modeling reversible pipeline operations. They are: (a) the binary variables $u_i^{(k)}$ and $v_i^{(k)}$ choosing the flow direction (direct flow if $u_i^{(k)} = 1$, or reverse flow if $v_i^{(k)} = 1$) and the inputted batch i during the pumping run k; (b) the continuous variables rr_k and rd_k with values confined to the closed interval [0, 1] to indicate the occurrence of a change in the flow direction from forward to backward ($rr_k = 1$) or vice versa ($rd_k = 1$); (c) the continuous variables $\omega d_{k,p}$ and $\omega r_{k,p}$ also restricted to the closed interval [0, 1] to denote that the product injected in direct or reverse flow during run k is p, whenever they equal one. The last group of variables is needed to better track both the interface generation and the setup tasks between consecutive operations when flow reversals are permitted. On the other hand, it is important to know through the variables rr_k and rd_k if a flow reversal occurs at the start of run k. If so, a filler batch should be injected during the previous operation $(k-1)$.

3. Model assumptions

The model formulation is based on the following assumptions:

(1) The pipeline system comprises a pair of sources that are located one at the upstream origin and the other at the downstream end section.
(2) Each source node can input batches in either forward or backward flow direction.
(3) At most one direction can be active at any time.
(4) Different sets of products are available at each source.
(5) Each source can also receive flows of products coming from the other source. They are indeed dual-purpose (input/output) terminals.
(6) Storage tanks at source nodes receive flows of products from the associated refineries, at product-dependent rates.
(7) Other output terminals at intermediate locations can also receive product flows coming from the two extreme sources.
(8) Output terminals dispatch products to nearby customers at given delivery rates.
(9) Storage tanks of each product are managed in an aggregate manner.
(10) Finite setup times between consecutive runs from the same input station or due to flow reversals are taken into account.
(11) Constant p–p' interface volumes are considered for evaluating the pipeline operation costs.
(12) A filler batch should be injected into the pipeline to sweep out the whole linefill before a flow reversal.

4. Problem constraints

The proposed mathematical formulation for the scheduling of reversible pipelines with dual (input/output) terminals at both extreme sections and several output nodes along the line includes eight groups of constraints standing for: (1) sizing batches and choosing the length and sequence of pumping runs, products, active sources and flow directions; (2) planning flow reversals and filler injections; (3) tracing the generation of new interfaces between adjacent batches; (4) managing setup operations at input terminals; (5) tracking the size and location of batches in the pipeline; (6) checking feasibility conditions for batch injections and product deliveries to terminals; (7) monitoring product inventories in terminal tanks; and (8) operating the pipeline during on-peak energy periods. Their mathematical expressions are given below.

4.1. Lot sizing and pumping run sequencing constraints

4.1.1. Sequencing pumping runs

Because the set of pumping runs K is chronologically arranged, run k is started after completing run $(k-1)$ and performing the setup operation

$$C_k - L_k \geq C_{k-1} + ST_k, \quad \forall k \in K \tag{1}$$

In constraint (1), ST_k is the setup time required to start with operation k, and the variables L_k and C_k are the length and the completion time of run k. As shown in Section 4.4, if runs $(k-1)$ and k are performed at the same input station the variable ST_k depends on the products being consecutively injected. If not, run k reverses the flow and ST_k is determined by the length of the flow reversal operation. Constraint (2) assumes that every pumping run is completed before the end of the planning horizon. Moreover, L_k is the earliest time at which run k can finish

$$L_k \leq C_k \leq H \leq h_{\max}, \quad \forall k \in K \tag{2}$$

4.1.2. Choosing the active input terminal and the flow direction at every run

At most a single batch injection can be performed at any time. According to restriction (3), the pipeline can operate in direct or reverse flow, but not in both. The new batch can be pumped into the line either from: (a) the direct source j_D to go forward toward downstream terminals, or (b) the reverse source j_R to travel backward toward upstream depots. Binary variables $u_i^{(k)}$ and $v_i^{(k)}$ are defined to choose the active input station, the batch and the flow direction for run $k \in K$. If $u_i^{(k)} = 1$, j_D is the active source and the inputted batch $i \in ID$ moves forward in direct flow. Instead, $v_i^{(k)} = 1$ means that the batch $i \in IR$ is injected in the line from terminal j_R. If run k is never performed (i.e. a “dummy” element), $u_i^{(k)} = v_i^{(k)} = 0$ for any new lot $i \in I^{new} = ID \cup IR$, and the LHS of constraint (3) is null

$$\sum_{i \in ID} u_i^{(k)} + \sum_{i' \in IR} v_{i'}^{(k)} \leq 1, \quad \forall k \in K \tag{3}$$

4.1.3. Sizing new batches and choosing the length of pumping runs

To select the volume of an inputted batch, lower and upper limits on the lot size must be taken into account, which may also depend on the flow direction. In constraints (4) and (5), those values are given by ($q^D_{\min}$, $q^D_{\max}$) and ($q^R_{\min}$, $q^R_{\max}$) for direct (D) and reverse (R) flow, respectively. The feasible size range only applies to non-fictitious batches. If k is a dummy run, $Q_i^{(k)} = 0$ for any new lot $i \in I^{new}$

$$q^D_{\min} u_i^{(k)} \le Q_i^{(k)} \le q^D_{\max} u_i^{(k)}, \quad \forall k \in K, i \in ID \tag{4}$$

$$q^R_{\min} v_i^{(k)} \le Q_i^{(k)} \le q^R_{\max} v_i^{(k)}, \quad \forall k \in K,\ i \in IR \tag{5}$$

Similarly, the pump rate can take different values if the pipeline operates in direct or reverse flow. It is so because the terrain profile over which the pipeline is laid out usually favors one of the movements and the operating pressure may change with the flow direction. The length of an active run k is determined by constraint (6), which controls that the pump rate is within the feasible ranges ($pr^D_{\min}$, $pr^D_{\max}$) or ($pr^R_{\min}$, $pr^R_{\max}$), for direct or reverse flow respectively. For a dummy run k, all the summations are null and its length L_k is equal to zero

$$\left[\sum_{i \in ID} \frac{Q_i^{(k)}}{pr^D_{\max}} + \sum_{i' \in IR} \frac{Q_{i'}^{(k)}}{pr^R_{\max}}\right] \le L_k \le \left[\sum_{i \in ID} \frac{Q_i^{(k)}}{pr^D_{\min}} + \sum_{i' \in IR} \frac{Q_{i'}^{(k)}}{pr^R_{\min}}\right], \quad \forall k \in K \tag{6}$$

4.1.4. Symmetric breaking constraints

Symmetric solutions are prevented without risk of excluding the optimum by pumping new batches into the pipeline in an appropriate order. On the one hand, direct flow lots should be injected from the input terminal j_D at the origin in the same order that they arise in the corresponding set ID. In other words, the direct lot $i \in ID$ can only be pumped through run $k \in K$ if the preceding lot $(i-1) \in ID$ has already been inserted into the line during a previous operation $k' < k$. In this way, fictitious batches that are never injected will arise last in the set ID. On the other hand, reverse flow batches $i \in IR$ should be pumped at the source node j_R in the opposite order that they are listed. If run k inserts a batch $i \in IR$ in reverse flow direction, then lot $(i+1) \in IR$ should have been pumped before, at a prior run $k' < k$. As a result, dummy lots appear first in the set IR. Symmetric breaking constraints are given by expressions (7) and (8)

$$u_i^{(k)} \le \sum_{k' \in K, k' < k} u_{i-1}^{(k)}, \quad \forall k \in K, i \in ID, i > first(ID) \tag{7}$$

$$v_i^{(k)} \le \sum_{k' \in K, k' < k} v_{i+1}^{(k)}, \quad \forall k \in K, i \in IR, i < last(IR) \tag{8}$$

4.1.5. Assigning products to batches

Every batch $i \in I^{new}$ must contain a single refined product. Let us use the 0–1 variable $y_{i,p}$ to denote the assignment of product p to the batch i. As the sets of products (P_D, P_R) available at sources (j_D, j_R) are different, a pair of product assignment constraints given by (9) and (10) are incorporated into the problem formulation. In this way, the model assigns one of the available products at the input terminals (j_D, j_R) to every new batch injected into the pipeline from either sources

$$\sum_{p \in P_D} y_{i,p} \le 1, \quad \forall i \in ID \tag{9}$$

$$\sum_{p \in P_R} y_{i,p} \le 1, \quad \forall i \in IR \tag{10}$$

If no product is assigned to lot $i \in I^{new}$ (i.e. $y_{i,p} = 0$ for any product p), then lot i is a fictitious batch never pumped into the line. On the contrary, the assignment of some product p to batch $i \in I^{new}$ implies that lot i is injected by a certain run k. More specifically, if the direct-flow batch $i \in ID$ features $y_{i,p} = 1$ for a particular product $p \in P_D$, then $u_i^{(k)} = 1$ for some run $k \in K$. In turn, $v_i^{(k)}$ must be equal to 1 for at least a certain run k whenever $y_{i,p} = 1$, with $p \in P_R$. Both conditions are mathematically described by restraints (11) and (12)

$$\sum_{p \in P_D} y_{i,p} \le \sum_{k \in K} u_i^{(k)} \le \left|K\right| \sum_{p \in P_D} y_{i,p}, \quad \forall i \in ID \tag{11}$$

$$\sum_{p \in P_R} y_{i,p} \le \sum_{k \in K} v_i^{(k)} \le \left|K\right| \sum_{p \in P_R} y_{i,p}, \quad \forall i \in IR \tag{12}$$

4.2. Flow reversal constraints

4.2.1. Planning changes in the flow direction

Let us introduce two continuous variables rr_k and rd_k whose values are confined to the closed interval [0, 1]. They are meant to determine if the flow direction is inverted at the start of run k. The condition $rr_k = 1$ indicates that the pipeline flow switches from forward to backward direction, and the active source shifts from j_D to j_R at run k. If instead $rd_k = 1$, the flow direction changes from backward to forward at run k. When both variables are zero ($rr_k = rd_k = 0$), there is no change in the flow direction at the start of run k. Flow reversals are traced through constraints (13)–(16). The condition $rr_k = 1$ arises if the active source for run $(k-1)$ is j_D ($\sum_{i \in ID} u_i^{(k-1)} = 1$) and run k is performed at the source node j_R ($\sum_{i \in IR} v_i^{(k)} = 1$). The opposite situation leads to $rd_k = 1$. Monitoring flow reversals is important for determining setup times, tracking interfaces and filler injections

$$rr_k \ge \sum_{i' \in IR} v_{i'}^{(k)} + \sum_{i \in ID} u_i^{(k-1)} - 1, \quad \forall k \in K \tag{13}$$

$$rr_k \le \sum_{i' \in IR} v_{i'}^{(k)};\ rr_k \le \sum_{i \in ID} u_i^{(k-1)}, \quad \forall k \in K \tag{14}$$

$$rd_k \ge \sum_{i \in ID} u_i^{(k)} + \sum_{i' \in IR} v_{i'}^{(k-1)} - 1, \quad \forall k \in K \tag{15}$$

$$rd_k \le \sum_{i \in ID} u_i^{(k)};\ rd_k \le \sum_{i' \in IR} v_{i'}^{(k-1)}, \quad \forall k \in K \tag{16}$$

4.2.2. Injecting a filler batch

Before switching the active source and reversing the flow direction, a filler lot must be injected to sweep out the current linefill to the assigned destinations. If $rr_{k+1} = 1$, the flow direction changes from forward to backward at the start of $(k+1)$. Therefore, a filler lot i featuring a size $W_i^{(k)}$ equal to the pipeline volume pv should be pumped into the line from j_D during the previous run k. The pumped volume $Q_i^{(k)}$ can be greater than $W_i^{(k)}$ due to the delivery of some amount of filler product to the output terminals over run k. The filler can be any of the products $p \in P_D$ available at the source j_D. The model will probably choose the one generating a low-size interface. The filler batch injection before reversing the flow from forward to backward is imposed by inequality (17): the lot $i \in ID$ inserted in the line from source j_D during run k should have a size $W_i^{(k)} = pv$ if a flow reversal occurs in the next operation $(k+1)$ and $rr_{k+1} = 1$. Similarly, the lot $i \in IR$ pumped at station j_R during run k right before a flow reversal (i.e., when $rd_{k+1} = 1$) should have a size $W_i^{(k)} = pv$ because of constraint (18)

$$W_i^{(k)} \ge pv(u_i^{(k)} + rr_{k+1} - 1), \quad \forall i \in ID, k \in K \tag{17}$$

$$W_i^{(k)} \ge pv(v_i^{(k)} + rd_{k+1} - 1), \quad \forall i \in IR, k \in K \tag{18}$$

4.3. Tracking the generation of new product interfaces

We assume that the injection of a filler lot before a flow reversal pushes all the product interfaces flowing into the pipeline toward the tanks of the opposite extreme terminal. Therefore, changing the flow direction at run k implies removing all old interfaces generated up to run $(k-1)$ and creating an additional one. As a result, the interface volume does not depend on the flow direction but on the product sequence. Because of the filler injection condition, every product pumped in any direction generates an interface with the product inputted at the preceding run, even if there is a flow reversal. New continuous variables $wd_{k,p}$ and $wr_{k,p}$ are needed to determine if the direct or reverse lot injected by run k contains product p. Their values are restricted to the closed interval [0, 1]. Constraint (19) indicates that $wd_{k,p}=1$ only if a lot $i \in ID$ carrying product $p \in P_D$ ($y_{i,p}=1$) is inserted at source j_D through run k ($u_i^{(k)}=1$). Similarly, restraint (20) sets the value of $wr_{k,p}$ to one if operation k taking place at the extreme terminal j_R injects a lot $i \in IR$ containing product $p \in P_R$

$$wd_{k,p} \geq u_i^{(k)} + y_{i,p} - 1, \quad \forall i \in ID, k \in K, p \in P_D \tag{19}$$

$$wr_{k,p} \geq v_i^{(k)} + y_{i,p} - 1, \quad \forall i \in IR, k \in K, p \in P_R \tag{20}$$

The values of the variables $wd_{k,p}$ and $wr_{k,p}$ for two consecutive runs $(k-1)$ and k determine the interface produced by the injection k. Let us assume that the parameter $iface_{p,p'}$ stands for the size of the interface generated by pumping product p' right after a batch of product p. Moreover, let the continuous variable $WIF_{p,p'}^{(k)}$ represent the volume of the interface created by run k, just in case products p and p' are inserted in this order into the line through the consecutive injections $(k-1)$ and k. If so, $WIF_{p,p'}^{(k)} = iface_{p,p'}$. Otherwise, $WIF_{p,p'}^{(k)} = 0$. If the flow direction is reversed at the start of run k, operations $(k-1)$ and k take place at different source nodes, and $(k-1)$ stands for the filler injection. Otherwise, both runs are performed at the same input terminal. These conditions are enforced by constraints (21)–(24). Although they are posed as inequalities, $WIF_{p,p'}^{(k)}$ is set equal to either $iface_{p,p'}$ or zero because the model seeks to minimize the interface costs

$$WIF_{p,p'}^{(k)} \geq iface_{p,p'}(wd_{k-1,p} + wd_{k,p'} - 1), \quad \forall k > 1; p, p' \in P_D; p \neq p' \tag{21}$$

$$WIF_{p,p'}^{(k)} \geq iface_{p,p'}(wr_{k-1,p} + wr_{k,p'} - 1), \quad \forall k > 1; p, p' \in P_R; p \neq p' \tag{22}$$

$$WIF_{p,p'}^{(k)} \geq iface_{p,p'}(wd_{k-1,p} + wr_{k,p'} - 1), \quad \forall k > 1, p \in P_D, p' \in P_R, p \neq p' \tag{23}$$

$$WIF_{p,p'}^{(k)} \geq iface_{p,p'}(wr_{k-1,p} + wd_{k,p'} - 1), \quad \forall k > 1, p \in P_R, p' \in P_D, p \neq p' \tag{24}$$

Assuming that the pipeline is initially filled with product p^*, the interface generated by the first run $k=1$ injecting product p from either source j_D or j_R is given by constraints (25) and (26)

$$WIF_{p^*,p}^{(k)} \geq iface_{p^*,p} wd_{k,p}, \quad \forall p \in P_D (p \neq p^*), k = 1 \tag{25}$$

$$WIF_{p^*,p}^{(k)} \geq iface_{p^*,p} wr_{k,p}, \quad \forall p \in P_R (p \neq p^*), k = 1 \tag{26}$$

Fig. 3 presents a simple example to illustrate the interface generation when flow reversals are allowed. The pipeline has two input stations (j_D, j_R) at the extreme sections but no intermediate terminal. It is initially filled with product $P1$. During the first run $k1$, the pipeline operates in direct flow and product $P2$ is injected. The interface volume generated by run $k1$ is $WIF_{P1,P2}^{(k1)} = iface_{P1,P2}$. The second run $(k2)$ pumps product $P3$ (a filler batch) also in direct flow. Hence, $WIF_{P2,P3}^{(k2)} = iface_{P2,P3}$. At the end of run $k2$, both interfaces $P1$–$P2$ and $P2$–$P3$ are stored in tanks of terminal j_R for reprocessing, and the pipeline is full of product $P3$. The flow direction is reversed in the next operation $k3$ and product $P4$ is injected at the source j_R. The new interface created by run $k3$ is $P3$–$P4$, i.e. the products inserted during the last two injections $k3$–$k4$. Both batches travel through the pipeline in reverse direction toward terminal j_D during operation $k3$.

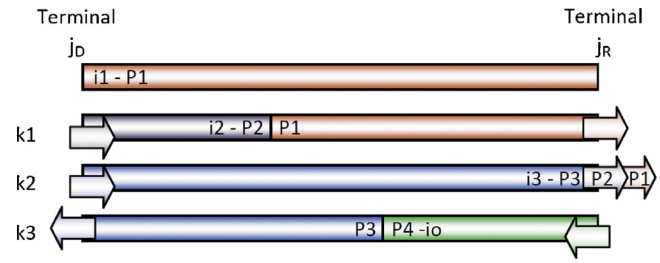

Fig. 3. A simple example illustrating the interface generation in reversible pipelines.

4.4. Determining setup times

Two consecutive runs $(k-1, k) \in K$ in a reversible pipeline can be performed at the same input station, or alternatively a flow reversal takes place and the input activity moves from one source to the other at the start time of run k. The model parameter $\tau_{p,p'}$ represents the length of the setup operation for the first case, i.e. if runs $(k-1, k)$ are performed at the same source node and inject products p and p', respectively. In such a case, if the active source is j_D, $wd_{k-1,p} = wd_{k,p'} = 1$, or $wr_{k-1,p} = wr_{k,p'} = 1$ if both injections occur at the reverse node j_R. On the other hand, the setup time for reversing the flow direction is given by the parameter ρ_{DR} to shift from forward to backward movement, and by ρ_{RD} if the flow changes from back to front direction. Therefore, the setup time ST_k between two consecutive runs $(k-1, k) \in K$ at the same terminal is determined by constraints (27)–(28), while the setup time for flow reversals is given by restriction (29)

$$ST_k \geq \tau_{p,p'}(wd_{k-1,p} + wd_{k,p'} - 1), \quad \forall k > 1; p, p' \in P_D; p \neq p' \tag{27}$$

$$ST_k \geq \tau_{p,p'}(wr_{k-1,p} + wr_{k,p'} - 1), \quad \forall k > 1; p, p' \in P_R; p \neq p' \tag{28}$$

$$ST_k \geq \rho_{DR} rr_k + \rho_{RD} rd_k, \quad \forall k > 1 \tag{29}$$

4.5. Batch tracking constraints

4.5.1. Tracking the batch size over time

Batch tracking constraints similar to those proposed for unidirectional pipelines in continuous-time formulations can be applied to follow changes in the size of every in-transit batch. Let us use the continuous variable $W_i^{(k)}$ to represent the size of lot i at the completion of run k (i.e. at time C_k). The size of batch i can be increased during run k by the injection of an additional volume $Q_i^{(k)}$ coming from an input node, and reduced by diverting portions of it into some receiving terminals $j \in J$ ($\sum_{j \in J} D_{i,j}^{(k)}$). Terminal j_D pumps direct lots $i \in ID$ containing products $p \in P_D$, while source j_R injects products $p \in P_R$ transported by the reverse batches $i' \in IR$. In turn, all the pipeline terminals can be the destinations of old and new batches moving along the line. Then, Eq. (30) permits to follow the change in the batch size with time

$$W_i^{(k)} = W_i^{(k-1)} + Q_i^{(k)} - \sum_{j \in J} D_{i,j}^{(k)}, \quad \forall i \in I, k \in K \tag{30}$$

For lots $i \in I^{old}$ already in the pipeline at time $t=0$, the variable $W_i^{(k-1)}$ is equal to wo_i, a known datum representing the size of the old batch i at the start of the first pumping operation $k=1$.

Moreover, the input nodes (j_D, j_R) cannot receive a product flow from the pipeline while they are injecting a new batch. That condition is imposed through constraints (31) and (32)

$$\sum_{i \in I} D_{i,j_D}^{(k)} \le d_{\max} \left(1 - \sum_{i' \in ID} u_{i'}^{(k)}\right), \quad \forall k \in K \tag{31}$$

$$\sum_{i \in I} D_{i,j_R}^{(k)} \le d_{\max} \left(1 - \sum_{i' \in IR} v_{i'}^{(k)}\right), \quad \forall k \in K \tag{32}$$

4.5.2. The fluid incompressibility assumption

Refined products transported by reversible pipelines are assumed to be incompressible liquids. Hence, every time a new volume enters the pipeline from an input node, an equivalent amount must be discharged into some receiving terminals

$$\sum_{i \in ID} Q_i^{(k)} + \sum_{i' \in IR} Q_{i'}^{(k)} = \sum_{j \in J} \sum_{i \in I} D_{i,j}^{(k)}, \quad \forall k \in K \tag{33}$$

Depending on the flow direction, at most a single term in one of the LHS summations of Eq. (33) takes a positive value for any run k. The remaining ones vanish.

4.5.3. Monitoring batch locations in the pipeline over time

Let us introduce the variable $F_i^{(k)}$ standing for the frontal coordinate of batch $i \in I$ in the pipeline at the end of run k. Then, the relationship between the frontal coordinates of two consecutive lots i and $(i+1)$ at time C_k is given by Eq. (34)

$$F_i^{(k)} - W_i^{(k)} = F_{i+1}^{(k)}, \quad \forall i \in I, k \in K \tag{34}$$

Let us resume the simple example shown in Fig. 3. The injection of the filler batch $i3$ in direct flow pushes the whole content of batches $i1$ and $i2$ into terminal j_R during run $k2$. Then $W_{i1}^{(k2)} = W_{i2}^{(k2)} = 0$, $W_{i3}^{(k2)} = pv$, and from Eq. (34) it follows that $F_{i1}^{(k2)} = F_{i2}^{(k2)} = F_{i3}^{(k2)} = pv$ at time C_{k2}. According to the proposed mathematical formulation, batches $i1$ and $i2$ do still exist with a zero content, featuring front/back coordinates equal to pv at the completion of run $k2$. The next run $k3$ injects batch io from source j_R in the reverse direction. Then, $F_{i3}^{(k3)} = pv - W_{io}^{(k3)} = F_{i2}^{(k3)} = F_{i1}^{(k3)}$. Let us assume that a new run $k4$ fully ejects the batch $i3$ from the line into terminal j_D. As a result, batches $i1$, $i2$ and $i3$ will feature a null content and their front/back coordinates are equal to: $F_{i1}^{(k4)} = F_{i2}^{(k4)} = F_{i3}^{(k4)} = 0$ at the end of injection $k4$.

4.6. Feasibility conditions for batch injections and product deliveries to terminals

4.6.1. Feasibility conditions to divert all or part of a batch to a receiving terminal

Let $x_{i,j}^{(k)}$ be a binary variable denoting that some amount of product is diverted from batch $i \in I$ to terminal j during run k whenever $x_{i,j}^{(k)} = 1$. Otherwise, $x_{i,j}^{(k)} = 0$ and no product delivery from batch i to terminal j occurs at injection k, as stated by constraint (35)

$$d_{\min} x_{i,j}^{(k)} \le D_{i,j}^{(k)} \le d_{\max} x_{i,j}^{(k)}, \quad \forall i \in I, k \in K, j \in J \tag{35}$$

However, batch i should be in a proper location to divert its product into terminal j during run k. Some feasibility conditions should be satisfied when $x_{i,j}^{(k)} = 1$. The mathematical expressions of such feasibility conditions change with the flow direction. If batch i moves in direct flow, the feasibility constraints are given by inequalities (36) and (37). They pose that the product delivery from batch i to terminal j is feasible only if: (a) the upper coordinate $F_i^{(k)}$ of batch i has surpassed the location of terminal $j \neq j_D$ at the end of run k, and (b) its lower coordinate at the end of the previous run $k-1$ $(F_i^{(k-1)} - W_i^{(k-1)})$ is lesser than the terminal coordinate σ_j by at least the volume transferred to receiving terminals $j' \le j$ $(j' \neq j_D)$ during run k. The latter condition is relaxed by the volume $Q_i^{(k)}$ only for the lot $i \in ID$ being injected in direct flow at terminal j_D over run k. This is so because for that batch it holds: $F_i^{(k-1)} - W_i^{(k-1)} = 0$, and the feasibility condition (37) is surely satisfied

$$F_i^{(k)} \ge \sigma_j \left(\sum_{i' \in ID} u_{i'}^{(k)} + x_{i,j}^{(k)} - 1\right), \quad \forall i \in I, k \in K, j \in J, j \neq j_D \tag{36}$$

$$\begin{aligned} &F_i^{(k-1)} - W_i^{(k-1)} + \sum_{j' \in J, j_D < j' \le j} D_{i,j'}^{(k)} \\ &\quad \le \sigma_j + (pv - \sigma_j)\left(2 - \sum_{i' \in ID} u_{i'}^{(k)} - x_{i,j}^{(k)}\right) + Q_i^{(k)}, \\ &\quad \forall i \in I, k \in K, j \in J, j \neq j_D \end{aligned} \tag{37}$$

On the other hand, batch i moving in the opposite direction can transfer some quantity of product to terminal $j \neq j_R$ only if constraints (38) and (39) are fulfilled. They control that: (a') the lower coordinate of batch i $(F_i^{(k)} - W_i^{(k)})$ has surpassed the location of terminal j at the end of run k, and (b') its upper coordinate $F_i^{(k-1)}$ at the end of run $k-1$ is higher than σ_j by at least the volume transferred to depots $j' \ge j$ $(j' \neq j_R)$ during run k, unless batch $i \in IR$ is the one currently pumped into the line

$$\begin{aligned} &F_i^{(k)} - W_i^{(k)} \le \sigma_j + (pv - \sigma_j)\left(2 - \sum_{i' \in IR} v_i^{(k)} - x_{i,j}^{(k)}\right), \\ &\quad \forall i \in I, k \in K, j \in J, j \neq j_R \end{aligned} \tag{38}$$

$$\begin{aligned} &F_i^{(k-1)} - \sum_{j' \in J, j \le j' < j_R} D_{i,j'}^{(k)} \ge \sigma_j \left(\sum_{i' \in IR} v_{i'}^{(k)} + x_{i,j}^{(k)} - 1\right) - Q_i^{(k)}, \\ &\quad \forall i \in I, k \in K, j \in J, j \neq j_R \end{aligned} \tag{39}$$

4.6.2. Feasibility conditions to inject a batch into the pipeline

If the direct batch $i \in ID$ is pumped into the pipeline at source j_D during run k $(u_i^{(k)} = 1)$, its lower coordinate should remain at the pipeline origin over that injection. Otherwise, it cannot receive the flow of product from terminal j_D. In other words, $F_i^{(k)} - W_i^{(k)} = \sigma_{jD} = 0$ at the completion time C_k. Analogously, the frontal coordinate of lot $i \in IR$ injected from the reverse source j_R should be positioned at the other extreme $(F_i^{(k)} = \sigma_{jR} = pv)$ while it is being pumped. These feasibility constraints for batch injections are given by inequalities (40)–(41)

$$F_i^{(k)} - W_i^{(k)} \le pv(1 - u_i^{(k)}), \quad \forall i \in ID, k \in K \tag{40}$$

$$F_i^{(k)} \ge pv \quad v_i^{(k)}, \quad \forall i \in IR, k \in K \tag{41}$$

Revisiting the simple example illustrated in Fig. 3, an interesting conclusion can be derived. If a new run $k4$ fully ejects the filler batch $i3$ from the line into terminal j_D, then the batches $i1$, $i2$ and $i3$ will have a null size and front/back coordinates equal to zero at the end of $k4$. As they belong to the set ID and are positioned at the pipeline origin when run $k4$ is completed, those lots can be inserted again in direct flow from source j_D during runs $k > k4$. However, each of the three lots has a product already assigned. If injected again, they should contain the same product previously allocated ($p1$ for $i1$, $p2$ for $i2$, and so on). A similar statement applies to new batches $i \in IR$ injected at source j_R and fully delivered to terminals $j < j_R$. When the flow direction is inverted, they comeback to terminal j_R with a null size and lower/upper coordinates equal to pv. In this way, the proposed number of new batches in the sets ID and IR, and consequently both the model size and the computational burden are substantially diminished. This is an interesting feature of the proposed mathematical formulation.

4.7. Product inventories in terminal tanks

4.7.1. Fulfillment of customers' demands from pipeline terminals

Once the oil refined products shipped through the pipeline have been discharged into the receiving depots, they must be delivered to nearby customers to fulfill their product demands before the end of the time horizon. The total demand of product p to be covered by terminal j during the current horizon is a known datum given by $dem_{p,j}$. Moreover, the delivery rate of product p at terminal j is restricted by the terminal capacity for dispatching barges, trucks and/or trains, given by the parameter $dr_{p,j}$. As a result, an upper bound on the total volume of product p dispatched from terminal j to nearby customers at run k, represented by the continuous variable $DM_{p,j}^{(k)}$, is given by constraint (42). It is the amount of p that can be delivered from terminal j to customers during the time interval $[C_{k-1}, C_k]$

$$DM_{p,j}^{(k)} \le dr_{p,j}(C_k - C_{k-1}), \quad \forall k \in K, p \in P, j \in J \tag{42}$$

It may happen that some market demand of product p supplied by terminal j cannot be satisfied within the planning horizon. To account for these potential events, product backorders will be allowed. To do so, the continuous variable $B_{p,j}$ is introduced in Eq. (43) representing the unsatisfied demand of product p at terminal j. Such product backorders will have a unit penalty cost $bc_{p,j}$ in the objective function

$$\sum_{k \in K} DM_{p,j}^{(k)} = dem_{p,j} + B_{p,j}, \quad \forall p \in P, j \in J \tag{43}$$

4.7.2. Amount of product p supplied by in-transit batches to terminals during run k

Let $DP_{i,j,p}^{(k)}$ be a continuous variable denoting the amount of product p diverted from the flowing batch i to the receiving terminal j during run k. If batch i contains product p (i.e., $y_{i,p} = 1$), then $DP_{i,j,p}^{(k)} = D_{i,j}^{(k)}$. Otherwise, $DP_{i,j,p}^{(k)} = 0$. Constraints on the value of $DP_{i,j,p}^{(k)}$ are given by Eqs. (44) and (45)

$$DP_{i,j,p}^{(k)} \le d_{\max}\, y_{i,p}, \quad \forall i \in I, k \in K, p \in P, j \in J \tag{44}$$

$$\sum_{p \in P} DP_{i,j,p}^{(k)} = D_{i,j}^{(k)} \quad \forall i \in I, k \in K, j \in J \tag{45}$$

4.7.3. Amount of product p injected in the pipeline from an input terminal during run k

The volume of product p in batch i pumped into the pipeline from an input terminal j during run k is computed through the variable $QP_{i,p}^{(k)}$. If batch i contains product p (i.e., $y_{i,p} = 1$), then $QP_{i,p}^{(k)} = Q_i^{(k)}$. Otherwise, $QP_{i,p}^{(k)} = 0$. As the pipeline system comprises two input terminals (j_D and j_R), and each source can pump lots belonging to different sets (ID and IR) containing distinct groups of products (P_D and P_R), source-dependent equations are introduced in the model to define the value of $QP_{i,p}^{(k)}$. They are given by Eqs. (46)–(47) and (48)–(49) for direct and reverse operations, respectively

$$QP_{i,p}^{(k)} \le q_{\max}^{D} y_{i,p}, \quad \forall i \in ID, k \in K, p \in P_D \tag{46}$$

$$\sum_{p \in P_D} QP_{i,p}^{(k)} = Q_i^{(k)}, \quad \forall i \in ID, k \in K \tag{47}$$

$$QP_{i,p}^{(k)} \le q_{\max}^{R} y_{i,p}, \quad \forall i \in IR, k \in K, p \in P_R \tag{48}$$

$$\sum_{p \in P_R} QP_{i,p}^{(k)} = Q_i^{(k)}, \quad \forall i \in IR, k \in K \tag{49}$$

4.7.4. Product inventory constraints to avoid tank overloading and empty conditions

The tank farms of input terminals j_D and j_R are usually replenished either by directly receiving the production output of neighboring refineries and/or by discharging the content of oil tankers coming from farther refiners. In any case, we assume that product storages are replenished at an overall constant rate $fr_{p,j}$, without interruption all along the planning horizon. Hence, the amount of product p received by terminal j from external sources at run k is given by: $fr_{p,j}(C_k - C_{k-1})$.

During the interval $[C_{k-1}, C_k]$, the inventory of product p in the tank farm of terminal j can increase by receiving new product supplies from external sources, and portions of batches of product p flowing through the line. Besides, it can be diminished because of the injection of new batches, and the dispatch of products to satisfy customers' demands. Let $ID_{p,j}^{(k)}$ be a continuous variable denoting the inventory level of product p in terminal j at time C_k. Eqs. (50)–(52) determine the value of $ID_{p,j}^{(k)}$ for terminals j_D and j_R, and the intermediate pure-receiving terminals $j_D < j < j_R$

$$ID_{p,j}^{(k)} = ID_{p,j}^{(k-1)} + fr_{p,j}(C_k - C_{k-1}) + \sum_{i \in I} DP_{i,j,p}^{(k)} - \sum_{i \in ID} QP_{i,p}^{(k)} - DM_{p,j}^{(k)}, \quad \forall k \in K, p \in P, j = j_D \tag{50}$$

$$ID_{p,j}^{(k)} = ID_{p,j}^{(k-1)} + fr_{p,j}(C_k - C_{k-1}) + \sum_{i \in I} DP_{i,j,p}^{(k)} - \sum_{i \in IR} QP_{i,p}^{(k)} - DM_{p,j}^{(k)}, \quad \forall k \in K, p \in P, j = j_R \tag{51}$$

$$ID_{p,j}^{(k)} = ID_{p,j}^{(k-1)} + \sum_{i \in I} DP_{i,j,p}^{(k)} - DM_{p,j}^{(k)}, \quad \forall k \in K, p \in P, j \in J, j_D < j < j_R \tag{52}$$

To avoid tank overloading and empty conditions, the inventory level at the end of any run k must satisfy the constraints (53). In other words, the value of $ID_{p,j}^{(k)}$ should belong to the feasible range given by $[id_{p,j}^{\min}, id_{p,j}^{\max}]$

$$id_{p,j}^{\min} \le ID_{p,j}^{(k)} \le id_{p,j}^{\max}, \quad \forall k \in K, p \in P, j \in J \tag{53}$$

4.8. Operating the pipeline during on-peak energy periods

See Appendix.

5. Objective function

The selected problem goal is to minimize the pipeline operation cost including:

(a) *Pumping costs*. These expenses are assumed to be directly proportional to the pumped volume, depending on the product injected and the flow direction. The unit pumping cost for product p injected at the input terminal j is given by the parameter $pc_{p,j}$.

(b) *Inventory carrying costs*. This term of the objective function is estimated by computing the average inventory level of each product in every terminal $j \in J$ carried over the time horizon. Since the model handles a continuous-time domain, the average inventory level of p at terminal j is approximated by: $\sum_{k\in K} ID_{p,j}^{(k)} / |K|$.

In other words, the average inventory level of product p in terminal j is given by the sum of the product inventory levels at the end of every pumping run, divided by the number of runs. If no dummy run appears in the optimal solution, the proposed expression provides a very good estimation of the average inventory level. The total inventory carrying cost is obtained by: (i) multiplying the estimated average inventory level of product p in terminal j by the cost of carrying a single unit of product p in that terminal over the planning horizon ($ic_{p,j}$), and (ii) summing up inventory costs over all products and all depots.

(c) *Interface costs*. These costs are due to product contamination at the interface of adjacent batches, and are obviously dependent on the products they contain (namely p and p') and the contaminated volume. The unit reprocessing/degrading cost for the mix of products p and p' is given by $ifc_{p,p'}$.

(d) *Flow reversal costs*. Every time the pipeline flow is inverted, many pump and valve operations should be accomplished, thus yielding high additional costs. Moreover, direct-to-reverse and reverse-to-direct changes in flow direction may have different costs, given by c_{DR} and c_{RD}, respectively.

(e) *Backorder costs*. They are directly proportional to the product demands that are tardily satisfied beyond the planning horizon. The unit backorder cost for product p supplied by terminal j is given by the parameter $bc_{p,j}$.

(f) *Peak-hour electricity costs*. They are calculated by multiplying the total time (measured in hours) during which the pipeline is operated in daily peak periods ($TK_{k,e}$) by the unit cost pk.

(g) Finally, the problem objective function is given by Eq. (54)

$$\begin{aligned}\min z = & \sum_{p\in P_D} pc_{p,j_D}\left[\sum_{k\in K}\sum_{i\in ID} QP_{i,p}^{(k)}\right] + \sum_{p\in P_R} pc_{p,j_R}\left[\sum_{k\in K}\sum_{i\in IR} QP_{i,p}^{(k)}\right] \\ & + \sum_{j\in J}\sum_{p\in P}\frac{ic_{p,j}}{|K|}\left[\sum_{k\in K} ID_{p,j}^{(k)}\right] + \sum_{p\in P}\sum_{p'\in P, p'\neq p} ifc_{p,p'}\left[\sum_{k\in K} WIF_{p,p'}^{(k)}\right] \\ & + \sum_{k\in K}(c_{DR} rr_k + c_{RD} rd_k) + \sum_{j\in J}\sum_{p\in P} bc_{p,j}B_{p,j} + \sum_{k\in K}\sum_{e\in E} pkTK_{k,e}\end{aligned} \tag{54}$$

6. Results and discussion

Three examples have been solved using the proposed MILP formulation. The first example was introduced by Magatão et al. (2004) and later revisited by Magatão et al. (2011). It deals with the scheduling of a real-world bidirectional pipeline with a pair of input/output terminals at both ends, assuming that a single flow reversal can at most be performed over the planning horizon. The aim of Example 1 is to show the computational advantages of the continuous-time representation against discrete approaches. Example 2 is a variant of Example 1 with three additional features: (a) multiple flow reversals (as many as necessary) are allowed, (b) continuous supplies of products from oil refineries into the terminal tanks are considered, and (c) given product demands from nearby customers should be dispatched from the pipeline terminals before the end of the planning horizon. This example is introduced to illustrate other interesting features of the proposed model. In addition to its computational efficiency, it is capable of precisely determining the required number of pipeline flow reversals and the optimal time instants at which they should be performed to minimize operating costs. It also manages to get a perfect coordination between pumping runs, product supplies from refineries and dispatches of fuel tankers to customers in order to avoid tank overloading or inventories below the minimum level. Finally, Example 3 considers a reversible pipeline configuration that also includes an intermediate depot receiving lots of products from both extreme terminals, as depicted in Fig. 4. The presence of intermediate depots requires increasing the number of elements in the sets *ID* and *IR*, and consequently both the model size and the CPU time also increase. In all cases, the MILP models are solved to optimality on an Intel® Xeon® CPU (2.67 GHz) with GAMS/GUROBI 5.0.1 (Brooke, Kendrick, Meeraus, & Raman, 2006) as the MILP solver, using 6 parallel threads.

6.1. Example 1

Example 1 is concerned with the short-term operational planning of a real-world bidirectional pipeline connecting a harbor to an inland refinery. The pipeline has a length of 93.5 km and a total volume of 7314 m^3. It transports four products ($P1$–$P4$) from the refinery to the harbor (direct flow), and other four oil derivatives ($P5$–$P8$) in the opposite direction (reverse flow). In either flow direction, batches are pumped at a maximum rate of 300 (m^3/h) and travel for more than 24 h to reach the opposite extreme. The length of the planning horizon has been fixed at h_{max} = 144 h (six days). For operational reasons, the pipeline flow can be reversed only once over the planning horizon. Although the proposed continuous approach is able to manage any float value, the pipeline capacity has been fixed at 7200 m^3 to make a fair comparison against discrete representations like the ones developed by Magatão et al. (2004, 2011). To solve Example 1, such discrete formulations divide the planning horizon into time intervals of 6 h, and the pipeline volume into fix-sized packs of 1800 m^3. Because no physical gadget is used to separate successive batches, the generation of interfaces while injecting new lots into the line is unavoidable. In addition, some products should not be consecutively injected into the pipeline due to excessive contamination, thus requiring a "plug" (i.e., a small batch of a compatible product) to separate them. Pairs of non-

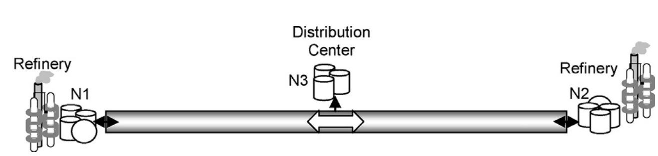

Fig. 4. Pipeline system studied in Example 3.

Table 1
Pairs of non-compatible products and terminal demands for Example 1.

	Succeding product								Demand (m^3)	
	*P*1	*P*2	*P*3	*P*4	*P*5	*P*6	*P*7	*P*8	N1	N2
*P*1		X		X		X		X	–	1800
*P*2	X				X				–	3600
*P*3				X				X	–	1800
*P*4	X		X		X		X		–	1800
*P*5		X		X		X		X	1800	–
*P*6	X				X				1800	–
*P*7				X				X	3600	–
*P*8	X		X		X		X		3600	–

compatible products requiring a separating plug are indicated with an “X” in Table 1. For instance, product *P*1 can only be pumped immediately after products *P*3, *P*5 or *P*7. Otherwise, it requires the insertion of a plug to prevent contamination with other products. One of the main objectives of the pipeline planner is to avoid plugs as much as possible because they increase both the number of interfaces and the operation cost. Table 1 also includes the demands for products *P*5–*P*8 and *P*1–*P*4 at terminals N1 and N2, respectively, to be fulfilled before the end of the planning horizon.

Furthermore, electricity costs jump by a factor of five during on-peak energy periods, between 6:00 pm and 9:00 pm every day. Hence, it is likely that the pipeline schedule includes idle periods during the time intervals 18–21 h, 42–45 h, 66–69 h, and so on. Nonetheless, unnecessary pipeline stops should be avoided because idle pipelines must be maintained pressurized and still require some operation cost. As adopted by Magatão et al. (2011), the cost of a separating plug, the pressurizing hourly cost during idle periods, and the pumping cost in off-peak hours are all set equal to one.

The problem goal is to find the optimal pipeline schedule that permits to exactly satisfy all products requirements over the six-day time horizon at minimum total cost, including pressurizing, pumping and plug costs. Because the maximum length of the planning horizon is 144 h, it is expected that the problem makespan takes a lower value by eliminating unnecessary idle periods. Pump rates can vary between 150 and 300 m^3/h, and the size of any batch injection should be neither smaller than 180 m^3 nor higher than 10,800 m^3 for operational reasons. As at most a single flow reversal is permitted, it has been chosen: $|ID| = |P_{N1}| = 4$, $|IR| = |P_{N2}| = 4$, and $|K| = 1.5 \cdot |P| = 1.5 \cdot 8 = 12$. The value of $|K|$ must be greater than $|P|$ because it may be required more than a single pump run to inject some products due to pipeline stoppages during on-peak energy periods. The factor 1.5 comes from the fact that half of the batch injections are expected, on average, to be interrupted in order to avoid energy consumption during on-peak energy periods. However, this value is just an initial guess for $|K|$. This number will be recursively increased by one until the optimal solution shows no further change. The influence of the value of $|K|$ in the solution quality and the computational performance of the model is analyzed in Section 6.3.

One of the optimal solutions for Example 1 is presented in Fig. 5. It consists of 5 product injections at node N2 ($P8^{3600}$ – $P6^{1800}$ – $P7^{3600}$ – $P5^{2700}$ – $P5^{6300}$) in reverse flow, followed by 6 pumping runs at source N1 ($P1^{1800}$ – $P3^{1800}$ – $P2^{2700}$ – $P2^{900}$ – $P4^{5400}$ – $P4^{3600}$). Superscripts stand for injection sizes in m^3. Overall, 11 pumping operations are performed over a makespan of 129 h and the pipeline

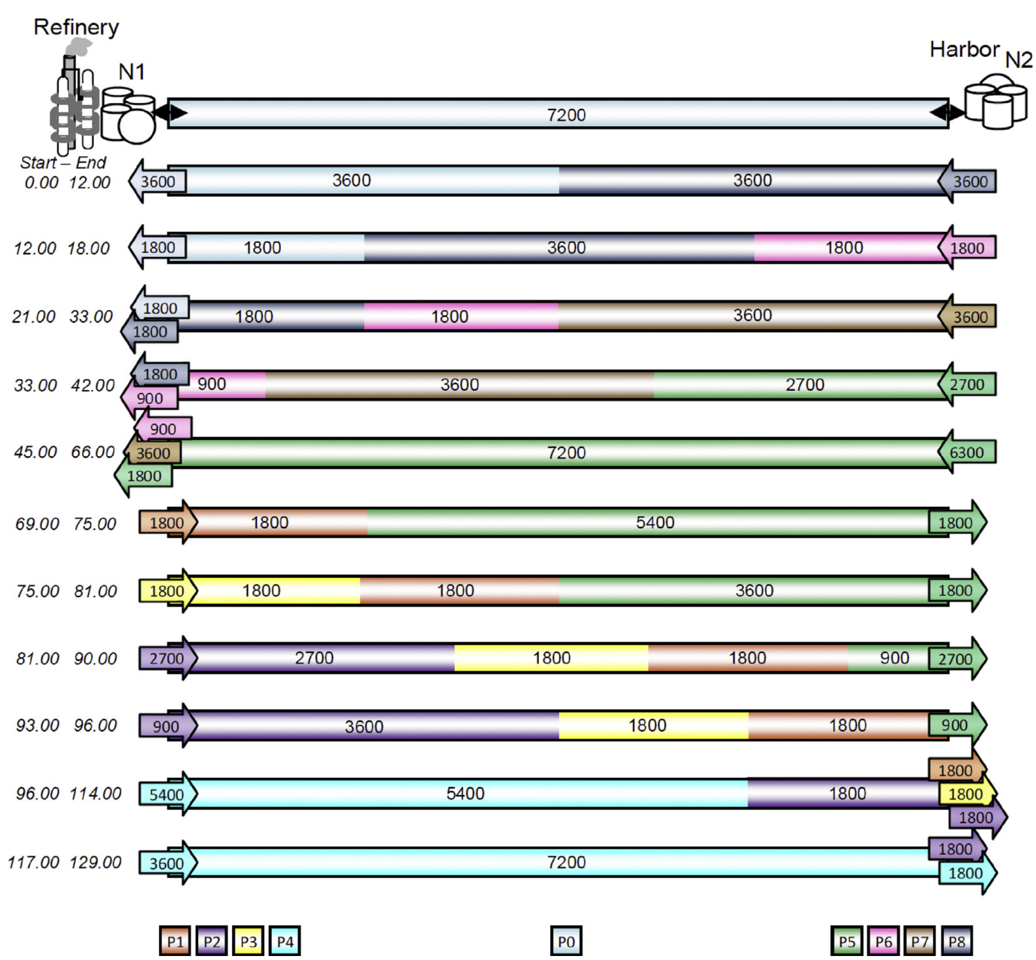

Fig. 5. Optimal pipeline operation for Example 1 (reverse flow first).

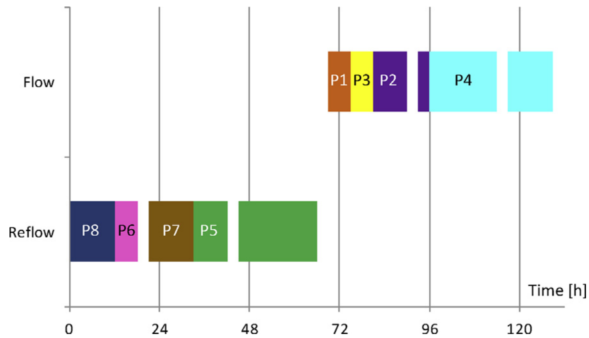

Fig. 6. Optimal sequence of pumping runs for Example 1 (reverse flow first).

is stopped along every on-peak energy period (see the Gantt chart in Fig. 6). Two consecutive pumping runs are needed for the injection of products *P*2, *P*4 and *P*5. Although many pairs of products are incompatible as shown in Table 1, no plug insertion is required since the products involved in consecutive injections are properly selected.

Contrarily to discrete approaches requiring an iterative procedure for testing different horizon lengths (Magatão et al., 2004, 2011), the proposed approach finds the best schedule and the optimal makespan in a single step. Magatão et al. (2011) reported a total CPU time of 894.7 s for testing 31 integer values of the planning horizon (from 114 h to 144 h), using a combined CLP–MILP approach. In our case, the optimal solution is found and guaranteed in 263.6 CPUs by solving a monolithic MILP model comprising 5737 equations, 2640 continuous variables, and 715 binaries. Computational results are summarized in Table 2.

An alternative optimal solution is obtained by starting the pipeline operation in direct flow, i.e. by making $\sum_{i \in ID} u_i^{(k1)} = 1$. However, two more pumping operations are needed, while the number of discrete variables and the solution time both increase to 827 and 582.3 s, respectively. Six injections are planned from node N1 ($P1^{1800} - P3^{1800} - P2^{1800} - P2^{1800} - P4^{4500} - P4^{4500}$) followed by seven pumping runs in the opposite direction ($P8^{1800} - P8^{1800} - P6^{1800} - P7^{2700} - P7^{900} - P5^{5400} - P5^{3600}$).

Another test for the model validation consists in slightly modifying product demands. It is considered a demand increase of 10% for products *P*5, *P*6, *P*7 and *P*8 at node N1, and 10% reductions in the requests for products *P*1, *P*2, *P*3 and *P*4 at node N2. As expected, the model size and the computational performance of our continuous-time scheduling approach for reversible pipelines do not experience major changes. A total of 13 operations are required and the best solution is found in 45 CPU s, whereas its optimality is guaranteed in 1378.8 CPU s. The new optimal solution involves a total of 7 product injections at node N2 ($P8^{3960} - P6^{1440} - P6^{540} - P7^{3960} - P5^{1800} - P5^{6300} - P5^{1080}$) in reverse flow, followed by 6 pumping runs at source N1 ($P1^{1620} - P3^{1620} - P2^{1980} - P2^{1260} - P4^{5040} - P4^{3780}$). Note that the number of injections at N2 increases from 5 to 7 due to the energy-peak periods forcing to partitioning both the pumping of *P*6 into 2 runs, and the injection of *P*5 into 3 pumping runs instead of 2. As in previous schedules, the pipeline is stopped along every on-peak energy period, while the makespan increases from 129 to 129.6 h since the overall demand is slightly larger. If the new demand scenario is solved using a discrete model, the size of the formulation increases by a factor of ten to exactly obtain the same optimal solution.

6.2. Example 2

The second case study involves a 14-in. reversible pipeline having an approximate length of 100 km and a total capacity of 10,000 m^3. It connects the tank farms of two oil refineries. Pipeline operations should be optimally scheduled over a two-week planning horizon (336 h). Example 2 is introduced to show how the continuous formulation can help pipeline operators to efficiently coordinate the pumping schedule at the input nodes with the refinery production plan. Given the production schedules at the oil refineries, the proposed approach provides the number and timing of flow reversals to meet customers' demands at minimum total cost. One of the refineries supplies products *P*1 and *P*2 to the input node N1 at constant rates of 100 and 150 m^3 per hour, while the other facility provides products *P*3 and *P*4 around the clock to node N2 at fixed rates of 50 and 80 m^3/h, respectively.

Production flows coming from refineries are directly discharged into the tank farms of terminals N1 and N2, whose maximum/minimum inventory levels are shown in Table 3. Besides, Table 3 also reports the initial stocks of products in terminal tanks, the customers' demands to be covered from each terminal before the end of the time horizon, and the inventory carrying cost for every unit of product stored in nodes N1 and N2 above the minimum level over the planning horizon.

Moreover, both terminals can deliver products *P*1, *P*2, *P*3, and *P*4 to neighboring customers at maximum rates of 200, 250, 100 and 150 (m^3/h), respectively. In addition to timely fulfilling customer orders from both sources, the pipeline operational plan should be properly designed to prevent from either tank overloading or an inventory level below the minimum. On the other hand, the pumping rate in direct flow (from N1 to N2) should be neither lower than 100 m^3/h nor higher than 600 m^3/h, whereas the admissible pumping rate in reverse flow must remain between 50 and 400 m^3/h. Finally, unit pumping costs and interface reprocessing charges for every ordered pair of products are given in Table 4.

The optimal solution for the proposed case study was found in 5.51 CPU s and is presented in Fig. 7. Volumes of products supplied from refineries to nodes N1 and N2 during every injection are reported inside the colored circles at both extremes of the pipeline. Further computational results for this example are reported in Table 5.

As shown in the first line of Fig. 7, the pipeline is initially filled with product *P*1. At the optimal operational plan, the pipeline starts working in direct flow by pumping 140 additional units of product *P*1 during the first 48.21 h (i.e. approximately 2 days) at a rate of 290 (m^3/h). The first injection pushes 140 units of *P*1 into the tanks of terminal N2 and the pipeline remains full of product *P*1 at time *t* = 48.21 h. In parallel with this pumping run, the following shipments of products from the terminals to the customers are made:

a. Products $P1^{96.4}$ and $P2^{100}$ (superscripts indicate the volume, in 10^2 m^3) are dispatched from N2 to nearby customers at rates of 200 and 207 m^3/h, respectively, to fulfilling customer demands and, at the same time, cutting down inventory carrying costs. In

Table 2
Computational results for Example 1.

	Binary variables	Cont. variables	Equations	CPU time (s)	Optimal solution	Direct batches $\|ID\|$	Reverse batches $\|IR\|$	Number of runs $\|K\|$
Example 1	715	2640	5737	263.6	129	4	4	11

Table 3
Inventory levels, demands and inventory carrying costs for Example 2.

	N1				N2			
	Initial 10^2 m^3	Max/min 10^2 m^3	Dem 10^2 m^3	Inv cost USD/m^3	Initial 10^2 m^3	Max/min 10^2 m^3	Dem 10^2 m^3	Inv cost USD/m^3
*P*1	150	200/50	180	0.19	150	200/50	340	0.19
*P*2	260	300/60	200	0.16	160	300/60	400	0.16
*P*3	100	150/50	230	0.24	100	150/50	–	0.24
*P*4	150	250/50	360	0.24	150	250/50	–	0.24

Table 4
Unit Pumping and Interface Costs for Example 2.

	Interface costs (USD) Successive product				Pumping costs (USD/m^3) Injection node	
	*P*1	*P*2	*P*3	*P*4	N1	N2
*P*1	–	18,400	34,000	23,500	0.70	–
*P*2	18,400	–	25,000	41,300	0.60	–
*P*3	34,000	25,000	–	30,000	–	1.00
*P*4	23,500	41,300	30,000	–	–	0.60

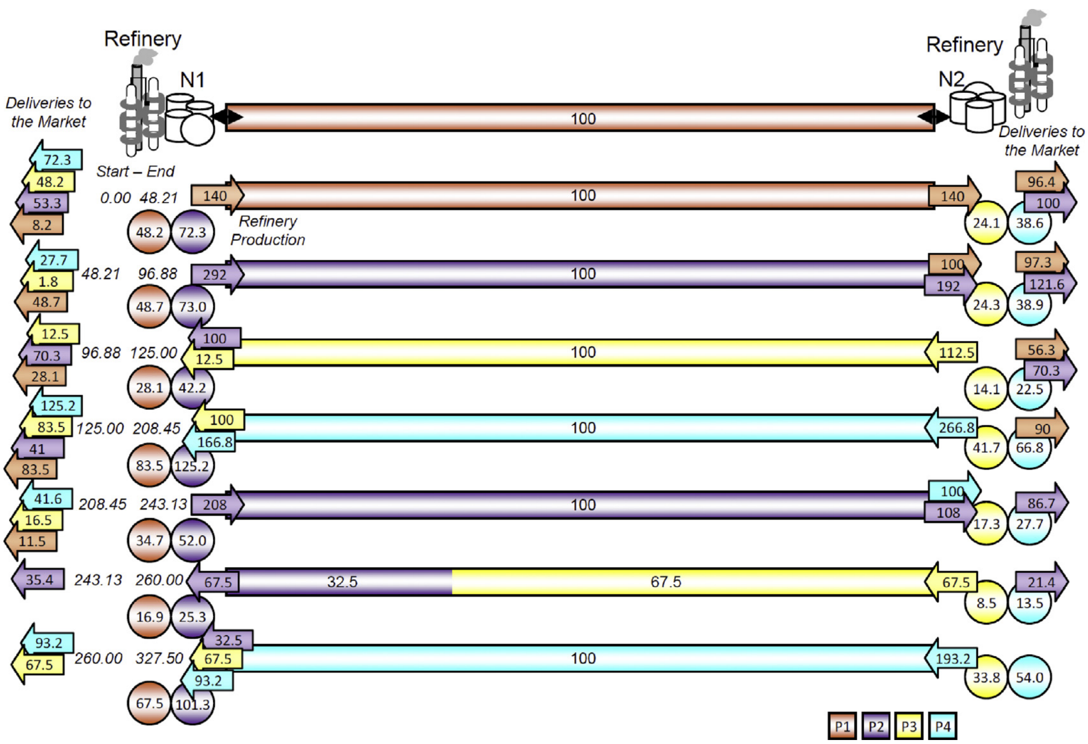

Fig. 7. Optimal schedule of pipeline operations for Example 2.

fact, the inventory of *P*2 at node N2 drops to the minimum level at $t = 48.21$ h.

b. Products $P1^{8.2}$, $P2^{53.3}$, $P3^{48.2}$ and $P4^{72.3}$ are delivered from source N1 to customer locations.

At $t = 48.21$ h, it begins the injection of product $P2^{292}$ at node N1 in forward direction to deliver lots of products $P1^{100}$ and $P2^{192}$ to node N2. At the same time, tankers containing $[P1^{48.7}, P3^{1.8}, P4^{27.7}]$ and $[P1^{97.3}, P2^{121.6}]$ are shipped to customers from terminals N1 and N2, respectively. By analyzing the product inventory profiles shown in Fig. 8, it is observed that almost all product availabilities at node N1 decrease to their minimum levels at the end of the second run, i.e. at $t = 96.88$ h. Then, it is time to reverse the flow. From $t = 96.88$ h to 125.00 h, product $P3^{112.5}$ is pumped from N2 to N1 at the maximum admissible rate of 400 (m^3/h), thus forcing the pipeline linefill containing exclusively $P2^{100}$ to comeback to terminal N1. Next, product $P4^{266.8}$ is pumped in the same direction at a flow rate of 319 (m^3/h) until the inventory of *P*4 at node N2 drops to its minimum level, that is at $t = 208.45$ h. At that time and once again, the flow is inverted to pump product $P2^{208}$ at N1 into the

Table 5
Computational results for Examples 2 and 3.

	Binary variables	Cont. variables	Equations	CPU time (s)	Optimal solution	Direct batches |*ID*|	Reverse batches |*IR*|	Number of runs |*K*|
Example 2	160	521	1128	5.51	264.41	2	2	7
Example 3	273	875	1920	171	248.53	3	3	7

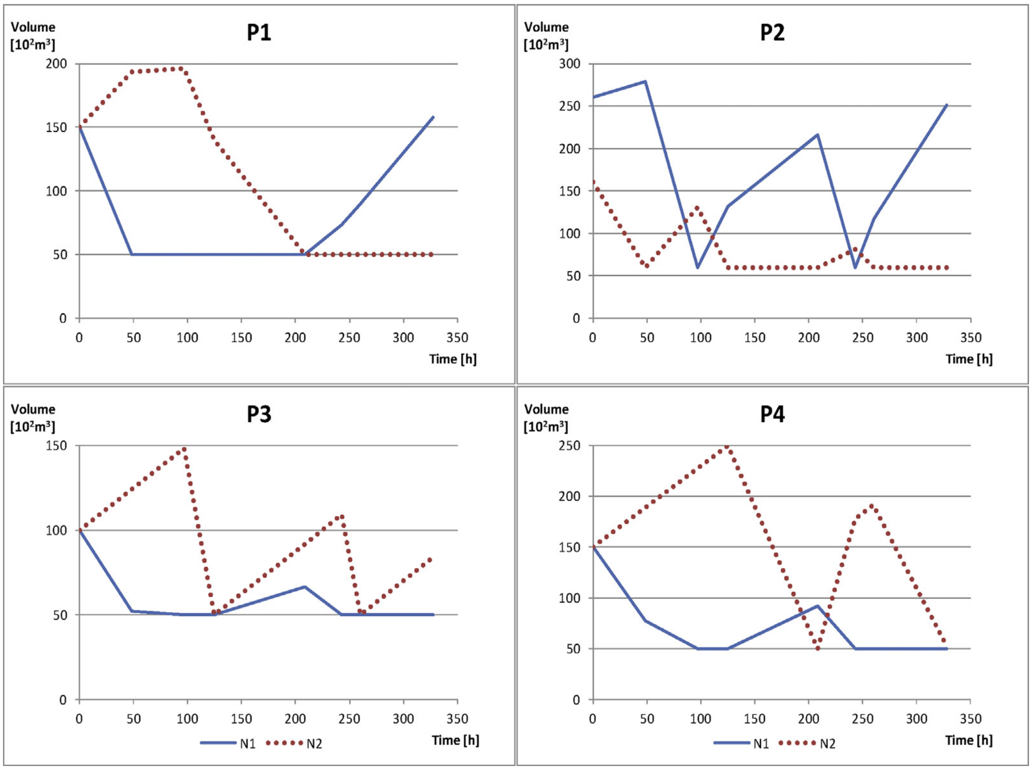

Fig. 8. Product inventory profiles at terminals N1 and N2 of Example 2.

line, pushing the pipeline linefill consisting of $P4^{100}$ into terminal N2.

Overall, the optimal solution defines a pumping flow pattern given by: $P1_D$ – $P2_D$ – $P3_R$ – $P4_R$ – $P2_D$ – $P3_R$ – $P4_R$ with three flow reversals. Subscripts "D" and "R" indicate direct and reverse flows. Two lots of $P2_D$ and a single batch of $P4_R$ pumped into the pipeline immediately before a flow reversal play the role of filler lots in direct and reverse flow, respectively. When the flow is inverted, the filler product returns to its original tank and produces, together with the constant refinery supply, a remarkable increase in the product inventory level. When these events occur, the inventory profile presents a high positive slope. This is the case for product *P*4 at terminal N2 from $t = 208.45$ h to $t = 243.13$ h (see Fig. 8).

Similar to previous approaches (Cafaro & Cerdá, 2004, 2012) it is assumed that some safety stock given by the minimum inventory level is kept at any terminal to temporarily cover the time delay in the arrival of a batch. Such condition may briefly arise throughout a run discharging more than one batch. For instance, the available stock of *P*2 above the minimum level at N2 is null at $t = 48.21$ and the new batch of *P*2 replenishing tanks of N2 will arrive at $t = 48.21 + 100/6 = 64.88$ h, i.e. after the batch of *P*1 is totally discharged at N2. However, the delivery of *P*2 from N2 to nearby customers should start earlier using the safety stock to respect the dispatching rate. For sizing the use of the safety stock, simple constraints can be added to the formulation in order to control the inventory levels at the earliest arrival times of the batches. These constraints are presented in Appendix.

As already mentioned, the most interesting feature of the proposed plan is the precise coordination between refinery production flows, product deliveries and batch movements in the line by effectively managing inventory levels at tank farms. For instance, the stock of product *P*3 in terminal N2 reaches its maximum admissible level at $t = 96.88$ h, but at that time the injection of *P*3 into the pipeline starts and the stock is rapidly lowered to the minimum level. Moreover, the stock of *P*4 at N2 is also in its upper bound when the pumping of *P*3 has finished ($t = 125$ h), thus being imperative to begin with the injection of *P*4 into the pipeline. On the other hand, product deliveries to customers show remarkable features. In particular, the last delivery of product *P*2 from node N1 to the market is delayed until time $t = 243.13$ h for a simple reason. Product *P*2 is used as filler in direct flow and there will be no enough amount of *P*2 to make the linefill batches reach the opposite extreme if some dispatches of *P*2 to the customers are performed earlier.

6.3. Computational sensitivity analysis

With the aim of studying the model computational performance, this section presents the results achieved for 21 instances of the problem formulation of Example 2, when increasing numbers of batches and runs are adopted. Results are summarized in Table 6.

In the first instance (Case 1), the set *ID* includes two elements standing for batches to be pumped in direct flow, two elements are posed in the set *IR*, and the number of pumping runs $|K|$ is equal to 6. Because the number of batch injections is lower than required, it is not possible to achieve a pipeline schedule that fully satisfies all terminal requirements within the planning horizon. Backorders are not avoided by increasing the number of batches in the sets *ID* and *IR* (see Cases 2–7). However, by just adding one more element to the set *K* the optimal solution is found at once (Case 8). Further increases in the value of $|K|$ produce no further improvement in the objective function (see Cases 9–21). As already discussed in Section 2.2, the required cardinalities of sets *ID*, *IR* and *K* are not known beforehand and must be estimated before solving the problem. Good estimations used to initially solve this example are given by: $|ID| = |P_{N1}| = 2$, $|IR| = |P_{N2}| = 2$, $|K| = \alpha_{dir}\ |P_{N1}| + \alpha_{rev}\ |P_{N2}| = 2{\cdot}2 + 2{\cdot}2 = 8$. Parameters α_{dir} and α_{rev} stand for the estimated number of times that the pipeline should operate in direct and

Table 6
Effect of the number of batches and runs on the computational performance.

Case	$\lvert ID\rvert$	$\lvert IR\rvert$	$\lvert K\rvert$	Eq.	CVar	BVar	Obj. Funct. (10^3 USD)	$\sum_{p\in P}\sum_{j\in J}B_{p,j}$ (10^2 m^3)	CPU time (s)
1	2	2	6	968	588	140	913.32	67.5	3.16
2	2	3	6	1126	682	168	913.32	67.5	3.49
3	3	2	6	1126	682	168	730.10	48.2	4.70
4	3	3	6	1286	776	196	730.10	48.2	6.72
5	3	4	6	1442	870	224	730.10	48.2	5.70
6	4	3	6	1442	870	224	655.58	41.3	10.47
7	4	4	6	1600	964	252	655.58	41.3	8.61
8[a]	2	2	7	1128	681	160	264.41	0.0	5.51
9	2	3	7	1312	790	192	264.41	0.0	7.27
10	3	2	7	1312	790	192	264.41	0.0	15.83
11	3	3	7	1496	899	224	264.41	0.0	24.03
12	3	4	7	1680	1008	256	264.41	0.0	28.62
13	4	3	7	1680	1008	256	264.41	0.0	82.75
14	4	4	7	1864	1117	288	264.41	0.0	41.12
15[b]	2	2	8	1288	774	180	264.41	0.0	14.93
16	2	3	8	1498	898	216	264.41	0.0	17.48
17	3	2	8	1498	898	216	264.41	0.0	38.63
18	3	3	8	1708	1022	252	264.41	0.0	168.63
19	3	4	8	1918	1146	288	264.41	0.0	114.66
20	4	3	8	1918	1146	288	264.41	0.0	300.30
21	4	4	8	2128	1270	324	264.41	0.0	310.30

[a] Optimal solution found.
[b] Initial trial proposed.

reverse flow, respectively, to fulfill all terminal requirements without exceeding the tank capacities.

From Table 6 it follows that the growth in both the model size and the computational effort is mostly influenced by the increase in the number of batches, either in direct or reverse flow. Fortunately, our MILP continuous model permits to re-utilize batches when the pipeline is operated more than once in the same direction. In other words, variables $u_i^{(k)}$ or $v_i^{(k)}$ may be equal to one at two or more runs k but always involving the same product assigned to element i. This is so because the domain of the binary variable $y_{i,p}$ does not include the run index k. Then, it is very likely that the number of flow reversals needed to fulfill market demands does not affect the values of $\lvert ID\rvert$ and $\lvert IR\rvert$ required for achieving the optimal pipeline schedule. This is a valuable strength of the proposed MILP formulation. Anyway, the CPU time taken to find the optimal solution for a planning horizon comprising two weeks remains quite reasonable, well below 100 s even for cases involving more batches and pumping runs than those used in the optimal solution.

6.4. Example 3

Example 3 is a variant of Example 2 that incorporates a new distribution center (called N3) in the middle of the reversible pipeline

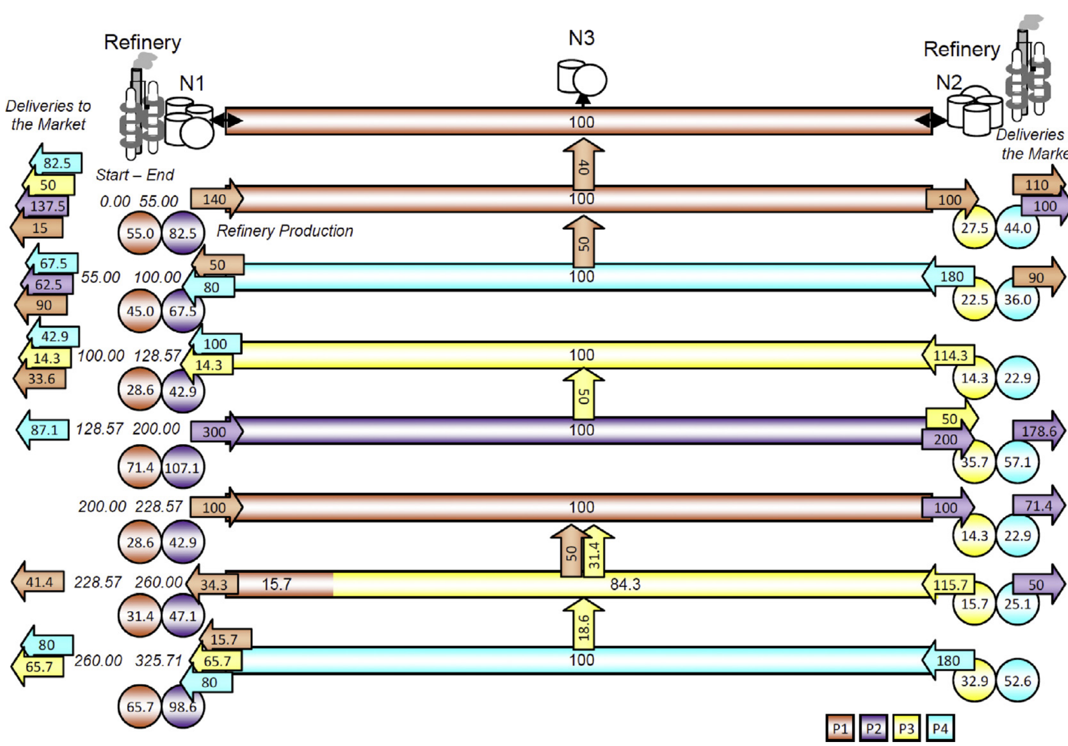

Fig. 9. Optimal schedule for the pipeline network of Example 3.

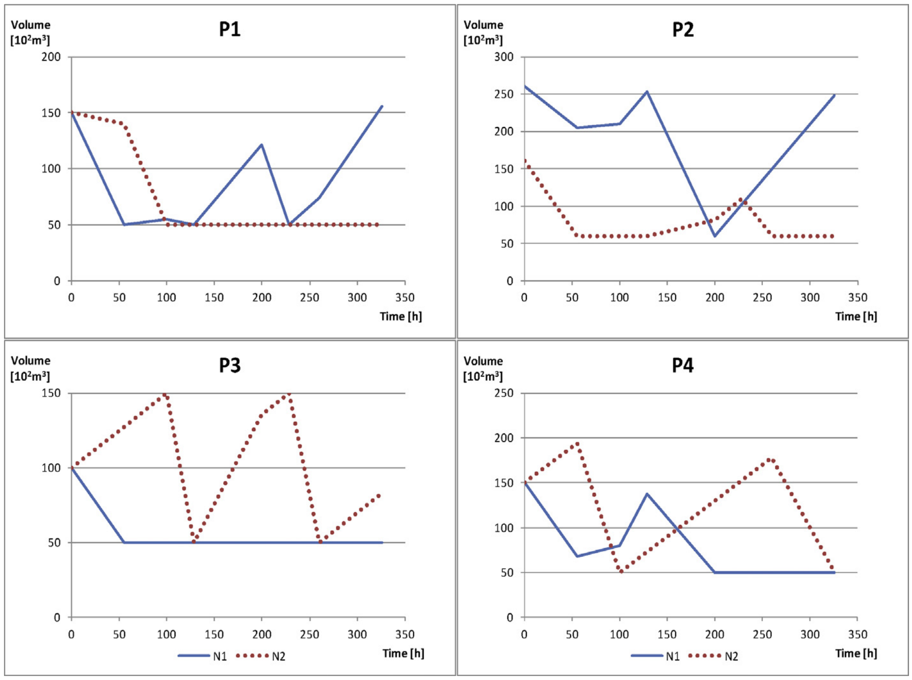

Fig. 10. Product inventory profiles at terminals N1 and N2 of Example 3.

system. The aim of Example 3 is to analyze how much the pipeline company can reduce their logistic costs by making use of an intermediate depot to fulfill some of the market requirements. In fact, it is now assumed that terminal N3 can receive, store and deliver products *P*1 (supplied from node N1) and *P*3 (supplied from node N2) to neighboring customers. However, tanks of both products at N3, whose minimum/maximum capacities are set to 50/150 and 50/100 10^2 m^3 for *P*1 and *P*3, respectively, are initially at their minimum levels. Besides, the unit inventory carrying costs for *P*1 and *P*3 at N3 are similar to the values for the other terminals, i.e. 0.19 and 0.24 USD/m^3. Some of the demands for product *P*1 covered by terminal N2 (amounting to 140 units) and 100 units of *P*3 originally dispatched from N1 in Example 2 are now to be supplied from depot N3. All the other demands and costs remain unchanged with regards to Example 2. On the other hand, terminal N3 can deliver products *P*1 and *P*3 to customers at maximum rates of 200 and 100 (m^3/h), respectively.

The optimal operational schedule for Example 3, involving 7 pumping runs, is found in 171.0 CPUs and is depicted in Fig. 9. Volumes of products supplied from refineries to nodes N1 and N2 during each pumping run are given inside the colored circles at both ends of the pipeline. As reported in Table 5, the least operation cost amounts to 248,531 USD, i.e. a 6% cost reduction with regards to Example 2. Pumping costs, inventory carrying and interface reprocessing charges are all cut down. Besides, the solution makespan is slightly lowered from 327.5 h to 325.7 h. As expected, the new configuration demands a higher computational effort since the model size increases. The sets of direct (*ID*) and reverse batches (*IR*) now contain 3 elements each and, despite the number of pumping operations remains unchanged, the number of binary variables rises from 160 to 273. The higher number of lots is related to the need of changing the product sequence, both in direct and reverse flow, at different pumping cycles. In the first cycle, the pumping sequence is $P1_D$ – $P4_R$ – $P3_R$, while in the second cycle the string is given by $P2_D$ – $P1_D$ – $P3_R$ – $P4_R$. Compared to Example 2, whose optimal sequence was $P1_D$ – $P2_D$ – $P3_R$ – $P4_R$ – $P2_D$ – $P3_R$ – $P4_R$, an 8800 USD interface cost saving is achieved.

Regarding product inventory profiles, product *P*3 at terminal N2 reaches its maximum admissible level twice, at t = 100 h and t = 228.57 h, i.e. just at the moment of injecting lots of that product into the pipeline. Immediately after both injections, the inventory of *P*3 at N2 reaches its minimum admissible level (see Fig. 10). On the other hand, all amounts of product diverted from the pipeline to the intermediate terminal N3 are rapidly transferred to clients.

An interesting feature of the proposed solution is the management of the filler lot at flow reversals (t = 55 h, t = 128.57 h, and t = 228.57 h). Instead of fully returning to the original tank (which is clearly an inefficient operation requiring an additional pumping cost and extra tank capacity), the filler lot is partially diverted to the intermediate terminal N3 while coming back to either N1 or N2. In this way, the average inventory levels at the tank-farms of both refineries are diminished.

7. Conclusions

This work presents a novel MILP mathematical formulation for the scheduling of reversible-flow multiproduct pipelines. It is a continuous representation in both time and volume domains proposed for the operational planning of a pipeline system connecting a pair of refineries or harbors to transport oil products in both directions. The bidirectional pipeline has a pair of input/output stations at both line ends to pump petroleum derivatives in direct and reverse flow. Moreover, intermediate depots can also receive products from both sources. To generate the input and output schedules in a single step, the model takes into account: (i) product demands from nearby customers to be satisfied by every terminal before the end of the planning horizon, (ii) continuous product supplies from refineries to the extreme terminals, and (iii) daily on-peak periods with much higher electricity unit cost. In addition, multiple flow reversals are allowed by the proposed formulation to meet customer demands while satisfying all problem constraints. The injection of a filler lot to sweep out the pipeline linefill before a flow reversal and the selection of the filler product are automatically made by the model. The problem goal is to find the feasible pipeline schedule that minimizes the total operation cost. A critical problem issue is to choose the number of batches to inject in direct and reverse flow over the planning horizon, and the number of pumping operations. These values determine the model size but are unknown before solving the problem. Simple expressions are given to estimate these critical model parameters, whose values increase with the number of products, terminals and flow reversals.

Three examples have been solved using the proposed formulation. The first one deals with a real-world bidirectional pipeline transporting 8 products from/to a pair of input/output terminals located at both extreme sections. A single flow reversal is allowed to compare the computational results with those reported in the literature by other authors using discrete approaches. Significant savings in operation costs and CPU time can be achieved as long as discrete methodologies do not guarantee optimality and need an iterative process to determine the optimal makespan. The second example considers another reversible pipeline carrying only 4 products, but adding further complexities to the problem. In this case, there are continuous product supplies from refineries to extreme terminals, which in turn dispatch shipments of products to meet demands of nearby customers. As a result, multiple flow reversals are automatically planned by the model to prevent from tank overloading or inventories below the minimum level. Moreover, the times at which the flow direction should be inverted are also provided by the model solution. When the inventory of a product reaches a maximum level at some source node, it is immediately pumped into the line to avoid tank overloading. If instead the inventory of a product drops to the minimum level, the input activity moves to the opposite source node to pump the scarce product toward the terminal with depleted inventories. The third example considers a modified reversible pipeline configuration that also includes an intermediate depot receiving lots of products from both refineries. The presence of intermediate depots permits to get savings in interface and pumping costs but at the same time requires to enlarge the number of batches to be injected in both direct and reverse flow, and consequently the model size and the CPU time increase. Nonetheless, the optimal solution is still found at a relatively low CPU time.

Future works will be focused on generalizing the proposed formulation to deal with the operational planning of mesh-structure pipeline networks, including reversible and non-reversible segments.

Acknowledgments

The authors would like to acknowledge financial support from FONCyT-ANPCyT under Grant PICT 1073, from CONICET under Grant PIP-2221, and from “Universidad Nacional del Litoral” under CAI+D 66-335.

Appendix A. Operating the pipeline in daily peak-hours

As originally proposed by Cafaro and Cerdá (2004), the new problem formulation for the scheduling of reversible pipelines is able to decide whether or not the pipeline should be running during daily on-peak energy hours, at the expense of paying a higher electricity unit cost. Let the parameters s_e and f_e stand for the start and end times of the peak period $e \in E$, where E is the set of peak periods over the time horizon. To account for pipeline operation in peak periods, the following binary variables are included:

(a) The variable $ws_{k,e}$ denoting that run k does not start later than the initial time of peak period e ($C_k - L_k \le s_e$) if $ws_{k,e} = 1$. Otherwise, $ws_{k,e} = 0$.
(b) The variable $wf_{k,e}$ equal to one if run k is not completed before the end time of peak period e ($C_k \ge f_e$).

The values of both variables are determined through constraints (A1) and (A2)

$$s_e(1 - ws_{k,e}) \le C_k - L_k \le s_e + (h_{\max} - s_e)(1 - ws_{k,e}), \quad \forall k \in K, e \in E \tag{A1}$$

$$f_e wf_{k,e} \le C_k \le f_e + (h_{\max} - f_e) wf_{k,e}, \quad \forall k \in K, e \in E \tag{A2}$$

Let $TK_{k,e}$ be a continuous variable denoting the length of run k performed within the peak period $e \in E$. Its value can be determined in terms of the binary variables $ws_{k,e}$ and $wf_{k,e}$ as follows:

(i) If $ws_{k,e} = wf_{k,e} = 1$, run k begins before the start of period e and ends after period e has finished. Then: $C_k - L_k \le s_e < f_e \le C_k$. Hence, $TK_{k,e} = f_e - s_e$ as imposed by constraint (A3) only if both variables $ws_{k,e}$ and $wf_{k,e}$ are equal to one. Otherwise, constraint (A3) is redundant

$$TK_{k,e} \ge (f_e - s_e)(ws_{k,e} + wf_{k,e} - 1), \quad \forall k \in K, e \in E \tag{A3}$$

(ii) If $ws_{k,e} = wf_{k,e} = 0$, run k starts after the initial time of period e (s_e) and finishes before the end of period e (f_e). Then: $s_e < C_k - L_k \le C_k < f_e$. In this case, run k is fully accomplished during peak period e, and $TK_{k,e} = L_k$, as prescribed by constraint (A4) if both binary variables are null

$$TK_{k,e} \ge L_k - l_{\max}(ws_{k,e} + wf_{k,e}), \quad \forall k \in K, e \in E \tag{A4}$$

(iii) If $ws_{k,e} = 0$ and $wf_{k,e} = 1$, run k starts after $t = s_e$ and ends after $t = f_e$. As a result, two cases should be considered:
1. Run k starts before time $t = f_e$. Then: $s_e < C_k - L_k \le f_e \le C_k$. In this case, the portion of run k accomplished over peak period e is given by: $TK_{k,e} = f_e - (C_k - L_k)$.
2. Run k starts after time $t = f_e$. Then: $s_e < f_e < C_k - L_k \le C_k$. Here, run k is totally operated outside the peak period e, and $TK_{k,e} = 0$.

Both instances can be taken into account through the single constraint (A5). Note that for case (iii.2), the RHS of A5 takes a negative value

$$TK_{k,e} \ge f_e(wf_{k,e} - ws_{k,e}) - (C_k - L_k), \quad \forall k \in K, e \in E \tag{A5}$$

(iv) If $ws_{k,e} = 1$ and $wf_{k,e} = 0$, run k starts before $t = s_e$ and ends before $t = f_e$, giving rise to two situations:

1. Run k ends after time $t = s_e$. Then: $C_k - L_k \le s_e \le C_k < f_e$. In this case, the portion of run k accomplished during peak period e is: $TK_{k,e} = C_k - s_e$.
2. Run k ends before time $t = s_e$. Then: $C_k - L_k \le C_k < s_e < f_e$, and run k is totally operated before the peak period e. Consequently, $TK_{k,e} = 0$.

Both instances are considered through constraint (A6)

$$TK_{k,e} \ge C_k - s_e - h_{\max}(1 - ws_{k,e} + wf_{k,e}), \quad \forall k \in K, e \in E \tag{A6}$$

Finally, speed-up constraints derived from the chronological arrangement of sets K (runs) and E (peak periods) are presented in constraints (A7) and (A8)

$$ws_{k,e-1} \le ws_{k,e} \le ws_{k-1,e}, \quad \forall k \in K, e \in E \tag{A7}$$

$$wf_{k-1,e} \le wf_{k,e} \le wf_{k,e-1}, \quad \forall k \in K, e \in E \tag{A8}$$

Constraint (A7) states that run k starts after time s_{e-1} if it begins after time s_e ($ws_{k,e-1} = 0$ if $ws_{k,e} = 0$). In turn, run $(k-1)$ starts before time s_e if the succeeding run k begins earlier than s_e ($ws_{k-1,e} = 1$ if $ws_{k,e} = 1$). Constraint (A8) imposes similar conditions for the variable $wf_{k,e}$ by comparing the end times of the run k and the peak period e with those of the previous run $(k-1)$ and the previous peak period $(e-1)$.

Appendix B. Controlling the temporary use of the safety stock

In order to control the amount of product that may be temporarily used from the safety stock of receiving terminals with the aim of covering the time delay in the arrival of a batch, some new constraints can be added to the formulation. First, we determine a lower bound on the earliest arrival time of the batch i during the injection k, as in Eqs. (B1) (for direct flow) and (B2) (for reverse flow)

$$AT_{i,j}^{(k)} \geq AT_{i-1,j}^{(k)} + pr_{\max}^{D} D_{i-1,j}^{(k)} - h_{\max}\left(2 - \sum_{i' \in ID} u_{i'}^{(k)} - x_{i,j}^{(k)}\right), \quad \forall i \in I, k \in K, j \in J, j \neq j_D \tag{B1}$$

$$AT_{i,j}^{(k)} \geq AT_{i+1,j}^{(k)} + pr_{\max}^{R} D_{i+1,j}^{(k)} - h_{\max}\left(2 - \sum_{i' \in IR} v_{i'}^{(k)} - x_{i,j}^{(k)}\right), \quad \forall i \in I, k \in K, j \in J, j \neq j_R \tag{B2}$$

If batch i is not delivered to terminal j during run k (i.e., $x_{i,j}^{(k)} = 0$), variable $AT_{i,j}^{(k)}$ has no physical meaning. Moreover, $AT_{i,j}^{(k)} \geq C_k - L_k$.

Next, the inventory level of every product p at terminal j is controlled at the arrival time of every batch i delivered to j during run k through the continuous variable $IA_{p,j,i}^{(k)}$, as in Eqs. (B3) and (B4) (for direct and reverse flow, respectively). The arrival time of a batch is the worst condition for a product shortage

$$IA_{p,j,i}^{(k)} \leq IA_{p,j,i-1}^{(k)} + fr_{p,j}(AT_{i,j}^{(k)} - AT_{i-1,j}^{(k)}) + \sum_{i' < i} DP_{i',j,p}^{(k)} - DA_{p,j,i}^{(k)} + id_{p,j}^{\max}\left(2 - \sum_{i' \in ID} u_{i'}^{(k)} - x_{i,j}^{(k)}\right), \quad \forall i \in I, k \in K, p \in P, j \neq j_D \tag{B3}$$

$$IA_{p,j,i}^{(k)} \leq IA_{p,j,i+1}^{(k)} + fr_{p,j}(AT_{i,j}^{(k)} - AT_{i+1,k}) + \sum_{i' > i} DP_{i',j,p}^{(k)} - DA_{p,j,i}^{(k)} + id_{p,j}^{\max}\left(2 - \sum_{i' \in IR} v_{i'}^{(k)} - x_{i,j}^{(k)}\right), \quad \forall i \in I, k \in K, p \in P, j \neq j_R \tag{B4}$$

For the first and the last batches in the linefill ($i1$, iN), the inventory level at the previous arrival time satisfies $IA_{p,j,i1-1}^{(k)} = IA_{p,j,iN+1}^{(k)} = ID_{p,j}^{(k-1)}$. Besides, $DA_{p,j,i}^{(k)}$ is the amount of product p delivered to customers of terminal j from the arrival time of the previous batch ($i-1$ in direct flow, $i+1$ in reverse flow) up to the arrival time of the batch i at the same terminal. Eqs. (B5)–(B7) monitor the amount of product delivered to the customers between subsequent batch arrivals

$$DA_{p,j,i}^{(k)} \leq dr_{p,j}(AT_{i,j}^{(k)} - AT_{i-1,j}^{(k)}) + d_{\max}\left(2 - \sum_{i' \in ID} u_{i'}^{(k)} - x_{i,j}^{(k)}\right), \quad \forall i \in I, k \in K, p \in P, j \neq j_D \tag{B5}$$

$$DA_{p,j,i}^{(k)} \leq dr_{p,j}(AT_{i,j}^{(k)} - AT_{i+1,j}^{(k)}) + d_{\max}\left(2 - \sum_{i' \in IR} v_{i'}^{(k)} - x_{i,j}^{(k)}\right), \quad \forall i \in I, k \in K, p \in P, j \neq j_R \tag{B6}$$

$$DM_{p,j}^{(k)} = \sum_{i \in I} DA_{p,j,i}^{(k)}, \quad \forall k \in K, p \in P, j \in J \tag{B7}$$

Finally, the temporary use of the safety stock (in volume units) is determined though Eq. (B8), and should be penalized in the model objective function

$$SS_{p,j,i}^{(k)} \geq id_{p,j}^{\min} - IA_{p,j,i}^{(k)}, \quad \forall i \in I, k \in K, p \in P, j \in J \tag{B8}$$

References

Boschetto, S. N., Magatão, L., Brondani, W. M., Neves-Jr, F., Arruda, L. V. R., Barbosa-Póvoa, A. P. F. D., et al. (2010). An operational scheduling model to product distribution through a pipeline network. *Industrial and Engineering Chemistry Research*, *49*, 5661–5682.

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

Cafaro, D. C., & Cerdá, J. (2004). Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation. *Computers and Chemical Engineering*, *28*, 2053–2068.

Cafaro, D. C., & Cerdá, J. (2008a). Dynamic scheduling of multiproduct pipelines with multiple delivery due dates. *Computers and Chemical Engineering*, *32*, 728–753.

Cafaro, D. C., & Cerdá, J. (2008b). Efficient tool for the scheduling of multiproduct pipelines and terminal operations. *Industrial and Engineering Chemistry Research*, *47*, 9941–9956.

Cafaro, D. C., & Cerdá, J. (2009). Optimal scheduling of refined products pipelines with multiple sources. *Industrial and Engineering Chemistry Research*, *48*, 6675–6689.

Cafaro, D. C., & Cerdá, J. (2011). A rigorous mathematical formulation for the scheduling of tree-structure pipeline networks. *Industrial and Engineering Chemistry Research*, *50*, 5064–5085.

Cafaro, D. C., & Cerdá, J. (2012). Rigorous scheduling of mesh-structure refined petroleum pipeline networks. *Computers and Chemical Engineering*, *38*, 185–203.

Castro, P. M. (2010). Optimal scheduling of pipeline systems with a resource-task network continuous-time formulation. *Industrial and Engineering Chemistry Research*, *49*, 11491–11505.

García-Sánchez, A., Arreche, L. M., & Ortega-Mier, M. (2008). Combining simulation and tabu search for oil-derivatives pipeline scheduling. *Studies in Computational Intelligence*, *128*, 301–325.

Hane, C. A., & Ratliff, H. D. (1995). Sequencing inputs to multi-commodity pipelines. *Annals of Operations Research*, *57*(1), 73–101.

Herrán, A., de la Cruz, J. M., & de Andrés, B. (2010). A mathematical model for planning transportation of multiple petroleum products in a multi-pipeline system. *Computers and Chemical Engineering*, *34*, 401–413.

Herrán, A., de la Cruz, J. M., & de Andrés, B. (2012). Global search metaheuristics for planning transportation of multiple petroleum products in a multi-pipeline system. *Computers and Chemical Engineering*, *37*, 248–261.

Lopes, T. M. T., Ciré, A. A., de Souza, C. C., & Moura, A. V. (2010). A hybrid model for a multiproduct pipeline planning and scheduling problem. *Constraints*, *15*, 151–189.

Magatão, L., Arruda, L. V. R., & Neves-Jr, F. (2004). A mixed integer programming approach for scheduling commodities in a pipeline. *Computers and Chemical Engineering*, *28*, 171–185.

Magatão, L., Arruda, L. V. R., & Neves-Jr, F. (2011). A combined CLP–MILP approach for scheduling commodities in a pipeline. *Journal of Scheduling*, *14*, 57–87.

MirHassani, S. A., & Jahromi, H. F. (2011). Scheduling multi-product tree-structure pipelines. *Computers and Chemical Engineering*, *35*, 165–176.

Moura, A. V., de Souza, C. C., Cire, A. A., & Lopes, T. M. (2008). Planning and scheduling the operation of a very large oil pipeline network. *Lecture Notes in Computer Science*, *5202*, 36–51.

Rejowski, R., & Pinto, J. M. (2003). Scheduling of a multiproduct pipeline system. *Computers and Chemical Engineering*, *27*, 1229–1246.

Rejowski, R., & Pinto, J. M. (2008). A novel continuous time representation for the scheduling of pipeline systems with pumping yield rate constraints. *Computers and Chemical Engineering*, *32*, 1042–1066.

Relvas, S., Matos, H. A., Barbosa-Póvoa, A. P. F. D., Fialho, J., & Pinheiro, A. S. (2006). Pipeline scheduling and inventory management of a multiproduct distribution oil system. *Industrial and Engineering Chemistry Research*, *45*, 7841–7855.

Sasikumar, M., Prakash, P. R., Patil, S. M., & Ramani, S. (1997). PIPES: A heuristic search model for pipeline schedule generation. *Knowledge-Based Systems*, *10*, 169–175.