Computers & Operations Research 73 (2016) 27–42

Contents lists available at ScienceDirect

Computers & Operations Research

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

Short-term scheduling of multiple source pipelines with simultaneous injections and deliveries

Hossein Mostafaei [a,b], Pedro M. Castro [b], Alireza Ghaffari-Hadigheh [a,*]

[a] *Department of Applied Mathematics, Azarbaijan Shahid Madani University, Tabriz, Iran*
[b] *Centro de Matemática Aplicaçõoes Fundamentais e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal*

ARTICLE INFO

Article history:
Received 10 January 2015
Received in revised form
14 October 2015
Accepted 17 March 2016
Available online 24 March 2016

Keywords:
Mathematical modeling
Mixed-integer linear programming
Generalized disjunctive programming
Multiproduct pipelines

ABSTRACT

This paper addresses the optimal scheduling of straight pipelines featuring multiple intermediate nodes acting as dual-purpose stations, with a continuous-time Mixed-Integer Linear Programming formulation partly derived from Generalized Disjunctive Programming. The new model allows for an intermediate station to act as an output and input terminal at the same time so as to reduce the number of segment switches between active and idle, and consequently decrease operating costs. Contrary to previous approaches, decisions related to batch sizing, batch sequencing and timing are determined in a single step. Several examples of growing complexity are solved to illustrate the effectiveness and computational advantage of the proposed model in both solution quality and CPU time.

1. Introduction

Pipelines are the most reliable and cost-effective way of transporting large amounts of refined petroleum products from major supply sources to distribution centers [1]. Pipelines usually operate without interruption and can carry a variety of petroleum products (e.g. gasoline, kerosene, jet fuel, diesel, heating oil, liquefied petroleum gases), accounting for around two-thirds of all refined petroleum products shipped in the US [2,3]. Common carrier pipeline networks have grown extensively and have become more complex structurally. While this makes it more challenging to meet the goal of satisfying demand on time at the different locations (recall that the pipeline scheduling problem contains the Hamiltonian path that is known to be NP-complete when interface restrictions are taken into account [4]), it may also increase the benefits from optimal operation and planning. Note that due to the large investment and operation costs, even small improvements can involve considerable amounts of money [5].

The planning and scheduling of multiproduct pipelines have been addressed from different perspectives such as knowledge-based heuristic techniques [6], simulation tools [7,8], decomposition methods [9,10] and mathematical programming models, which are typically of the mixed-integer linear type (MILP) but may also feature non-linear constraints (MINLP) [11]. The latter are usually divided into discrete and continuous time models. A discrete-time approach divides the scheduling horizon into time intervals of fixed duration and the pipeline into packages of uniform sizes, each containing a single product [12–16]. In contrast, in a continuous-time approach, the duration of the time slots will be determined by the optimization so as to adjust to the optimal batch sizes and pumping rates. Continuous-time approaches are the latest trend and so will be overviewed next. Meanwhile, and concerning the structural arrangement of the pipeline, one may have the simplest case with a unique refinery and a depot [17–20], a single refinery with several off-take terminals [2,21,14,22–25], a straight system with multiple input and output stations [26–29], a tree-like pipeline [30–32], to mesh-structured pipeline networks [33–36].

When products are injected into a pipeline, mixing occurs at their interface that results into product contamination. Such interface volume typically returns to the refinery for reprocessing and the cost of this operation is very high [37]. The extent of mixing is a function of the characteristics of the pipeline (length and diameter), the flow rate and the physical properties of the products involved [2,21]. One way to decrease the interface cost is to minimize the number of pipeline stoppages. It requires determining the sequence and timing of stripping operations in receiving terminals for every pumping operation, which is referred to as detailed scheduling.

Cafaro et al. [38,39,11] presented two-level hierarchical solution approaches featuring continuous-time MILPs for the detailed scheduling of a single source pipeline with multiple output terminals. The

* Corresponding author.
E-mail addresses: h.mostafaei65@azaruniv.edu (H. Mostafaei), pmcastro@fc.ul.pt (P.M. Castro), hadigheha@azaruniv.edu (A. Ghaffari-Hadigheh).

http://dx.doi.org/10.1016/j.cor.2016.03.006
0305-0548/© 2016 Elsevier Ltd. All rights reserved.

upper level (aggregate schedule) seeks to find the optimal sequence of product injections, the size and destination of each product batch. The lower level involves the pump optimization phase (detailed scheduling). They started [38] considering delivery to a single terminal at a time, later [39] allowing for simultaneous deliveries and more recently [11], accounting for flow rate dependent energy consumption with nonlinear constraints. A straight pipeline with multiple dual-purpose terminals was the subject of [40].

Mostafaei and co-workers [41,42] have shown that a substantial reduction in pump operating and restart/stoppage costs can be achieved with an MILP continuous-time formulation that is able to find the product sequencing, lot-sizing, pumping and delivery schedule in a single step. Ghaffari-Hadigheh and Mostafaei [41] have dealt with a straight pipeline, one refinery and simultaneous deliveries to multiple output terminals. The tree-like topology with a single refinery is tackled in [42] and, contrary to [43], allows for simultaneous deliveries and flow rate limitations that are a function of the pipeline segment diameter.

In this paper, we continue along the lines of a monolithic approach for the detailed scheduling of multiproduct pipelines. We now focus on straight pipelines with multiple dual-purpose terminals where simultaneous injections and deliveries are allowed. It can be viewed as an upgrade of our model in [29] for the same topology, which allowed a single terminal to inject into the pipeline at any given time. In order to ensure an efficient MILP formulation, we rely on Generalized Disjunctive Programming [44–46] for deriving some of the constraints, applying convex-hull or big-M reformulations of disjunctions and converting logic propositions into integer inequalities. This model derivation approach has been shown to be useful for generating complex constraints when relying on a continuous-time representation for scheduling [47–51].

The rest of the paper is organized as follows: Section 2 presents a brief description of the problem under study. Section 3 puts forward a motivating example to better highlight its features. Section 4 derives the mixed integer linear programming model for the detailed scheduling of the multi-source straight pipeline networks. Section 5 provides numerical examples to illustrate the advantages resulting from allowing simultaneous injections and from using a full-space rather than a decomposition approach. Finally, the conclusions are given in Section 6.

2. Problem statement

We consider a pipeline system that must transport oil products from some input terminals to a few output terminals. The straight pipeline connects the first input terminal to the last output terminal and there are a few single or dual purpose stations in between that can send/receive products to/from the pipeline. The section of the pipeline connecting consecutive terminals is called a segment with the volume of a product inside the segment being regarded as a batch or lot, see Fig. 1.

Consider the junction between a dual-purpose station and its pipeline segments. Assuming material movement, five different operating modes are allowed as shown in Fig. 2. In mode A, the terminal is not involved and the material moves from one segment to the next. In mode B, all the material arriving at the junction goes to the terminal, which acts as an output node. In mode D, part of the material goes to the terminal and part moves on. In mode C, the terminal acts as an input node and no material is arriving from segment $d-1$. The new operating mode being handled in this paper is mode E, with the terminal receiving material from segment $d-1$ while simultaneously sending a different material to segment d. Note that in real-world pipeline systems, pipeline segment d can receive both from $d-1$ and the terminal. Considering this additional operating mode will be the subject of future work.

The remaining assumptions reflecting practical considerations are as follows:

A1. Unidirectional flow in the pipeline (from left to right in the diagrams).
A2. The pipeline is completely full with incompressible liquids. In that way, when a given quantity of product is inserted into a segment of the pipeline, the same quantity (of the same or another product) leaves the segment at the other extremity.
A3. At any time, at most a single batch (product) injection can be performed at an input terminal.
A4. At any time, an output terminal can receive material from a single batch (product).
A5. Flow rates at input terminals and pipeline segments can vary within given ranges.
A6. To avoid high product contamination, some product sequences are forbidden.
A7. Product demands are to be satisfied at the end of the time horizon and are given data.

3. Motivating example

We start by showing the advantage of allowing simultaneous injections and deliveries at dual purpose stations (mode E in Fig. 2). The illustrative example involves the scheduling of a multiproduct pipeline of 100 units in volume (v.u.), transporting four products from three input terminals (N1, N2 and N3) to three output terminals (N2, N3 and N4), with nodes N2 and N3 acting as dual purpose stations (the pipeline topography and its initial situation can be found in the first line of Fig. 3). The admissible product injection rate at input nodes, given in (v.u./h), are: [0.80, 1.40] for node N1, [0.60, 1.20] for node N2 and [0.40, 0.80] for node N3 whereas the acceptable flow rate ranges (v.u./h) for pipeline segments are as follows: [0.80, 1.40] for the first segment, [0.5, 1.20] for segment 2, and [0.3, 0.80] for the last one where the pipeline presents a lower diameter.

The pipeline operator plans to supply 10 v.u. of product P2 ($P2_{10}$) and $P1_{30}$ to terminal N2, product $P2_{40}$ to terminal N3, and product $P4_{40}$ to terminal N4, within the next 4 days. There are three additional constraints: (i) only 100 (v.u.) of product P1 at

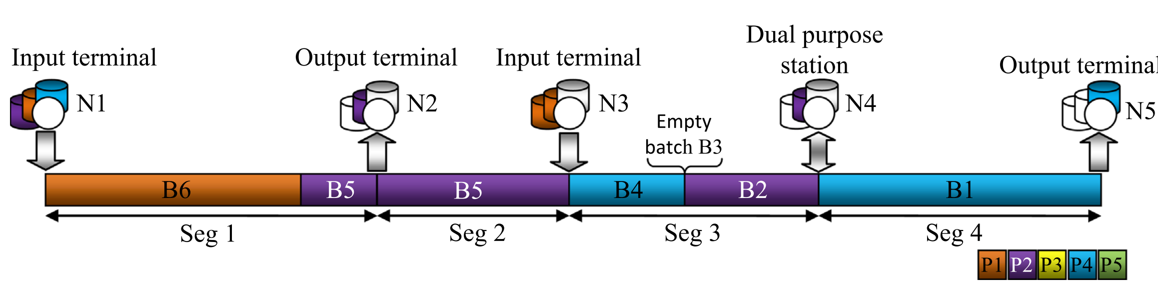

Fig. 1. Pipeline transportation problem schematic overview.

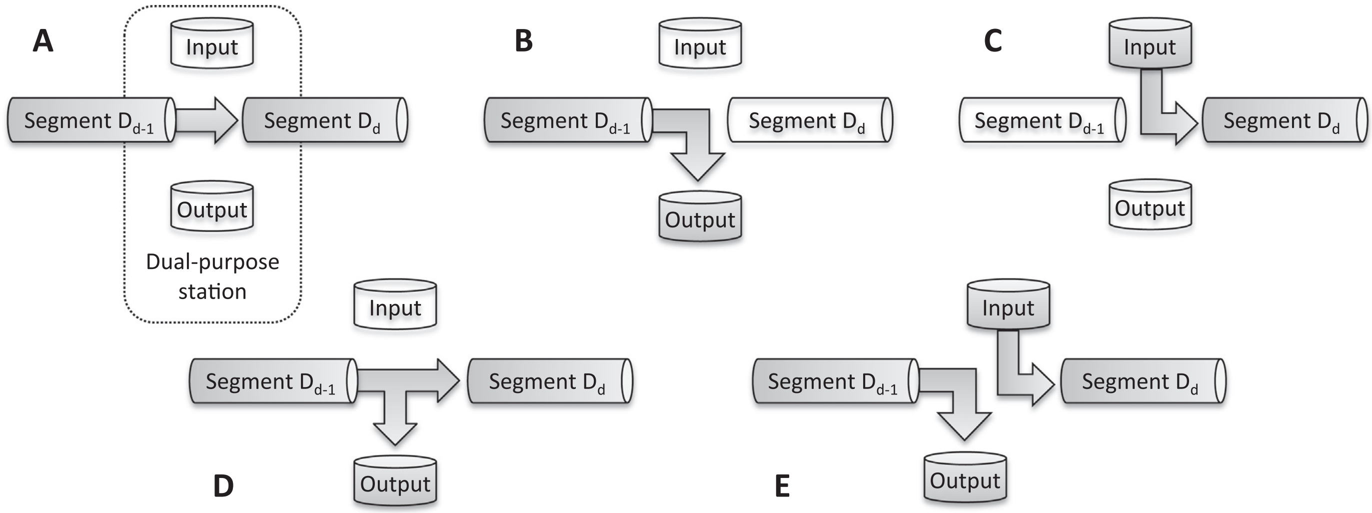

Fig. 2. Possible active operating modes of the pipeline at a dual-purpose terminal junction.

input node N1 and 20 (v.u.) of product P4 at input node N3 are available for the next 4 days, (ii) the product delivery to the receiving terminals can have a maximum size of 30 (v.u.) and a minimum size of 2.5 (v.u.), and (iii) any shipment pumped from input terminals can have a maximum size of 80 (v.u.) and a minimum size of 10 (v.u.). The aim is to meet product demands at the minimum number of flow restarts/stoppages and on/off pump switches.

First, we assume that simultaneous deliveries to receiving terminals are forbidden and that at any time, a single output terminal (between two active injection points) can receive products from the pipeline (here it refers to as non-simultaneous delivery mode; **NSD**). The top of Fig. 3 depicts the pipeline schedule for **NSD**. It includes five pumping operations and has a makespan of 86.89 h. Note that the third pumping run includes simultaneous injections at two input terminals N1 and N3. During the injection of batch B3 from N1, the receiving terminal N3 should extract 30 v.u. of product P2 from batch B2. On the other hand, batch B1 containing 20 units of product P4 is injected at node N3 to transfer a similar amount of the same product to the last terminal N4. The last two pumping runs insert an additional amount of product P1 through batch B3 to fulfill product demands at nodes N2 and N4. As can be seen from Fig. 3, segments 2 and 3 are inactive through the last operation that would need

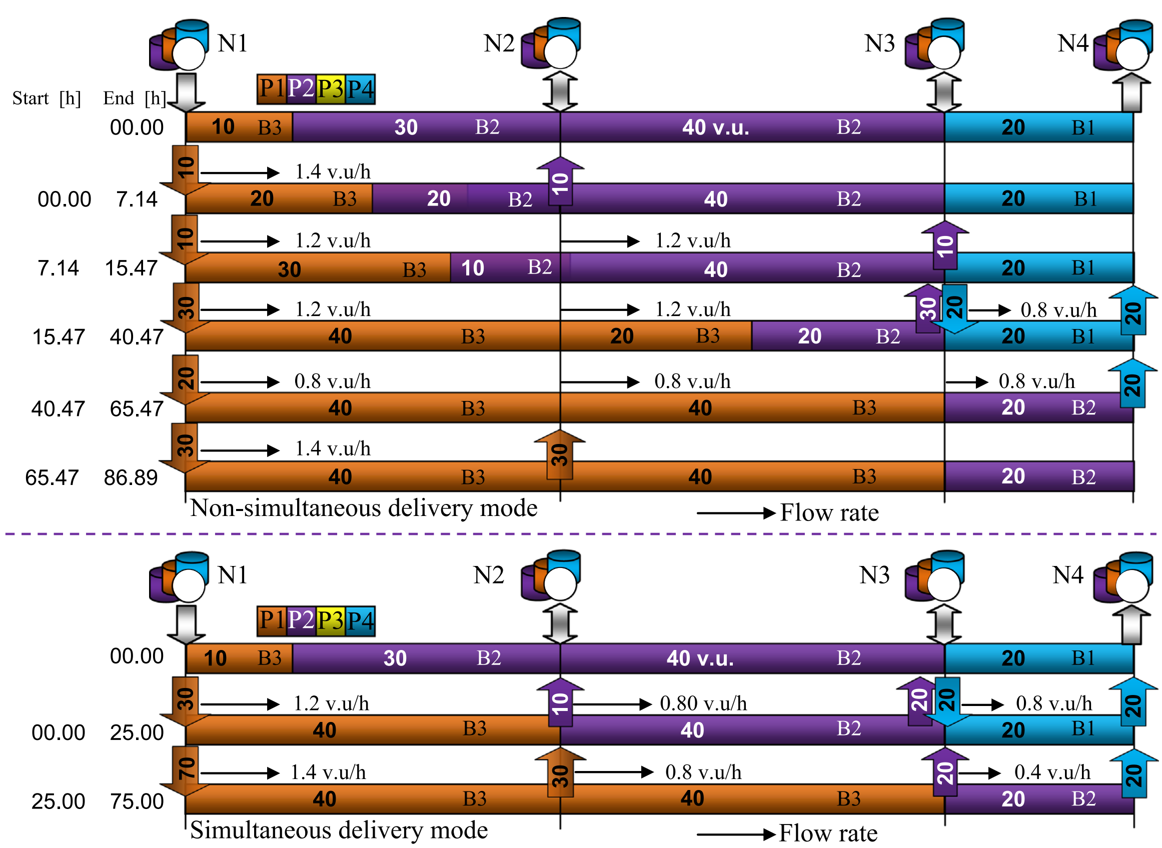

Fig. 3. Pipeline schedule for the motivating example by showing fewer flow restart/stoppages for the simultaneous delivery mode.

additional power consumption for streaming flow in the next runs.

The same example is tackled again but now allowing the output terminals to receive product from the pipeline at the same time (here referred to as simultaneous delivery mode; **SID**). The bottom of Fig. 3 depicts the best pipeline schedule for **SID**. It includes two pumping operations and has a makespan of 75.00 h that is 11.89 h below the value for NSD. When the first pumping run is executed, two input events take place: 30 units of product P1 are added to batch B3 at a flow rate of 1.2 (v.u./h) and two output terminals N2 and N3 simultaneously receive material from the pipeline. During this operation, dual-purpose node N3 inserts 20 units of product P4 into segment 3 at a flow rate of 0.8 (v.u./h), and the same amount of products is discharged at the last terminal D4. Run 2 simultaneously sends products from three batches (B3, B2 and B1) to depots N2, N3 and N4. The entire pipeline is kept in motion and therefore, there is no flow stoppage in pipeline segments.

4. New optimization model

We now present the proposed MILP formulation for the short-term scheduling of multiple source, straight pipeline systems, which employs a continuous representation in both time and volume scales. Compared to our previous work in Mostafaei et al. [29] we now: (i) allow multiple input terminals to feed the pipeline at the same time; (ii) detect when the flow in pipeline segments is resumed or stopped, since the cost of a pumping operation is mainly associated to how much energy will be required to resume the flow in idle pipeline segments. The list of sets, parameters and decision variables is shown in Tables 1, 2 and 3.

4.1. Defining batches

As stated before, the volume of a product in a segment is regarded as a batch. The existent batches already in segment d at the start of the scheduling horizon are considered as old batches ($I_d^{old} = \{i_n, i_{n+1}, \dots, i_{|I_d^{old}|+n-1} | 1 \le n\}$), with batch i_j $(1 < j)$ streaming immediately after batch i_{j-1} inside segment d. The batch numbering is set from the last segment ($d = |D|$) so that the first batch in the last segment, touching the end of that segment, is numbered i_1 (batch B1).

Remark 1. To account for the possibility of inserting a new batch between two real old batches in mid-input nodes, empty batches need to be defined. For example, the pipeline shown in Fig. 1 includes an empty batch B3 between batches B2 and B4.

The first element of set I_d^{old} $(d \ne |D|)$ is given by either $i_{|\bigcup_{d'=d+1}^{|D|} I_{d'}^{old}|}$ or $i_{|\bigcup_{d'=d+1}^{|D|} I_{d'}^{old}|+1}$. If two old batches, transporting the same products, touch each other at the junction of segments d and $d+1$, then they are given by $i_{|\bigcup_{d'=d+1}^{|D|} I_{d'}^{old}|}$. However, if the old batches have different products, the batch number of the first lot in segment d will be $i_{|\bigcup_{d'=d+1}^{|D|} I_{d'}^{old}|+1}$. For instance in Fig. 1, there is only one product in the last segment (Seg 4), and so we will have $I_4^{old} = \{B1\}$. For the third segment (Seg 3), since the old lots touching each other at the junction of segments 3 and 4 do not carry the same product, the first batch in segment 3 is numbered by $i_{|\bigcup_{d'=3+1}^{4} I_{d'}^{old}|+1=2}$, and the succeeding ones will be given by $i_3, \dots, i_{|I_3^{old}|+2-1}$, i.e., $I_3^{old} = \{B2, B3, B4\}$ (note that there are three old batches in segment 3 ($|I_3^{old}| = 3$), with batch B3 being an empty lot). Similarly for segment 2, we will have $I_2^{old} = \{B5\}$. For the first segment, since the old batches touching the junction of segments 1 and 2 transport the same product (P2), the first batch in the first segment is given by $i_{|\bigcup_{d'=1+1}^{4} I_{d'}^{old}|=|\{B1,B2,\dots,B5\}|=5}$, and the succeeding one is given by i_6.

Now, suppose that n new batches are injected into the pipeline shown in Fig. 1 during a known planning horizon (I^{new}; set of new batches that would be injected into the pipeline). Then, the set I_d^{new} given by $I_d^{new} = \{1, 2, \dots, |\bigcup_{d'} I_{d'}^{old}| + |I^{new}|\} \setminus \bigcup_{d' \ge d}^{|D|} I_{d'}^{old}$ is the set of new batches to be injected in segment d through the known planning horizon h_{max}. For example, if two new batches B7 and B8 ($|I^{new}| = 2$) are injected into the pipeline, then, set I_4^{new} will be defined as follows:

$$
\begin{aligned}
I_4^{new} = \{1, 2, \dots, |\bigcup_{d'} I_{d'}^{old}| + |I^{new}|\} \setminus \\
\bigcup_{d' \ge 4}^{|D|} I_{d'}^{old} = \{1, 2, \dots, 6 + 2\} - \{1\} = \{2, 3, \dots, 8\} \\
= \{B2, B3, \dots, B8\}.
\end{aligned}
$$

Remark 2. Set I_d given by $I_d = I_d^{new} \cup I_d^{old}$ is the set of product batches to have travelled along segment d during the planning horizon.

Remark 3. Segment d connects input node d at the origin of that segment with output terminal d at the end of that segment and so all mid nodes in the pipeline network are considered as dual purpose terminals.

4.2. Sequencing composite pumping runs

To arrange the pumping operations through the planning horizon, we first introduce the concept of composite pumping runs $k \in K$. A composite run represents a group of simultaneous pumping operations taking place at different input terminals during a time interval of the scheduling horizon. Simultaneous operations in a composite run should start and finish at the same time. Furthermore, batch injections in composite run k should begin after finishing operations related to run $k-1$. Let continuous variables S_k and L_k represent the starting time and the length of composite run $k \in K$, respectively. Because the set of composite runs is chronologically ordered,

$$S_k - S_{k-1} \ge L_{k-1}, \quad \forall\, k \in K\ (k \ge 2). \tag{1}$$

For the first composite run, let $S_1 = ST$ $(ST \ge 0)$, be a known value. Then, the duration of all composite runs should not exceed h_{max}, the overall length of the planning horizon.

$$\sum_{k \in K} L_k \le h_{max}. \tag{2}$$

4.3. Allocating products to batches

Let us use the binary variable $y_{i,p}$ to represent the allocation of product p to batch i. Every batch flowing inside the pipeline can at most transport a single product. Hence,

$$\sum_{p \in P} y_{i,p} \le 1, \quad \forall\, i \in I. \tag{3}$$

In order to reduce solution degeneracy, empty batches $i \in I^{new}$ featuring $y_{i,p} = 0$ for all $p \in P$ should be placed at the end of the sequence:

$$\neg \bigvee_{p \in P} y_{i-1,p} \Rightarrow \neg \bigvee_{p \in P} y_{i,p}, \quad \forall\, i \in I^{new}.$$

This logic proposition can be reformulated into Eq. (4).

$$\sum_{p \in P} y_{i,p} \le \sum_{p \in P} y_{i-1,p}, \quad \forall\, i \in I^{new}. \tag{4}$$

4.4. Size of batch injection and the length of composite run $k \in K$

Let 0–1 variable $w_{i,k,d}$ denote that a pumping operation belonging to composite run k inserts a batch $i \in I_d$ $(i \geq |I_d^{old}|)$ from the input node d to segment d whenever $w_{i,k,d} = 1$. Empty batches $i \in I^{new}$ featuring $y_{i,p} = 0$ for all $p \in P$ are never injected into the pipeline. In addition, a batch $i \in I^{new}$ may be pumped from input node d to the segment d in consecutive composite runs. Both conditions are imposed by the following equation:

$$\sum_{k \in k} \sum_{d \in D} w_{i,k,d} \leq |D| \, |K| \sum_{p \in P} y_{i,p}, \quad \forall\, i \in I^{new}. \tag{5}$$

In order to improve efficiency, dummy composite runs, featuring $\sum_{d \in D} \sum_{i \in I_d} w_{i,k,d} = 0$, should always be confined to the end:

$$\sum_{d \in D} \sum_{i \in I_d} w_{i,k,d} \leq |D| \sum_{d \in D} \sum_{i \in I_d} w_{i,k-1,d} \quad \forall\, k \in K\,(k \geq 1). \tag{6}$$

A particular composite run k from input node d is either associated to a batch $i \in I_d$, through binary variable $w_{i,k,d}$, or is a dummy run. In the former, there will be a certain volume $Vb_{i,k,d}$ going into segment d, which will affect the length of the pumping $LS_{k,d}$ run. Notice that both variables must lie between given lower and upper bounds. In contrast, they will be equal to zero in the latter option $(w_{k,d}^{no\ batch})$. This can be written as the disjunction [44,45,49] given below:

$$\bigvee_{i \in I_d} \begin{bmatrix} w_{i,k,d} \\ Vb_d^{min} \leq Vb_{i,k,d} \leq Vb_d^{max} \\ \frac{Vb_{i,k,d}}{vd_d^{max}} \leq LS_{k,d} \leq \frac{Vb_{i,k,d}}{vd_d^{min}} \end{bmatrix} \bigvee \begin{bmatrix} w_{k,d}^{no\ batch} \\ Vb_{i,k,d} = 0, \quad \forall\, i \in I_d \\ LS_{k,d} = 0 \end{bmatrix},$$
$$\forall\, k \in K\,(k \geq 1), \ d \in D.$$

The convex hull reformulation of the disjunction coupled with algebraic manipulations [49] to remove the resulting disaggregated variables and variable $w_{k,d}^{no\ batch}$ gives rise to the following constraints:

$$\sum_{i \in I_d} w_{i,k,d} \leq 1, \quad \forall\, k \in K\,(k \geq 1), \quad d \in D, \tag{7}$$

$$Vb_d^{min}\, w_{i,k,d} \leq Vb_{i,k,d} \leq Vb_d^{max}\, w_{i,k,d}, \quad \forall\, i \in I_d\ (i \geq \mathrm{end}(I_d^{old})),$$
$$k \in K\,(k \geq 1), \quad d \in D, \tag{8}$$

$$\sum_{i \in I_d} \frac{Vb_{i,k,d}}{vd_d^{max}} \leq LS_{k,d} \leq \sum_{i \in I_d} \frac{Vb_{i,k,d}}{vd_d^{min}}, \quad \forall\, k \in K\,(k \geq 1), \quad d \in D. \tag{9}$$

According to assumption (A3), at most a single batch injection can be performed at an input terminal at any time. Such an assumption is satisfied by constraint (7).

On the other hand, pumping operations related to a composite pumping run always start and finish at the same time. Constraints (10)–(11) enforce the equality:

$$LS_{k,d} \leq L_k, \quad \forall\, k \in K\,(k \geq 1), \ d \in D, \tag{10}$$

$$L_k \leq LS_{k,d} + h_{max}\left(1 - \sum_{i \in I_d} w_{i,k,d}\right), \quad \forall\, k \in K\,(k \geq 1), \ d \in D. \tag{11}$$

4.5. Product inventory in input terminal d

Let the continuous variable $VP_{i,p,k,d}$ represent the size of batch i conveying product p pumped from input node d during a composite run k. The value of variable $VP_{i,p,k,d}$ is zero whenever: (a) batch i does not transport product p, and/or (ii) there is no injection of product p from input node d to batch i during composite run k. Otherwise, $VP_{i,p,k,d} = Vb_{i,k,d}$. Both conditions are modelled through the below constraints:

$$\sum_{d \in D} \sum_{k \in K} VP_{i,p,k,d} \leq \left(|K| \sum_{d \in D} Vb_d^{max}\right) y_{i,p}, \quad \forall\, i \in I, \ \ p \in P, \tag{12}$$

$$\sum_{p \in P} VP_{i,p,k,d} = Vb_{i,k,d}, \quad \forall\, i \in I_d \left(i \geq end(I_d^{old})\right), \ \ k \in K\,(k \geq 1),$$
$$d \in D. \tag{13}$$

Because of inventory limitations, the total amount of product p injected from input node d is bounded. A good model should be capable of managing product inventories at input nodes. To this end, let $Inv_{p,d}$ be the maximum amount of product p that can be shipped from input node d over the time horizon h_{max}. Thus,

$$\sum_{k \in K} \sum_{i \in I_d} VP_{i,p,k,d} \leq Inv_{p,d}, \quad \forall\, p \in P, \, d \in D. \tag{14}$$

4.6. Size of products discharged to output terminals during a composite run

Let binary variable $x_{i,k,d}$ denote that output terminal d receives product from batch $i \in I_d$ during composite run k whenever $x_{i,k,d} = 1$. Each active output terminal d can only extract the product from a single batch. Hence,

$$\sum_{i \in I_d} x_{i,k,d} \leq 1, \quad \forall\, k \in K\,(k \geq 1), \ d \in D. \tag{15}$$

There will be no product deliveries to output terminals for dummy composite runs. Thus,

$$\sum_{d \in D} \sum_{i \in I_d} x_{i,k,d} \leq |D| \sum_{d \in D} \sum_{i \in I_d} w_{i,k,d}, \quad \forall\, k \in K\,(k \geq 1). \tag{16}$$

Let $VD_{i,k,d}$ be the volume of batch $i \in I_d$ discharged from segment d to output terminal d during composite run $k \in K$. It will be positive if and only if the composite run directs the batch to the output terminal $(x_{i,k,d} = 1)$. We have thus the following constraint:

$$VD_d^{min}\, x_{i,k,d} \leq VD_{i,k,d} \leq VD_d^{max}\, x_{i,k,d}, \quad \forall\, i \in I_d, \ \ k \in K\,(k \geq 1),$$
$$d \in D, \tag{17}$$

where VD_d^{min} and VD_d^{max} are, respectively, the lower and the upper bounds on the amount of material that can be transferred to output terminal d during composite run k.

Now, let $VPD_{i,p,k,d}$ be the volume of product p supplied by batch $i \in I_d$ to receiving terminal d during composite run k. The value of variable $VPD_{i,p,k,d}$ is equal to $VD_{i,k,d}$ whenever (a) batch i transports product p and (b) there is a stripping operation from batch i, transporting product p, at output terminal d during composite run k. Otherwise $VPD_{i,p,k,d} = 0$. Thus, the following constraints are valid:

$$\sum_{p \in P} VPD_{i,p,k,d} = VD_{i,k,d}, \quad \forall\, i \in I_d, \, k \in K\,(k \geq 1), \, d \in D, \tag{18}$$

$$\sum_{d \in D} \sum_{k \in k} VPD_{i,p,k,d} \leq \left(|K| \sum_{d \in D} VD_d^{max}\right) y_{i,p}, \quad \forall\, i \in I, \, p \in P. \tag{19}$$

4.7. Feed rate limitation to receiving terminals

Due to operational restrictions, some products cannot be diverted into receiving depots at full pressure. Assuming that the parameter $vp_{p,d}$ is the maximum delivery rate of product p to the receiving terminal d, we will have the following constraint:

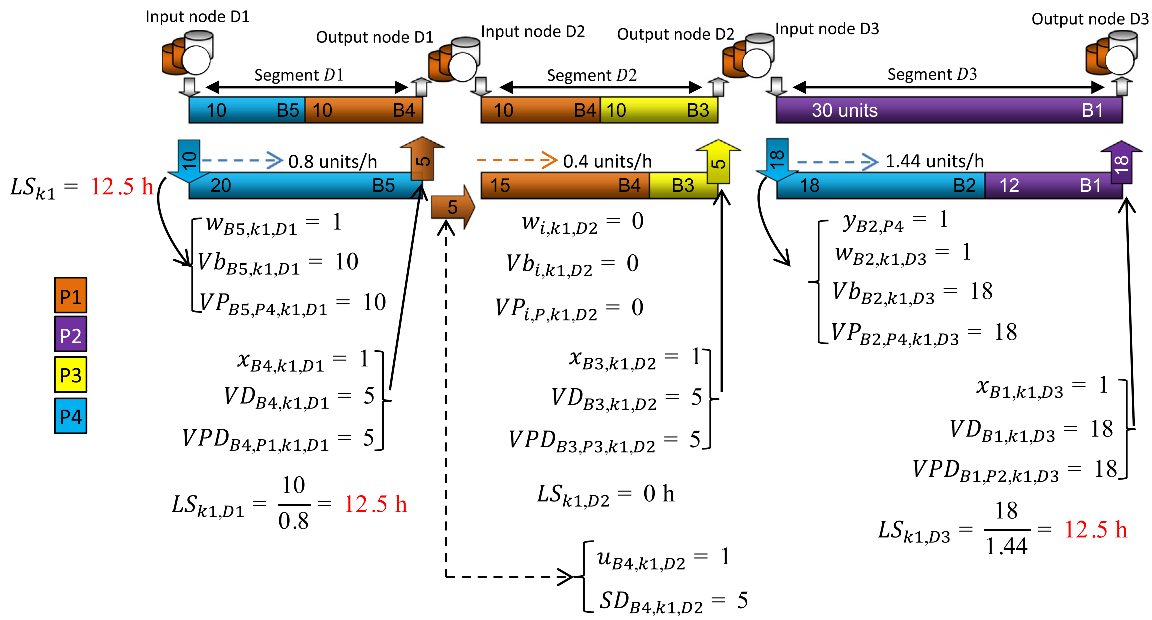

Fig. 4. A simple example illustrating constraints in parts 4.2–4.6 and 4.9.

$$\sum_{i \in I_d} VPD_{i,p,k,d} \le vp_{p,d} L_k, \quad \forall\, p \in P,\, k \in K\,(k \ge 1),\, d \in D. \tag{20}$$

Fig. 4 illustrates values of the variables appearing in Sections 4.2–4.6 and 4.9.

4.8. Batch tracking constraint

4.8.1. Size of batch i in segment d at the end of composite run k

Let continuous variable $W_{i,k,d}$ stand for the size of batch $i \in I_d$ at the end of composite run k, i.e., at time S_{k+1}. During the execution of run k, the size of batch $i \in I_d$ can be increased by receiving some material from either input node d ($Vb_{i,k,d}$) or the previous segment $d-1$ ($SD_{i,k,d}$), and decreased by transferring some portion of it to the output node d ($VD_{i,k,d}$) and segment $d+1$ ($SD_{i,k,d+1}$). The following volumetric balance results:

$$W_{i,k,d} = W_{i,k-1,d} + (Vb_{i,k,d} + SD_{i,k,d}) - (VD_{i,k,d} - SD_{i,k,d+1}),$$
$$\forall\, i \in I_d,\, k \in K\,(k \ge 1),\, d \in D, \tag{21}$$

For old batch $i \in I_d^{\text{old}}$, variable $W_{i,0,d}$ will be equal to $IW_{i,d}$, a known datum standing for the size of old batch i in segment d at the time ST.

If batch $i \in I^{\text{new}}$ still has not been injected into the pipeline through one of its input nodes, then its size $W_{i,k,d}$ in segment d is null at the end of composite run k. Hence,

$$\sum_{k' \in K}^{k} \sum_{d \in D} W_{i,k',d} \le |D|\,|K| \sum_{k' \in K}^{k} \sum_{d \in D} SEG_d\, w_{i,k',d},$$
$$\forall\, i \in I^{\text{new}},\, k \in K\ (k \ge 1), \tag{22}$$

At any time, the pipeline segments are completely full. Therefore, the total batch volume at the end of each composite run must be equal to the segment volume SEG_d:

$$\sum_{i \in I_d} W_{i,k,d} = SEG_d, \quad \forall\, k \in K,\, d \in D. \tag{23}$$

4.8.2. Location of batch i in segment d at the end of composite run k

Let continuous variable $F_{i,k,d}$ be the location of upper coordinate of batch i in segment d at the end of composite run k. It will be equal to the total volume of batches $i' \in I_d$ succeeding batch i at time S_{k+1}, plus the size of batch i at the end of composite run k (see Fig. 5):

$$F_{i,k,d} = \sum_{\substack{i' \in I_d \\ i' \ge i}} W_{i',k,d}, \quad \forall\, i \in I_d,\, k \in K,\, d \in D. \tag{24}$$

Note that by subtracting $W_{i,k,d}$ from both sides of (24), one can simply derive $F_{i,k,d} - W_{i,k,d} = \sum_{i' \ge i+1}^{i' \in I_d} W_{i',k,d} = F_{i+1,k,d}$. Thus, for every batch $i \in I_d$ we have $F_{i,k,d} - W_{i,k,d} = F_{i+1,k,d} \ge 0$. This fact is apparent from Fig. 5.

4.9. Supplying material from segment d − 1 to segment d

Let 0–1 variable $u_{i,k,d}$ denote that batch $i \in I_{d-1}$ is inserted to segment d ($d > 1$) during composite run k, whenever $u_{i,k,d} = 1$. Through run $k \in K$, a batch $i \in I_d$ can receive material from segment $d-1$ only if: (a) the frontal coordinate of batch i in segment $d-1$ has reached segment d and (b) before starting the composite run k, the lower coordinate of batch i in segment $d-1$ has not exceeded the origin of segment d. As a result, $u_{i,k,d} = 1$ implies

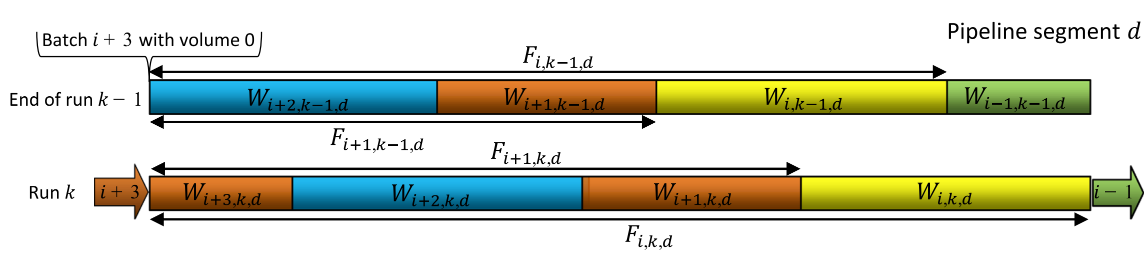

Fig. 5. Upper coordinate of batch *i*.

$F_{i+1,k-1,d-1} \le SEG_{d-1} = F_{i,k,d-1}$. Thus,

$$F_{i,k,d-1} \ge SEG_{d-1} u_{i,k,d}, \quad \forall\ i \in I_{d-1},\ k \in K(k \ge 1),\ d \in D(d \ge 2), \tag{25}$$

$$F_{i,k-1,d-1} - W_{i,k-1,d-1} \le SEG_{d-1}(2 - u_{i,k,d}), \quad \forall\ i \in I_{d-1},\ k \in K(k \ge 1),\ d \in D(d \ge 2), \tag{26}$$

Note that from constraints (23) and (24), it can be derived that $F_{i,k,d} = \sum_{i' \ge i}^{i' \in I_d} W_{i',k,d} \le \sum_{i' \in I_d} W_{i',k,d} = SEG_d$. We thus have $F_{i,k,d} \le SEG_d$ for all $i \in I_d$, $k \in K$, $d \in D$.

Let non-negative continuous variable $SD_{i,k,d}$ represent the size of batch i inserted from segment $d-1$ to segment d during composite run k. No material can be inserted to batch $i \in I_d$ from batch $i \in I_{d-1}$ existing in the segment $d-1$ if binary variable $u_{i,k,d}$ takes a value of zero. Thus

$$VD_d^{\min} u_{i,k,d} \le SD_{i,k,d} \le Vb_d^{\max} u_{i,k,d}, \quad \forall\ i \in I_{d-1},\ k \in K(k \ge 1),\ d \in D(d \ge 2). \tag{27}$$

Now assume that segment d receives material from the previous segment through the composite run k, and the coordinates of batch i in segment $d-1$ satisfy the following condition: $F_{i+1,k-1,d-1} \le SEG_{d-1} = F_{i,k,d-1}$. Then the volume of material transferred from batch i to segment d is limited by $W_{i,k-1,d-1}$. In many cases, it is possible that batch i receives material from either input node $d-1$ ($Vb_{i,k,d-1}$) or segment $d-2$ ($SD_{i,k,d-1}$) while some portion of batch i is inserted in segment d. Thus,

$$SD_{i,k,d} \le W_{i,k-1,d-1} + Vb_{i,k,d-1} + SD_{i,k,d-1}, \quad \forall\ i \in I_{d-1},\ k \in K(k \ge 1),\ d \in D(d \ge 2). \tag{28}$$

Let us assume that composite run k discharges the batch $i \in I_d$ to output terminal d. No material can be inserted from batch $i+1$ to segment $d+1$ because the pumping operation ends when one of the product deliveries is completed.

$$SD_{i+1,k,d+1} \le Vb_{d+1}^{\max}(1 - x_{i,k,d}), \quad \forall\ i \in I_d,\ k \in K(k \ge 1),\ d \in D(d < |D|). \tag{29}$$

No material can be transferred from segment $(d-1)$ to segment d during the execution of the composite run k when a pumping run belonging to k inserts some materials to segment d from input node d. Thus,

$$\sum_{i \in I_{d-1}} SD_{i,k,d} \le Vb_d^{\max}\left(1 - \sum_{i \in I_d} w_{i,k,d}\right), \quad \forall\ k \in K(k \ge 1),\ d \in D(d \ge 2). \tag{30}$$

4.10. Mass balance constraint

Because of the liquid incompressibility assumption (A2), when some batches are injected in a segment, the same volume should be discharged from that segment. Hence,

$$\sum_{i \in I_d} Vb_{i,k,d} + \sum_{i \in I_{d-1}} SD_{i,k,d} = \sum_{i \in I_d} VD_{i,k,d} + \sum_{i \in I_d} SD_{i,k,d+1}, \quad \forall\ i \in I_d,\ k \in K(k \ge 1),\ d \in D. \tag{31}$$

4.11. Supplying material from input nodes

Through composite run k, a batch i in segment d can receive material from input node d only if: (a) at time S_k, the lower coordinate of batch i (i.e., $F_{i+1,k-1,d} = F_{i,k-1,d} - W_{i,k-1,d}$) touches the origin of segment d, meaning that the value of variable $F_{i+1,k-1,d}$ should be equal to zero, and (b) at time S_k, the frontal coordinate of batch i in segment $d-1$ (i.e., $F_{i,k-1,d-1}$) has reached the origin of segment d. Both conditions can be written by the following disjunction:

$$\underline{\bigvee}_{i \in I_d}\begin{bmatrix} w_{i,k,d} \\ F_{i,k-1,d} - W_{i,k-1,d} = 0 \\ F_{i,k-1,d-1} \ge SEG_{d-1} \end{bmatrix}, \quad \forall\ k \in K(k \ge 1),\ d \in D.$$

Using the big-M reformulation we get:

$$F_{i,k-1,d} - W_{i,k-1,d} \le SEG_d(1 - w_{i,k,d}), \quad \forall\ i \in I_d,\ k \in K(k \ge 1),\ d \in D, \tag{32}$$

$$F_{i,k-1,d-1} \ge SEG_{d-1} w_{i,k,d}, \quad \forall\ i \in I_d,\ k \in K(k \ge 1),\ d \in D(d \ge 2). \tag{33}$$

Remark 4. Always batch i (empty or non-empty) will move between two batches $i-1$ and $i+1$ along the pipeline. Therefore, just batch i can be injected between batches $i-1$ and $i+1$ touching each other at the junction. Such a problem feature is also considered in constraints (32) and (33).

4.12. Supplying material to output terminals

During composite run k, output terminal d can receive material from batch $i \in I_d$ only if the following two conditions are satisfied: (a) the lower coordinate of batch i in segment d at time S_k (i.e., $F_{i+1,k-1,d}$) has not reached the origin of segment d, and (b) before starting run k, the upper coordinate of batch i in segment d (i.e., $F_{i,k-1,d}$) has already reached the origin of segment d. Both conditions can be written by the following disjunction:

$$\underline{\bigvee}_{i \in I_d}\begin{bmatrix} x_{i,k,d} \\ F_{i,k-1,d} - W_{i,k-1,d} \le SEG_d \\ F_{i,k-1,d} \ge SEG_d \end{bmatrix}, \quad \forall\ k \in K(k \ge 1),\ d \in D.$$

Using the big-M reformulation we have:

$$F_{i,k-1,d} - W_{i,k-1,d} \le SEG_d(2 - x_{i,k,d}), \quad \forall\ i \in I_d,\ k \in K(k \ge 1),\ d \in D, \tag{34}$$

$$F_{i,k-1,d} \ge SEG_d x_{i,k,d}, \quad \forall\ i \in I_d,\ k \in K(k \ge 1),\ d \in D. \tag{35}$$

Let us assume that the output terminal should receive material from the pipeline through pumping run k and that the coordinates of batch i in segment d at the end of pumping run k satisfy the condition: $F_{i,k-1,d} - W_{i,k-1,d} \le SEG_d = F_{i,k-1,d}$. Then, the maximum size that can be transferred from batch i to output terminal d is given by

$$VD_{i,k,d} + SD_{i,k,d+1} \le W_{i,k-1,d} + SD_{i,k,d} + Vb_{i,k,d}, \quad \forall\ i \in I_d,\ k \in K(k \ge 1),\ d \in D. \tag{36}$$

In fact constraint (36) acts as an upper bound on the material transferred from batch $i \in I_d$ to output terminal d and segment $d+1$.

4.13. Active and stopped volumes

Let 0–1 variable $v_{k,d}$ denote that pipeline segment d is active during composite run k whenever $v_{k,d} = 1$. Otherwise, $v_{k,d} = 0$. During a composite run k, a segment $d \in D$ will be active for two reasons: (a) it may receive some materials from the previous segment $d-1$ or (b) it may receive some materials from input node d. Thus,

$$\sum_{d \in D} v_{0,d} = 0, \tag{37}$$

$$\sum_{i \in I_d} w_{i,k,d} \le v_{k,d}, \quad \forall\, k \in K(k \ge 1),\ d \in D, \tag{38}$$

$$\sum_{i \in I_d} w_{i,k,d} \le v_{k,d} \le v_{k,d-1} + \sum_{i \in I_d} w_{i,k,d}, \quad \forall\, k \in K(k \ge 1),\ d \in D(d \ge 2), \tag{39}$$

$$\left(v_{k,d} - \sum_{i \in I_d} w_{i,k,d} \right) \le \sum_{i \in I_{d-1}} u_{i,k,d} \le |I_d| v_{k,d}, \quad \forall\, k \in K(k \ge 1),\ d \in D(d \ge 2). \tag{40}$$

All the pipeline segments are assumed to be idle at the starting time of planning horizon (37), meaning that for $k = 0$ we have $\sum_{d \in D} v_{k,d} = 0$. On the other hand, segment d will be active through composite run k whenever segment $d+1$ is active and no material is injected from input $d+1$. Such a problem feature is considered in constraint (39). In addition, the first segment receives some materials from the first input node when it is activated and vice versa. Thus

$$\sum_{i \in I_1} w_{i,k,1} = v_{k,1}, \quad \forall\, k \in K(k \ge 1). \tag{41}$$

On the other hand, when the output terminal d is active, segment d connecting nodes $d-1$ and d is active as well. Thus,

$$\sum_{i \in I_d} x_{i,k,d} \le v_{k,d}, \quad \forall\, k \in K(k \ge 1),\ d \in D. \tag{42}$$

Since most of the pumping costs are associated to flow restarts in idle pipeline segments, it is important to minimize the number of pipeline segments where the flow is resumed or stopped. To do so, let continuous variables $AV_{k,d}$ and $SV_{k,d}$ represent the activated and the stopped volumes of pipeline segment d by composite run k, respectively. When the state (idle or active) of a pipeline segment changes in consecutive pumping runs, the flow in that segment must be restored or stopped. When the flow in the segment d is restarted, the value of $AV_{k,d}$ will be equal to the volume of segment d. By identifying the state of segment d in two consecutive runs k and $k-1$, the values of variables $AV_{k,d}$ and $SV_{k,d}$ can be computed by the following disjunction:

$$\underline{\bigvee} \begin{bmatrix} Y_{k,d}^{\text{start}} \\ AV_{k,d} = SEG_d \\ SV_{k,d} = 0 \end{bmatrix} \underline{\bigvee} \begin{bmatrix} Y_{k,d}^{\text{stop}} \\ AV_{k,d} = 0 \\ SV_{k,d} = SEG_d \end{bmatrix} \underline{\bigvee} \begin{bmatrix} Y_{k,d}^{\text{no change}} \\ AV_{k,d} = 0 \\ SV_{k,d} = 0 \end{bmatrix}$$

$$\forall\, k \in K(k \ge 1),\ d \in D.$$

Note that restarting the flow in segment d due to run k is equivalent to saying that the segment is active through run k but inactive during $k-1$. The opposite condition identifies the stop of the pipeline segment. The logic propositions relating the different sets of binary variables are given below:

$$Y_{k,d}^{\text{start}} \Leftrightarrow \neg v_{k-1,d} \wedge v_{k,d}, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$Y_{k,d}^{\text{stop}} \Leftrightarrow v_{k-1,d} \wedge \neg v_{k,d}, \quad \forall\, k \in K(k \ge 1),\ d \in D.$$

The convex hull reformulation of the disjunction and the conversion of the logic expressions give rise to the following constraints:

$$AV_{k,d} = SEG_d.\ Y_{k,d}^{\text{start}}, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$SV_{k,d} = SEG_d.\ Y_{k,d}^{\text{stop}}, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$Y_{k,d}^{\text{start}} + Y_{k,d}^{\text{stop}} \le 1, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$Y_{k,d}^{\text{start}} \le v_{k,d}, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$v_{k-1,d} \le 1 - Y_{k,d}^{\text{start}}, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$v_{k,d} - v_{k-1,d} \le Y_{k,d}^{\text{start}}, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$v_{k,d} \le 1 - Y_{k,d}^{\text{stop}}, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$Y_{k,d}^{\text{stop}} \le v_{k-1,d}, \quad \forall\, k \in K(k \ge 1),\ d \in D,$$

$$v_{k-1,d} - v_{k,d} \le Y_{k,d}^{\text{stop}}, \quad \forall\, k \in K(k \ge 1),\ d \in D.$$

We can now eliminate variables $Y_{k,d}^{\text{start}}$ and $Y_{k,d}^{\text{stop}}$ from the formulation by combining some of the constraints, giving rise to the following equations:

$$AV_{k,d} \ge SEG_d(v_{k,d} - v_{k-1,d}), \quad \forall\, k \in K(k \ge 1),\ d \in D, \tag{43}$$

$$SV_{k,d} \ge SEG_d(v_{k-1,d} - v_{k,d}), \quad \forall\, k \in K(k \ge 1),\ d \in D, \tag{44}$$

$$AV_{k,d} + SV_{k,d} \le SEG_d, \quad \forall\, k \in K(k \ge 1),\ d \in D, \tag{45}$$

To avoid the unnecessary flow restarts and stoppages in the pipeline segments, variables $AV_{k,d}$ and $SV_{k,d}$ will be penalized in the objective function.

4.14. Flow rate limitation on pipeline segments

The stream flow rate in active segments should lie within the admissible range. When a segment of pipeline is directly fed by an input node, the flow rate is regulated by the pumping rate constraint i.e., Eq. (9). For a segment of pipeline that receives material from the adjacent segment, the following constraint is imposed:

$$vs_d^{\min} L_k - Vb_d^{\max} \left(1 - v_{k,d} + \sum_{i \in I_d} w_{i,k,d} \right) \le \sum_{i \in I_{d-1}} SD_{i,k,d} \le vs_d^{\max} L_k, \quad \forall\, k \in K(k \ge 1),\ d \in D(d \ge 2), \tag{46}$$

where interval $[vs_d^{\min}, vs_d^{\max}]$ is the admissible flow rate range in segment d.

4.15. Interface material between two consecutive batches

Since there is no physical barrier between batches inserted into the line, there is always some mixing at the interface. The interface volume between products p and p' usually depends on the physicochemical properties of these products and is assumed to be constant $MIX_{p,p'}$. Let continuous variable $INV_{i,p,p'}$ be the interface volume between two adjacent batches $(i-1)_{p'}$ and i_p. We have:

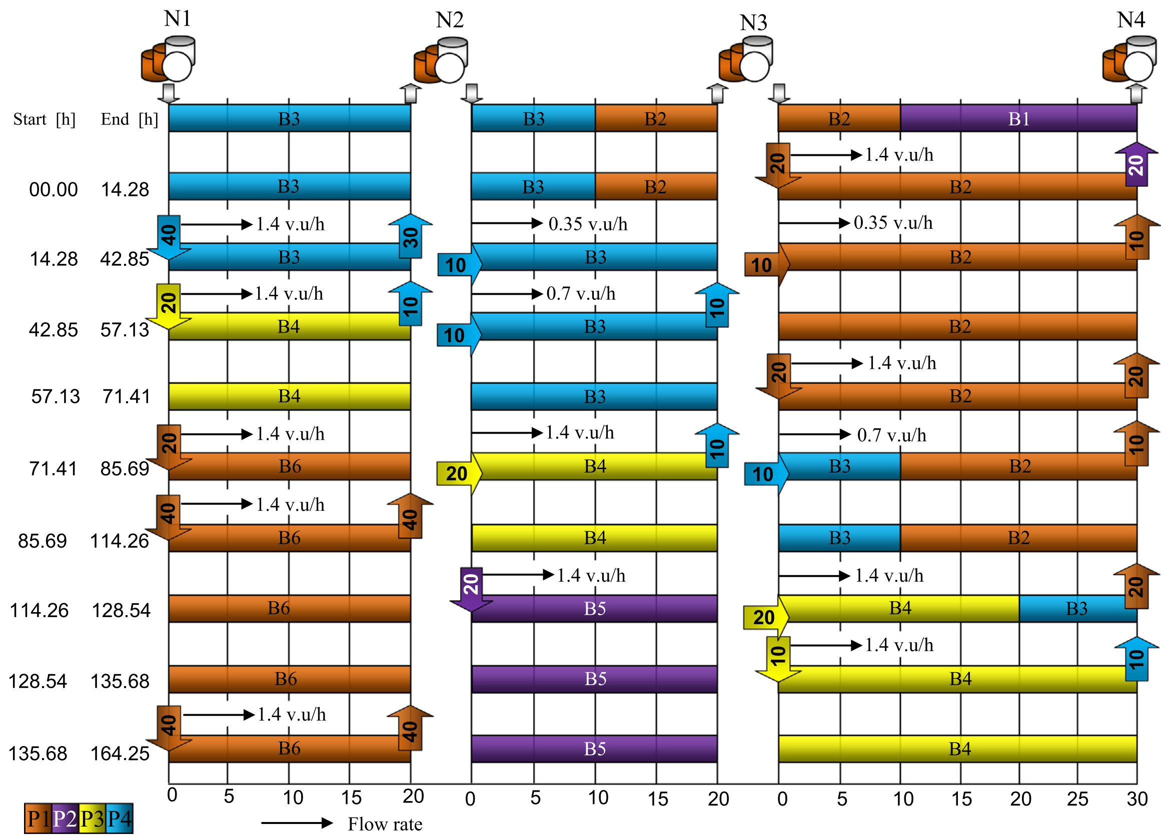

Fig. 6. Best pipeline scheduling for Example 1 with non-simultaneous injections [29].

Table 1
List of notation.

Index/sets	Description
$k, k' \in K$	Set of composite runs
$i, i' \in I$	Set of product batches $(I = \bigcup_{d \in D} I_d)$
I^{new}	Set of new batches to be injected into the pipeline $(I^{new} \subset I)$
$d, d' \in D$	Set of segments
I_d^{old}	Set of old batches in segment d
I_d^{new}	Set of new batches to be injected in segment d
I_d	Set of batches in segment d $(I_d = I_d^{new} \cup I_d^{old})$
$p, p' \in P$	Set of oil products

$$\bigvee_{p \in P} \bigvee_{p' \in P} \begin{bmatrix} Y_{i,p,p'}^{mix} \\ INV_{i,p,p'} = MIX_{p,p'} \end{bmatrix}, \quad \forall\, i \in I,$$

$$y_{i+1,p} \wedge y_{i,p'} \Rightarrow Y_{i,p,p'}^{mix}, \quad \forall\, i \in I\,(i < |I|),\ p,\ p' \in P,$$

Combining the constraints resulting from the convex hull reformulation of the disjunction and the reformulation of the logic proposition leads to the following equation:

$$INV_{i,p,p'} \ge MIX_{p,p'}(y_{i+1,p} + y_{i,p'} - 1) \quad \forall\, i \in I\,(i < |I|),\ p,\ p' \in P, \tag{47}$$

Because of the high product contamination caused by the wide interface, some product sequences inside the pipeline are forbidden. Let $Seq_{p,p'}$ be a 0–1 matrix of possible sequences between products p and p' ($Seq_{p,p'} = 1$, if the sequence is allowed). If the sequence is not allowed, it means that products p and p' cannot be present in consecutive batches:

Table 2
List of parameters.

Parameters	Description
h_{max}	Time horizon length (h)
ST	Starting time of the first composite run (h)
vs_d^{min}/vs_d^{max}	Minimum/maximum flow rate in the segment d (m^3/h)
vd_d^{min}/vd_d^{max}	Minimum/maximum injection rate at the input node d (m^3/h)
$vp_{p,d}$	Maximum delivery rate of product p to output terminal d (m^3/h)
SEG_d	Volume of segment d (m^3)
Vb_d^{min}/Vb_d^{max}	Minimum/maximum batch sizes injected from the input node d (m^3)
VD_d^{min}/VD_d^{max}	Minimum/maximum batch sizes diverted to the output node d (m^3)
$Seq_{p,p'}$	Boolean matrix of possible sequences between products p and p'
$MIX_{p,p'}$	Size of the interface between products p and p' (m^3)
$IW_{i,d}$	Size of old batch $i \in I_d^{old}$ in segment d (m^3)
$dem_{p,d}$	Demand for product p at output terminal d (m^3)
$Inv_{p,d}$	Inventory of product p at input terminal d (m^3)
$CP_{p,d}$	Pumping cost per unit of product p pumped from input node d (\$/m^3)
$CI_{p,p'}$	Cost of reprocessing a unit interface volume between products p and p' (\$/m^3)
$CB_{p,d}$	Unit backorder cost for tardily satisfying product p at output node d (\$/ m^3)
FC	Fixed cost for performing a pumping run (\$/run)
CA	Unit flow restart cost (\$/m^3)
CS	Unit flow stoppage cost (\$/m^3)

$$\neg Seq_{p,p'} \Rightarrow \neg\,(y_{i,p} \wedge y_{i-1,p'}), \quad \forall\, i \in I,\ p,\ p' \in P,$$

which can be reformulated into:

Table 3
List of variables.

Variable	Description
S_k	Starting time of composite pumping run k (h)
L_k	Length of composite pumping run k (h)
$LS_{k,d}$	Length of pumping run k at input node d (h)
$Vb_{i,k,d}$	Volume of batch $i \in I_d$ ($i \geq end(I_d^{old})$) injected from input node d during composite run k (m^3)
$VP_{i,p,k,d}$	Volume of batch $i \in I_d$ ($i \geq end(I_d^{old})$) containing product p injected from input node d during composite run k (m^3)
$VD_{i,k,d}$	Volume of batch $i \in I_d$ diverted to output node d during composite run k (m^3)
$VPD_{i,p,k,d}$	Volume of batch $i \in I_d$ containing product p diverted to output node d during composite run k (m^3)
$SD_{i,k,d}$	Size of batch $i \in I_{d-1}$ ($d > 1$) transferred from segment $d-1$ to segment d during composite run k (m^3)
$W_{i,k,d}$	Size of batch $i \in I_d$ in segment d at the end of composite run k (m^3)
$F_{i,k,d}$	Upper coordinate of batch $i \in I_d$ in segment d at the end of composite run k (m^3)
$INV_{i,p,p'}$	Interface volume between batches $i-1$ and i when they convey products p and p' (m^3)
$unsdem_{p,d}$	Unsatisfied demand of product p at output terminal d (m^3)
$AV_{k,d}$	Activated volume of segment d through composite run k (m^3)
$SV_{k,d}$	Stopped volume of segment d through composite run k (m^3)
$w_{i,k,d}$	1 if a portion of batch $i \in I_d$ ($i \geq end(I_d^{old})$) is injected from segment d through composite run k
$x_{i,k,d}$	1 a portion of batch $i \in I_d$ is diverted to output terminal d through composite run k
$u_{i,k,d}$	1 if a portion of batch $i \in I_{d-1}$ ($d > 1$) is transferred from segment $d-1$ to segment d through composite run k
$v_{k,d}$	1 if segment d is active through composite run k
$y_{i,p}$	1 if batch i conveys product p

$$y_{i,p} + y_{i-1,p'} \leq Seq_{p,p'} + 1, \quad \forall\, i \in I,\ p, p' \in P. \tag{48}$$

4.16. Meeting product demand

The main purpose of multi-product pipeline scheduling is to satisfy depot requirements at the right time. To meet the demand of product p at receiving terminal d, the total amount discharged during the time horizon should be as large as $dem_{p,d}$. Otherwise, slack variable $unsdem_{p,d}$, representing the backorder of product p at the receiving terminal d during the time interval $[0, h_{\max}]$, is activated by constraint (49) resulting in operational costs.

$$dem_{p,d} \leq \sum_{i \in I_d} \sum_{k \in K} VPD_{i,p,k,d} + unsdem_{p,d}, \quad \forall\, p \in P,\ d \in D. \tag{49}$$

4.17. Objective function

The objective function is to minimize the operational charges including pumping, interface, backorder, restarts and stoppages, and on/off pumping switching costs:

$$\begin{aligned}\min z_1 = {} & \sum_{d \in D} \sum_{k \in K} \sum_{p \in P} \sum_{i \in I_d} CP_{p,d} \times VP_{i,p,k,d} \\ & + \sum_{i \in I} \sum_{p \in P} \sum_{p' \in P} CI_{p,p'} \times INV_{i,p,p'} \\ & + \sum_{d \in D} \sum_{p \in P} CB_{p,d} \times unsdem_{p,d} + \sum_{k \in K} FC \times RUN_k \\ & + \sum_{d \in D} \sum_{k \in K} (CA \times AV_{k,d} + CS \times SV_{k,d}),\end{aligned} \tag{50}$$

where RUN_k is a binary variable indicating the existence of composite run k and its value satisfies the following Eq:

$$RUN_k \geq \frac{1}{|D|} \left(\sum_{d \in D} \sum_{i \in I_d} w_{i,k,d} \right), \quad \forall\, k \in K,$$

For better utilization of the pipeline capacity, it is desirable to minimize the makespan i.e., the amount of time required for meeting all product deliveries. It is to say that:

$$\min z_2 = \sum_{k \in K} L_k. \tag{51}$$

Remark 5. In all examples, the minimum operational cost (z_1) will be chosen as the primary target and the minimum makespan (z_2) as the secondary one. After solving the problem formulation with the primary target, we fix z_1 and then solve the model with the secondary target. To this end, the CPU time reported in computational sections will be the total time of the two optimization runs.

5. Results and discussion

Five example problems are now solved using the proposed mathematical model. Results are compared to other approaches from the literature. Specific features of the proposed model are highlighted in the following subsections. All MILP models were implemented in AIMMs 3.9 and solved using CPLEX 12.1 running in parallel deterministic mode (using up to 6 threads) on an Intel i5-4210U (2.7 GHz) CPU with 6 GB of RAM running Windows 7 (64-bit).

5.1. Single vs. simultaneous injections

5.1.1. Example 1

In our previous work [29], a single input terminal can inject into the pipeline at any given time, which can be classified as a Non-Simultaneous Injection (**NSI**). Note that **NSI** allows for simultaneous deliveries to multiple output terminals with the run of the input terminal. In contrast, the model proposed in this paper allows for simultaneous injections and deliveries (**SID**). In order to show the advantages of **SID** over **NSI**, consider a problem with three pipeline segments that convey four refined products (P1–P4) from three input terminals (N1–N3) to three output terminals (N2–N4). The first line of Fig. 6 depicts pipeline topography and its initial state at time $ST=0$.

Table 4
Supply, demand and cost for Example 1.

P	Supply (u.v.)			Demand (u.v.)			Pump cost ($/u.v.)			Interface volume (u.v.)/cost ($)			
	N1	N2	N3	N2	N3	N4	N1	N2	N3	P1	P2	P3	P4
P1	100	–	40	80	0	60	15	–	8	0	6/250	6/250	8/350
P2	–	20	–	–	–	20	–	10	–	6/250	–	7/300	9/400
P3	20	–	10	–	–	–	20	–	15	6/250	7/300	–	5/200
P4	40	–	–	40	20	10	25	0	0	8/350	9/400	5/200	0

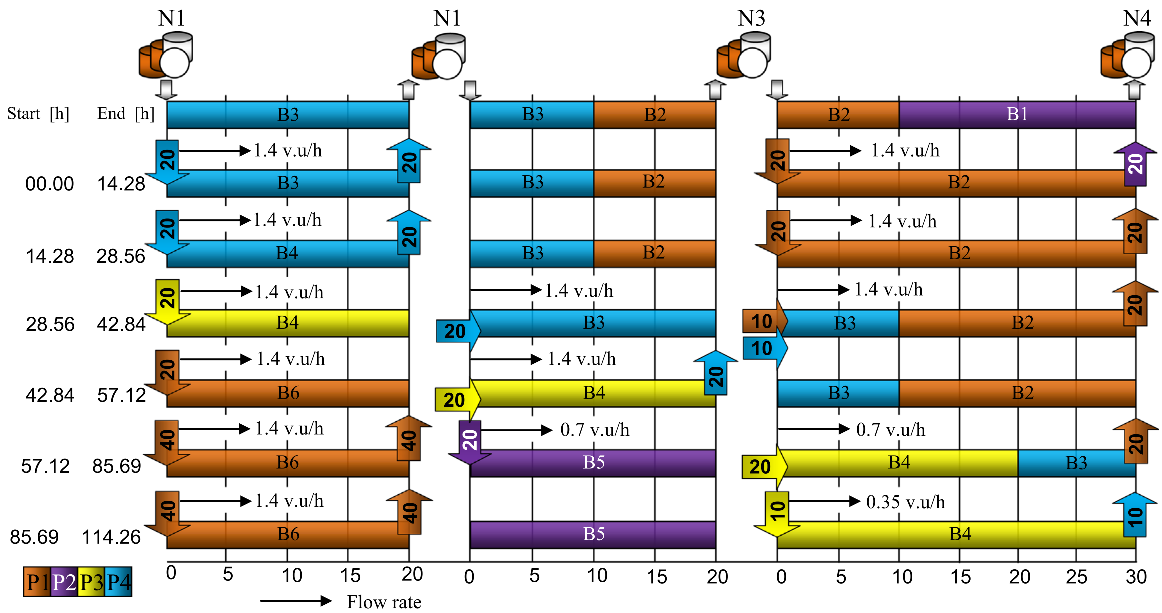

Fig. 7. Optimal pipeline scheduling for Example 1.

Table 5
Computational results of Example 1.

Case	Runs |K|	CPUs	Cont. var	Bin. var	Eq.	Makespan (h)	Back order (%)	Obj_{-Relax}	Obj_{-MILP} ($)	Integrality gap
NSI	9	62.67	1960	402	3472	164.25	0.0	7806	12,470	0.374
SID	5	10.03	1058	227	1805	114.26	8.69	7870	14,170	0.444
SID	**6**	**19.28**	**1242**	**267**	**2110**	**114.26**	**0.0**	**7870**	**12,470**	**0.368**
SID	7	24.23	1426	307	2415	114.26	0.0	7870	12,470	0.368

Table 6
Product supplies, demands, related pumping and interface costs for Example 2.

P	Supplies (10 × m^3)				Demands (10 × m^3)				Pump cost ($/$m^3$)				Interface volume (m^3)/cost (10^2 × $)					
	N1	N2	N3	N4	N2	N3	N4	N5	N1	N2	N3	N4	P1	P2	P3	P4	P5	P6
P1	600	200	–	–	–	400	–	200	15	12	–	–	0	1/34	×	×	1/28	1/64
P2	–	–	100	–	–	–	200	500	–	–	5	–	1/40	0	1/35	1/35	1/28	×
P3	400	–	–	–	200	–	500	100	20	–	–	–	×	1/36	0	1/20	1/60	1/54
P4	–	200	200	–	–	–	100	300	–	18	8	–	×	1/54	1/22	0	×	1/35
P5	400	–	–	200	–	–	–	200	10	–	–	6	1/36	1/28	1/36	×	0	×
P6	700	–	100	300	400	–	–	–	15	–	10	15	1/18	×	1/36	1/45	×	0

Product supplies and demands at pipeline terminals, pumping cost and interface cost between two pair of products are listed in Table 4. The horizon length is 168 h and the flow rate at every pipeline segment should not surpass 1.4 v.u./h. Besides, there are two additional limitations: (a) any injection can have a maximum size of 40 v.u. and (b) the product delivery to any receiving terminal can have a minimum size of 10 v.u. Moreover, the following values for the coefficients arising in objective function have been adopted: $CB_{p,d} = 200\$/\text{v. u}$, $FC = CA = CS = 0$.

The best pipeline schedule for **NSI** [29] is given in Fig. 6. It includes nine pumping operations with three of them diverting products to different terminals at the same time (see third, fourth and sixth lines). The pipeline works uninterruptedly at the maximum production rate of 1.4 (u.v./h). There are a total of 12 product deliveries into receiving terminals (represented by upward-pointing arrows) over 164.25 h, 3.75 h short of the planning horizon.

The optimal solution for **SID** given in Fig. 7 shows a much more efficient use of the pipeline, with just 114.26 h being required to meet the demand, a 2-day reduction compared to **NSI**. Note that both solutions feature the same value of the objective function ($12,470), with makespan being an additional key performance indicator (secondary objective, recall Remark 5). The computational statistics for Example 1 can be found in Table 5. Since the number of composite runs needs to be specified (the cardinality of set K) and we do not know a priori the number leading to the optimal solution, we start with $|K| = 5$ only to find out that the returned solution does not completely satisfy demand (back order=8.69%). More specifically, there are shortages of 10 v.u. of P1 and P4 at output terminal N4. This is no longer the case for $|K| = 6$, leading to the optimal solution. No further improvements are observed for $|K| = 7$.

Primarily due to the need of a lower value of $|K|$ for finding the optimal solution, the proposed MILP model has 40% fewer equations, 37% fewer continuous variables and 33% fewer 0–1 variables, compared to the model in [29]. There is also a slight reduction in integrality gap from 37.4% to 36.8%. It is thus not surprising that the computational time has been decreased by a factor of 3 (from 62.67 to 19.28 CPUs).

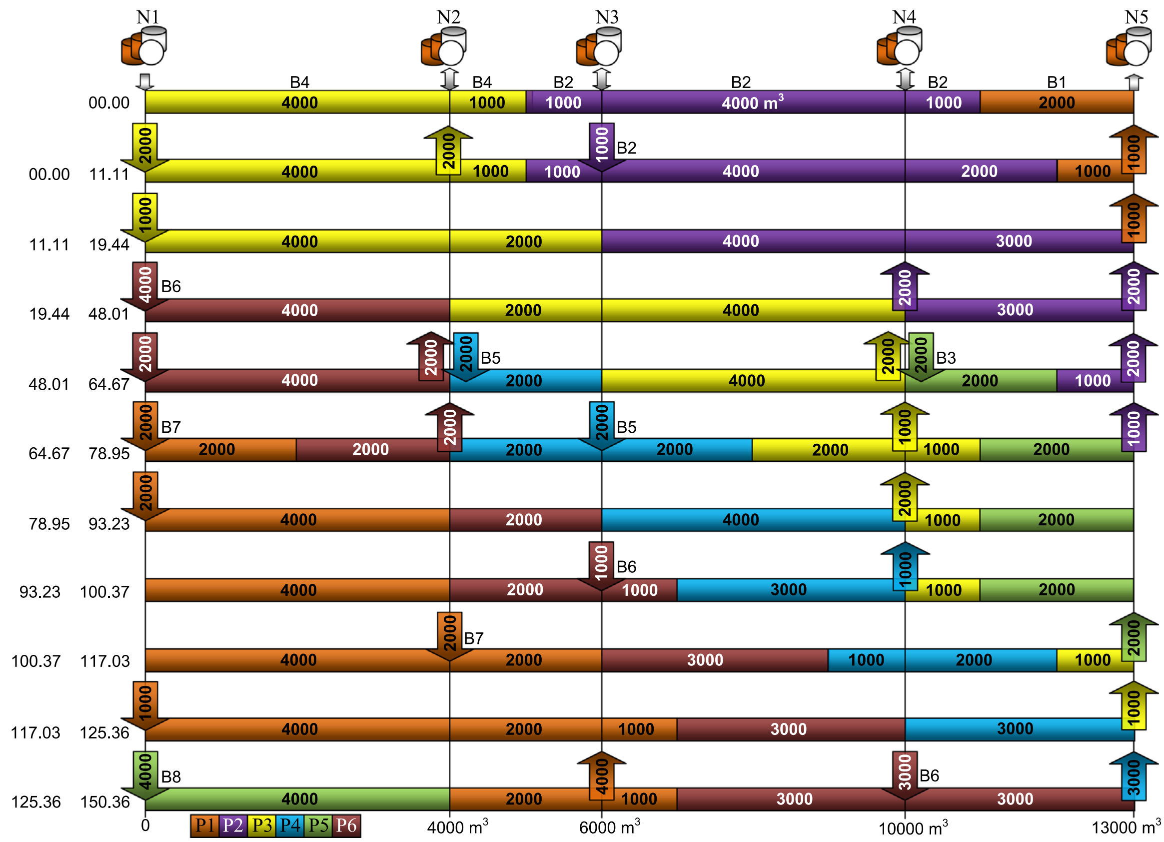

Fig. 8. Optimal pipeline schedule for Example 2 using the proposed model.

Table 7
Computational results of Example 2.

Case	Runs |K|	CPUs	Cont. var	Bin. var	Eq.	Opt. gap (%)	Back order (%)	Obj$_{-Relax}$	Obj$_{-MILP}$ ($)	Int. gap	Solver status	Program status
[29]	12	3465.21	5622	912	7652	0.0	4.83	399,915.0	572,850	0.301	NC[a]	Opt.[b]
[29]	13	10,856.40	6068	984	8220	0.0	0.0	399,865.5	437,900	0.086	NC	Opt.
[29]	14	18,000.00	6514	1056	8788	12.1	9.67	399,804.1	701,860	0.430	RI[c]	IS[d]
Our	9	468.21	3992	696	5314	0.0	0.0	404,402.0	437,900	0.076	NC	Opt.
Our	10	782.45	4426	768	5832	0.0	0.0	404,401.6	432,900	0.065	NC	Opt.
Our	11[e]	1069.21	4860	840	6350	0.0	0.0	404,401.3	432,900	0.065	NC	Opt.

[a] Normal completion.
[b] Optimal.
[c] Resource interrupt.
[d] Integer solution.
[e] One of the pumping operations is dummy at the optimum.

Table 8
Product supplies, demands, related pumping and interface costs for Example 3.

P	Supplies (m^3)		Demands (m^3)			Pump cost ($/$m^3$)		Interface volume (m^3)/cost (10^2 $)					
	N1	N3	N2	N3	N4	N1	N3	P1	P2	P3	P4	P5	P6
P1	26,000	–	5700	4800	17,500	1.735	–	0	1/42	1/24	1/32	1/31	1/26
P2	9800	10,800	–	–	9200	2.125	1.840	1/42	0	×	1/31	1/36	1/31
P3	18,100	6200	5600	–	7700	1.735	1.735	1/24	×	0	1/48	1/25	1/21
P4	18,200	–	6200	7300	7400	2.456	–	1/32	1/31	1/48	0	1/28	1/34
P5	16,500	5000	8200	–	8600	2.628	2.320	1/31	1/36	1/25	1/28	0	1/40
P6	8200	11,600	4700	–	7400	1.640	1.245	1/26	1/31	1/21	1/34	1/40	0

5.1.2. *Example 2*

A unidirectional pipeline transporting six products within a 200 h time horizon from four input terminals to four output terminals composes the system configuration for Example 2. Data for Example 2 are listed in Table 6. The admissible flow rate ranges in pipeline segments, given in (m^3/h), are [40, 180] for the first segment, [30, 160] for the second segment, [30, 140] for the third segment and [30,120] for the last one. Any shipment pumped from an input terminal can have a minimum size of 1000 m^3 and a maximum size of 4000 m^3. To avoid the generation of a higher number of interfaces, the minimum delivery size is set to 1000 m^3. Moreover, the following values for the parameters in the objective

Table 9
Computational results for Example 3.

Case	Runs |K|	CPUs	Cont. var	Bin. var	Eq.	Opt. gap (%)	Back order (%)	Makespan (h)	Obj_{-Relax}	Obj_{-MILP} ($)	Integrality gap
[29]	14	4021.30	4595	752	6656	0.0	2.79	–	178,814.5	806,511.4	0.778
[29]	15	6759.55	4896	802	7147	0.0	0.0	286.97	178,792.1	250,511.4	0.286
[29]	16	10,800.00	5197	852	7638	4.40	8.52	–	178,723.2	1,241,578.5	0.856
Our	12	683.54	3673	664	5176	0.0	0.69	–	186,067.9	374 687.9	0.503
Our	**13**	**1361.03**	**3965**	**715**	**5639**	**0.0**	**0.0**	**221.47**	**185,961.8**	**235,044.4**	**0.208**
Our	14[a]	2130.09	4257	766	6102	0.0	0.0	221.47	185870.9	235044.4	0.208

[a] One of the pumping operations is dummy at the optimum.

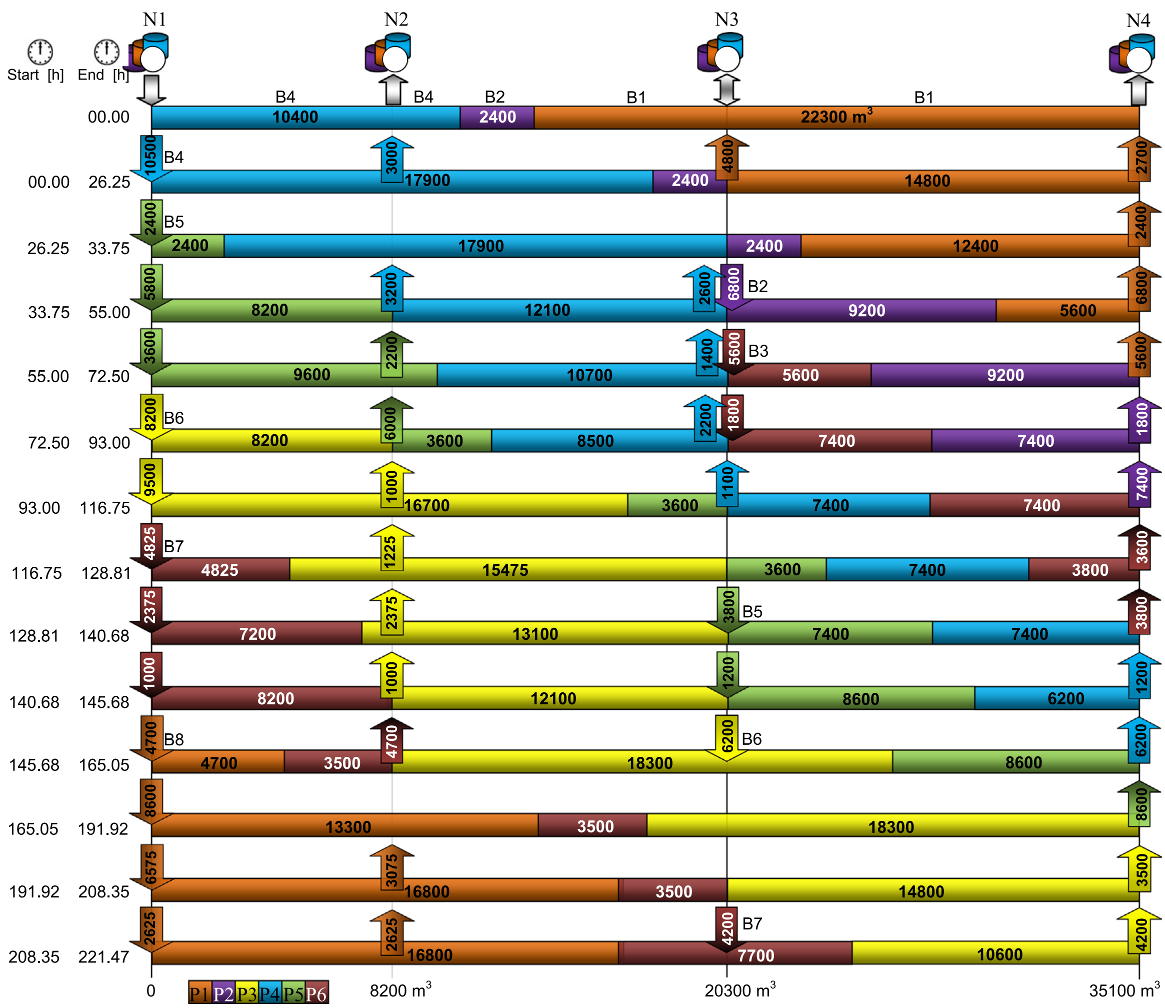

Fig. 9. Optimal pipeline schedule for Example 3 using the proposed model.

Table 10
Product supplies, demands, related pumping and interface costs for Example 4.

P	Supplies (m^3)		Demands (m^3)			Pump cost ($/$m^3$)		Interface volume (m^3)/cost (10^2 $)				
	N1	N3	N2	N3	N4	N1	N3	P1	P2	P3	P4	P5
P1	25,884	4548	5110	–	12,180	8.136	3.042	0	1/36	1/36	1/45	1/27
P2	167,772	–	41,340	39,210	27,700	9.492	–	1/36	0	×	1/27	1/50
P3	5316	21,384	–	–	17,820	10.170	4.056	1/36	×	0	×	1/35
P4	9468	–	3090	2120	1100	8.814	–	1/45	1/21	×	0	1/45
P5	20,532	30,876	–	–	37,170	8.814	3.380	1/23	1/51	1/30	1/54	0

function have been used: $CB_{p,d} = 200$ $/m^3, $FC = CA = CS = 0.0$.

The optimal solution for Example 2 is shown in Fig. 8. It features 15 pumping runs arranged in 10 composite runs and a makespan equal to 150.36 h. With non-simultaneous injections [29] the makespan is considerably higher (208.90 h) even though the difference in total operating cost is just 1.2% higher. The results in Table 7 show that both models can generate the suboptimal solution of $437,900, but the proposed model requires 4 less

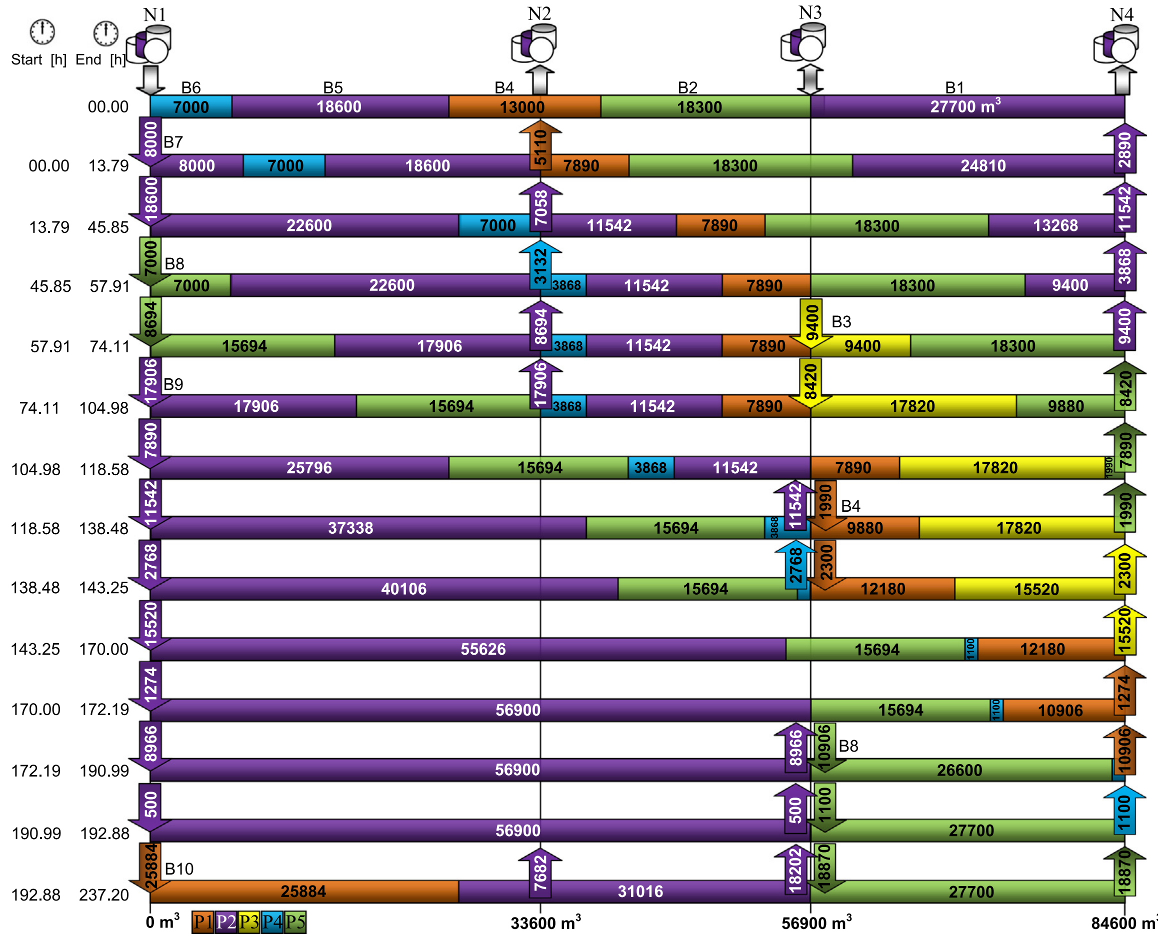

Fig. 10. Best detailed schedule for Example 4 obtained by the proposed approach.

Table 11
Computational results of Example 4.

Case	Runs |K|	CPU (s)	Cont. var	Bin. var	Eq	Makespan (h)	Active .v (10^2 m^3)	Stop. v (10^2 m^3)	ON/OFF cost($)	Obj_a [a] ($)	Obj_d [b] ($)	Obj. Fun ($)
[27]	8	687.4	2504	450	4605	240.0	–	–	–	1,454,700	–	–
[40]	19	0.107	1199	116	2190	240.0	2538	1692	19,000	–	44380	–
Our	13	874.23	3707	783	5704	237.2	1079	233	13,000	1,421,000	23,790	1,478,490

[a] The cost of pumping, interface and backorder at the optimum.
[b] The cost of pipeline flow restarts/stoppages and ON/OFF pump switchings at the optimum.

composite runs. As a consequence of the smaller problem size and lower integrality gap, the computational time has been reduced by a factor of 23. A single increment in the value of |K| causes the model in [29] to return a very poor solution of $701,860 (9.67% backorder) up to the maximum computational time of 18,000 CPUs, whereas the new model can find the optimal solution in just 782.45 CPUs.

5.1.3. Example 3

Example 3 is a real-world case study involving a portion of the Iranian pipeline system featuring a length of 345 km and a total volume of 35,100 m^3. It concerns the distribution of six commodities (P1–P6) from two sources (N1, N3) to three receiving terminals (N2–N4) over a 12-day (288 h) time horizon. Data for this example have been received from the National Iranian Oil Pipeline and Telecommunication Company (NIOPTC) and are given in Table 8. The maximum injection rate at node N1 is 400 m^3/h, whereas it is 320 m^3/h for the dual purpose node N3. The acceptable flow rate ranges at the pipeline segments, given in m^3/h, are: [80, 400] for segments N1–N2 and N2–N3 and [80, 320] for the last segment N3–N4, where the pipeline diameter is shorter. Additional strategic constraints should be considered: (a) a maximum delivery rate of 200 m^3/h for products P1, P3 and P4 at depot N2 and (b) a maximum delivery rate of 320 m^3/h for product P4 at N3. Besides, the maximum size of a batch injection is equal to 20,000 m^3, and the minimum size for a product delivery is fixed at 1000 m^3 to reduce the number of interfaces. Unit restart cost is 0.2 \$/m^3 for any segment, each composite run has a fixed cost of $1200, while backorder has a unit cost of 200 \$/m^3.

Solving the proposed MILP model for $|K| = 12$ we obtain a solution worth $374,687.9 in 683.54 CPUs, see Table 9. By increasing the cardinality of set K to 13 we get to the optimal schedule shown in Fig. 9, which already meets all product demands (cost = $235,044.4). The model was also solved with $|K| = 14$ to confirm solution optimality, with the computation time increasing from 1361.03 to 2130.09 CPUs. In fact, the composite run $k=14$ is a dummy operation at the optimum i.e., $\sum_{d\in D}\sum_{i\in I_d} w_{i,14,d} = 0$.

Concerning the comparison to the model in [29], the same behavior is observed for the latter: (i) inability to generate the optimal solution; (ii) requirement of a larger number of composite runs; (iii) significantly longer makespan.

5.2. Monolithic vs. two-level approach

5.2.1. Example 4

The aim of Example 4 is to show the advantages of the proposed single level framework against the two-level decomposition approach developed by Cafaro et al. [40]. It is a real-world case study [27] involving the scheduling of a multiproduct pipeline of some 1000 km in length, transporting five oil derivatives from two sources (N1, N3) to three receiving terminals (N2–N4). The maximum flow rate at the pipeline segments is 580 m^3/h and the horizon length is 240 h. Product demands at the receiving terminals, product inventories at source node, pumping cost at each input nodes and interface cost between batches of different products are listed in Table 10. Product sequences denoted with an (×) in Table 10 are forbidden. For example, product P3 cannot be injected immediately before or after neither P2 nor P4. Similar to the usage of Cafaro et al. [40], the following parameters have been used: $CB_{p,d}$ = 200 \$/m^3, FC = 1000 \$/run, CA = 0.10 \$/m^3 and CS = 0.

The best output schedule reported in [40] includes 19 composite runs and 29 product deliveries into receiving terminals. In contrast, the optimal schedule obtained by our model (Fig. 10) features 13 composite runs and 24 product deliveries. One interesting observation is that there is just one flow restart in segments N1–N2 and N3–N4, and just two flow resumes in the middle segment N2–N3. As a consequence, the pipeline activated volume at the optimum amounts to 107,900 m^3 compared with 253,800 m^3 reported in [40]. From the computational results in Table 11, one can see that the new schedule is about 46.4% less expensive than the one in [40] for the detailed level, corresponding to the savings of (i) 31% in ON/OFF pump operations; (ii) 57% in segment restarting costs; (iii) 86% in stopped volume. In contrast to Cafaro and co-workers models [27,40], the sizing and sequencing variables of our proposed model now need to be defined for each pipeline segment, thus leading to a larger problem size and CPU. Nevertheless, the CPU time required to find the optimal solution by the new formulation remains quite reasonable.

6. Concluding remarks and future scope

This paper has proposed a novel MILP formulation for the detailed scheduling of unidirectional non-branched pipelines employing a continuous representation of time and volume. Multiple terminals consisting of single or dual-purpose stations can be considered, with the model being able to manage injections and deliveries taking place simultaneously at different input and output terminals. In addition, the model can handle lower maximum flow rate constraints in downstream pipeline segments, emanating from a lower diameter. Contrarily to previous approaches performing the pipeline operational planning in two steps, the proposed continuous-time formulation seeks to find both the best sequence of product injections and dispatching operations in a single step. The objective function has been to minimize total operating cost, featuring a variety of terms and including segment restart and stoppage costs. The accuracy and robustness of the proposed model are tested with four case studies, two of them based on real life industrial data. The computational results show the capability of proposed MILP model to solve real-life multiproduct pipeline operational problems in quite reasonable CPU times.

Future work will focus on generalizing the proposed formulation to more complex pipeline systems and on extending the operating modes at station junctions so that a pipeline segment can simultaneously receive products from its input terminal and its feeding segment.

Acknowledgements

Hossein Mostafaei and Alireza Ghaffari-Hadigheh fully appreciate financial support from Azarbaijan Shahid Madani University. Pedro Castro gratefully acknowledges financial support from Fundação para a Ciência e Tecnologia through program Investigador FCT 2013.

References

[1] J.A. Fabro, S.L. Stebel, D. Rossato, H.L. Polli, L.V.R. Arruda, F. Neves Jr, et al., A MILP (Mixed Integer Linear Programming) decomposition solution to the scheduling of heavy oil derivatives in a real-world pipeline, Comput Chem Eng 66 (2014) 124–138.
[2] D.C. Cafaro, J. Cerdá, Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation, Comput Chem Eng 28 (2004) 2053–2068.
[3] C.J. Trench, How pipelines make the oil market work their networks, operation and regulation, Allegro Energy Group, New York, 2001.
[4] Milidiu RL, Liporace FDS. Planning of pipeline oil transportation with interface restrictions is a difficult problem. 2003 PUC-Rio Inf.MCC56/03.
[5] R.Z. Rros-Mercadoa, S. Kimb, E.A. Boyd, Efficient operation of natural gas transmission systems: a network-based heuristic for cyclic structures, Comput Oper Res 33 (2006) 2323–2351.
[6] M. Sasikumar, P.R. Prakash, S.M. Patil, PIPES: a heuristic search model for pipeline schedule generation, Knowl-Based Syst 10 (1997) 169–175.
[7] A. Garcia-Sanchez, L.M. Arreche, M. Ortega-Mier, Combining simulation and tabu search for oil-derivatives pipeline scheduling, Stud Comput Intell 128 (2008) 301–325.
[8] F.M. Mori, R. Luders, L.V.R. Arruda, L. Yamamoto, M.V. Bonacin, H.L. Polli, et al., Simulating the operational scheduling of a real world pipeline network, Comput Aid Chem Eng 24 (2007) 691–696.
[9] C.A. Hane, H.D. Ratliff, Sequencing inputs to multi-commodity pipelines, Ann Oper Res 57 (1995) 73–101.
[10] F. Neves Jr, L. Magatao, S.L. Stebel, S.N. Boschetto, L.C. Felizari, D.I. Czaikowski, et al., An efficient approach to the operational scheduling of a real-world pipeline network, Comput Aid Chem Eng 24 (2007) 697–702.
[11] V.G. Cafaro, D.C. Cafaro, C.A. Mendéz, J. Cerdá, MINLP model for the detailed scheduling of refined products pipelines with flow rate dependent pumping costs, Comput Chem Eng 72 (2014) 210–221.
[12] E.M. De Souza Filho, L. Bahiense, V.J.M. Ferreira Filho, Scheduling a multiproduct pipeline network, Comput Chem Eng 53 (2013) 55–69.
[13] L. Magatao, L.V.R. Arruda, F.A. Neves, A mixed integer programming approach for scheduling commodities in a pipeline, Comput Chem Eng 28 (2004) 171–185.
[14] R. Rejowski, J.M. Pinto, Efficient MILP formulations and valid cuts for multiproduct pipeline scheduling, Comput Chem Eng 28 (2004) 1511–1528.
[15] N. Shah, Mathematical programming techniques for crude oil scheduling, Comput Chem Eng 20 (1996) 1227–1232.
[16] Zyngier D, Kelly JD. Multi-product inventory logistics modeling in the process industries. Optimization and logistics challenges in the enterprise, vol. 30; 2009. p. 61–95.
[17] D.C. Cafaro, J. Cerdá, Efficient tool for the scheduling of multiproduct pipelines and terminal operations, Ind Eng Chem Res 47 (2008) 9941–9956.
[18] S. Relvas, H.M. Matos, A.F.D. Barbosa-Povoa, J. Fialho, A.S. Pinheiro, Pipeline scheduling and inventory management of a multiproduct distribution oil system, Ind Eng Chem Res 45 (2006) 7841–7855.
[19] S. Relvas, H.M. Matos, A.F.D. Barbosa-Povoa, J. Fialho, Reactive scheduling framework for a multiproduct pipeline with inventory management, Ind Eng Chem Res 46 (2007) 5659–5672.
[20] S. Relvas, S.N. Boschetto Magatao, A.F.D. Barbosa-Povoa, J. Fialho, Integrated scheduling and inventory management of an oil products distribution system, Omega: Int J Manag Sci 41 (2014) 955–968.
[21] R. Rejowski, J.M. Pinto, Scheduling of a multiproduct pipeline system, Comput Chem Eng 27 (2003) 1229–1246.
[22] D.C. Cafaro, J. Cerdá, Dynamic scheduling of multiproduct pipelines with multiple delivery due dates, Comput Chem Eng 32 (2008) 728–753.
[23] R. Rejowski, J.M. Pinto, A novel continuous time representation for the scheduling of pipeline systems with pumping yield rate constraints, Comput Chem Eng 32 (2008) 1042–1066.
[24] L. Yongtu, L. Ming, Z. Ni, A study on optimizing delivering scheduling for a multiproduct pipeline, Comput Chem Eng 44 (2012) 127–140.
[25] H. Mostafaei, A. Ghaffari-Hadigheh, A general modeling framework for the

long-term scheduling of multiproduct pipelines with delivery constraints, Ind Eng Chem Res 53 (2014) 7029–7042.
[26] D.C. Cafaro, J. Cerdá, Optimal scheduling of refined products pipelines with multiple sources, Ind Eng Chem Res 48 (2009) 6675–6689.
[27] D.C. Cafaro, J. Cerdá, Operational scheduling of refined products pipeline networks with simultaneous batch injections, Comput Chem Eng 34 (2010) 1687–1704.
[28] S.A. MirHassani, M. Abbasi, S. Moradi, Operational scheduling of refined product pipeline with dual purpose depots, Appl Math Model 37 (2013) 5723–5742.
[29] H. Mostafaei, Y. Alipouri, J. Shokri, A mixed-integer linear programming for scheduling a multi-product pipeline with dual-purpose terminals, Comput Appl Math 34 (2015) 979–1007.
[30] D.C. Cafaro, J. Cerdá, A rigorous mathematical formulation for the scheduling of tree-structure pipeline networks, Ind Eng Chem Res 50 (2011) 5064–5085.
[31] S.A. Mirhassani, H. Fani Jahromi, Scheduling multi-product tree-structure pipelines, Comput Chem Eng 35 (2011) 165–176.
[32] P.M. Castro, Optimal scheduling of pipeline systems with a resource-task network continuous-time formulation, Ind Eng Chem Res 49 (2010) 11491–11505.
[33] A. Herrán, J.M. de la Cruz, B. de Andrés, Mathematical model for planning transportation of multiple petroleum products in a multi-pipeline system, Comput Chem Eng 34 (2010) 401–413.
[34] A. Herrán, J.M. de la Cruz, B. de Andrés, Global search metaheuristics for planning transportation of multiple petroleum products in a multi-pipeline system, Comput Chem Eng 37 (2012) 248–261.
[35] T.M.T. Lopes, A.A. Cire, C.C. de Souza, A.V. Moura, A hybrid model for a multiproduct pipeline planning and scheduling problem, Constraints 15 (2010) 151–189.
[36] D.C. Cafaro, J. Cerdá, Rigorous scheduling of mesh-structure refined petroleum pipeline networks, Comput Chem Eng 38 (2012) 185–203.
[37] R. Techo, D.L. Holbrook, Computers scheduling the world's biggest product pipeline, Pipeline Gas J 4 (1974) 27.
[38] V.G. Cafaro, D.C. Cafaro, C.A. Mendéz, J. Cerdá, Detailed scheduling of operations in single-source refined products pipelines, Ind Eng Chem Res 50 (2011) 6240–6259.
[39] V.G. Cafaro, D.C. Cafaro, C.A. Mendéz, J. Cerdá, Detail scheduling of single-source pipelines with simultaneous deliveries to multiple offtake stations, Ind Eng Chem Res 51 (2012) 6145–6165.
[40] V.G. Cafaro, D.C. Cafaro, C.A. Mendéz, J. Cerdá, Optimization model for the detailed scheduling of multi-source pipelines, Comput Ind Eng 88 (2015) 395–409.
[41] A. Ghaffari-Hadigheh, H. Mostafaei, On the scheduling of real world multi-product pipelines with simultaneous delivery, Optim Eng 16 (2015) 571–604.
[42] H. Mostafaei, P.M. Castro, A. Ghaffari-Hadigheh, A novel monolithic MILP framework for lot-sizing and scheduling of multiproduct treelike pipeline networks, Ind Eng Chem Res 54 (2015) 9202–9221.
[43] H. Mostafaei, Y. Alipouri, M. Zadahmad, A mathematical model for scheduling of real-world tree-structured multi-product pipeline system, Math Methods Oper Res 81 (2014) 53–81.
[44] R. Raman, I.E. Grossmann, Modeling and computational techniques for logic based integer programming, Comput Chem Eng 18 (1994) 563–578.
[45] E. Balas, Disjunctive programming, Ann Discret Math 5 (1979) 3–51.
[46] I.E. Grossmann, F. Trespalacios, Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming, AIChE J 59 (2013) 3276–3295.
[47] P.M. Castro, I.E. Grossmann, Global optimal scheduling of crude oil blending operations with RTN continuous-time and multiparametric disaggregation, Ind Eng Chem Res 53 (2014) 15127–15145.
[48] P.M. Castro, D. Rodrigues, H.A. Matos, Cyclic scheduling of pulp digesters with integrated heating tasks, Ind Eng Chem Res 53 (2014) 17098–17111.
[49] P.M. Castro, I.E. Grossmann, Generalized disjunctive programming as a systematic modeling framework to derive scheduling formulations, Ind Eng Chem Res 51 (2012) 5781–5792.
[50] P.M. Castro, I.E. Grossmann, P. Veldhuizen, D. Esplin, Optimal maintenance scheduling of a gas engine power plant using generalized disjunctive programming, AIChE J 60 (2014) 2083–2097.
[51] P.M. Castro, I. Marques, Operating room scheduling with generalized disjunctive programming, Comput Oper Res 64 (2015) 262–273.