European Journal of Operational Research 000 (2018) 1–23

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Innovative Applications of O.R.

A matheuristic decomposition approach for the scheduling of a single-source and multiple destinations pipeline system

William Hitoshi Tsunoda Meira[a], Leandro Magatão[a,*], Susana Relvas[b], Ana Paula Barbosa-Póvoa[b], Flávio Neves Jr.[a], Lúcia V. R. Arruda[a]

[a] CPGEI-UTFPR - Graduate Program in Electrical and Computer Engineering, Universidade Tecnológica Federal do Paraná, Curitiba, Paraná 80230-901, Brazil
[b] CEG-IST - Centre for Management Studies, Instituto Superior Técnico, Universidade de Lisboa, Lisbon 1049-001, Portugal

A R T I C L E I N F O

Article history:
Received 3 August 2017
Accepted 12 January 2018
Available online xxx

Keywords:
Distribution
Multiproduct pipeline
Scheduling
Decomposition approach
Mixed integer linear programming

A B S T R A C T

An improvement on the scheduling of pumping and delivery operations in an installed pipeline network can lead to considerable profits to the using companies, such as oil companies. This paper proposes a decomposition approach that integrates heuristic procedures and mixed integer linear programming (MILP) models, a matheuristic, to solve the long-term scheduling of a pipeline system, which connects a single-source to multiple distribution centers. The approach provides a rigorous inventory management and flow rate control taking into account several operational aspects, such as simultaneous deliveries, and prespecified periods of tank maintenance and pipeline maintenance. To validate the developed approach, two case studies were devised. In case study 1, several instances of an illustrative network were solved and case study 2 addressed three examples of a real-world network: base instance; extended instance with maintenance periods; and model performance tests. Valid solutions that can be operationally implemented were obtained for all executions in a reasonable computational time. Detailed discussions of the obtained solutions are presented and indicate an inventory control in accordance with operational requirements.

1. Introduction

One of the major challenges in the oil industry is the management of the transportation activities of petroleum products from the refineries to the distribution centers. The transport is usually made through diverse means, such as: trucks, trains, vessels, and pipelines. In many countries, most part of their oil derivatives production (e.g. gasoline, diesel, kerosene, naphtha) are distributed through pipelines, justified because of the high volume capacity, reliability, economy, and safety compared to other means (Sasikumar, Ravi Prakash, Patil, & Ramani, 1997). Currently, most of the planning and scheduling activities in pipeline systems are decided by a group of specialists (schedulers), where their decisions are based on past experiences and manual calculations. In order to aid the decision-making process, optimization techniques have received great interest from the oil companies, where any improvement of the process, or better usage of the available resources, may increase considerably their profits (Boschetto et al., 2010).

A typical pipeline network topology consists of a single-source (a refinery) connecting one or more distribution centers (DCs) with a straight multiproduct pipeline. More complex topologies are possible, including systems with multiple sources (Cafaro & Cerdá, 2009), DCs that can act also as inputs (dual purpose) (Cafaro, Cafaro, Méndez, & Cerdá, 2015b; Mostafaei & Castro, 2017; Mostafaei, Castro, & Ghaffari-Hadigheh, 2016), multiple pipelines that configure tree-structures (Cafaro & Cerdá, 2011; MirHassani & Jahromi, 2011) or mesh-structures (Magatão, Magatão, Neves-Jr, & Arruda, 2015; Magatão et al., 2012; Polli, Magatão, Magatão, Neves-Jr, & Arruda, 2017). In this paper, we focus on the problem with one source and multiple destinations. At this topology, a common transport operation consists of a refinery pumping oil product batches, without any separator device, into the straight pipeline in order to attend the local demand of the connected DCs at the right time and the lowest possible cost. The interface of two consecutive batches of different products generates a contaminated volume. If this volume is substantially high, then the sequence is considered a forbidden operation. The scheduling of a multiproduct pipeline is a hard problem, whose complexity is NP-complete (Jittamai, 2004). It involves many operational restrictions, such as inventory control at DCs, demand attendance, flow rate control, product restrictions, simultaneous deliveries.

* Corresponding author.
E-mail address: magatao@utfpr.edu.br (L. Magatão).

https://doi.org/10.1016/j.ejor.2018.01.032

Nomenclature

Indices/sets

$e \in E$	set of events
$i, i' \in I$	set of intervals. The interval i starts at $e = i - 1$ and ends at $e = i$
$p, p' \in P$	set of products (oil derivatives)
$n \in N$	set of distribution centers (DCs)
$\{n, p\} \in NP$	sparse set containing the tuple $\{n, p\}$; node n being a DC where product p can be delivered. Not necessarily all products can be delivered to all DCs
$\{n, p, e\} \in NPE$	sparse set containing the tuple $\{n, p, e\}$, which associates every element of $\{n, p\} \in NP$ to every $e \in E$
$\{n, p, i\} \in NPI$	sparse set containing the tuple $\{n, p, i\}$, which associates every element of $\{n, p\} \in NP$ to every $i \in I$
$b, b' \in B$	set of initialization and allocated batches indexes. The first index is the closest batch to the last DC, increasing a unit for each following initialization batch and then the allocated batches as each one enters the network
$ib \in B^{init}$	set of only initialization batches indexes. This set is a subset of all batches ($B^{init} \subset B$)
$\{b, e\} \in BE$	set containing the tuple $\{b, e\}$, which combines each batch index b with an event e
$\{b, i\} \in BI$	set containing the tuple $\{b, i\}$, which associates each batch index b with an interval i
$\{b, e, n\} \in BEN$	set containing the tuple $\{b, e, n\}$, which associates each batch index b with an event e and a DC n
$\{b, i, n\} \in BIN$	set containing the tuple $\{b, i, n\}$, which associates each batch index b with an interval i and a DC n

Parameters

$U \gg 0$	upper bound value (e.g. $U = 10^6$)
$L \ll 0$	lower bound value (e.g. $L = -U$)
ϵ	small constant value used to avoid equalities (e.g. $\epsilon = 10^{-4}$)
T_i	duration of the interval i (hour)
IP_p	binary parameter indicating if product p is being pumped from the refinery at the initial event ($e = 0$)
IV_p	initial volume of product p being pumped from the refinery at the initial event ($e = 0$). It is greater than 0 only for the product p where $IP_p = 1$ (vu)
VC_n	volumetric coordinate of DC n from zero coordinate, $n = 0$ (vu)
$Fp_{p,i}^{min}$	minimum pumping flow rate of product p during the interval i (vu/hour)
$Fp_{p,i}^{max}$	maximum pumping flow rate of product p during the interval i (vu/hour)
$Fd_{n,p}^{min}$	minimum delivery flow rate of product p at the DC n (vu/hour)
$Fd_{n,p}^{max}$	maximum delivery flow rate of product p at the DC n (vu/hour)
Fs_p^{min}	minimum segment passage flow rate of product p for the considered pipeline segment during the actual SM iteration (vu/hour)
Fs_p^{max}	maximum segment passage flow rate of product p for the considered pipeline segment during the actual SM iteration (vu/hour)
Lb_p^{min}	minimum batch pumping size (vu)
Lb_p^{max}	maximum batch pumping size (vu)
Ld_p^{min}	minimum delivery size (vu)
$PP_{p,p'}$	matrix of forbidden pumping sequences between two consecutive products p and p'
$Dem_{n,p,i}$	demand of product p in DC n during the interval i (vu)
$Rec_{n,p,i}$	volume of product p being received in DC n during the interval i (vu). The values are obtained during the initialization batches simplification in the ASM and also in the SM
B_e^{last}	indicates the index of the closest batch b to the upstream DC at the event e, which also means the last batch pumped into the segment
B_i^{pass}	indicates the index of batch b passing along the upstream DC during interval i
V_i^{pass}	volume of the batch part B_i^{pass} passing along the upstream DC during interval i (vu)
F_i^{pass}	flow rate of the batch part B_i^{pass} passing along the upstream DC during interval i (vu/hour)
P_b	product p of the batch b
V_{ib}^{init}	volume of the initialization batch ib at the beginning of the horizon (vu)
$HPos_{ib}^{init}$	volumetric coordinate of the initialization batch head ib at the beginning of the horizon (vu)
$ID_{n,p}$	initial inventory volume of product p in DC n (vu)
$ID_{n,p,e}^{empty}$	volume considered empty for the aggregate inventory storage of product p in DC n at the event e (vu)
$ID_{n,p,e}^{cap}$	volume capacity of the aggregate inventory storage of product p in DC n at the event e (vu)
$ID_{n,p,e}^{min}$	minimum operational level for the aggregate inventory storage of product p in DC n at the event e (vu)
$ID_{n,p,e}^{max}$	maximum operational level for the aggregate inventory storage of product p in DC n at the event e (vu)
$ID_{n,p,e}^{mintg}$	minimum target level for the aggregate inventory storage of product p in DC n at the event e (vu)
$ID_{n,p,e}^{maxtg}$	maximum target level for the aggregate inventory storage of product p in DC n at the event e (vu)
Cid^{emp}	penalty cost per vu of inventory empty violation at an event
Cid^{cap}	penalty cost per vu of inventory capacity violation at an event
Cid^{min}	penalty cost per vu of inventory minimum operational level violation at an event
Cid^{max}	penalty cost per vu of inventory maximum operational level violation at an event
Cid^{mintg}	penalty cost per vu of inventory minimum target level violation at an event
Cid^{maxtg}	penalty cost per vu of inventory maximum target level violation at an event
Cfr^{lmean}	penalty cost per vu/hour lower violation of the mean flow rate at an event
Cfr^{umean}	penalty cost per vu/hour upper violation of the mean flow rate at an event
Cfr^{min}	penalty cost per vu/hour of flow rate minimum flow violation at an event
Cfr^{max}	penalty cost per vu/hour of flow rate maximum flow violation at an event
Cfr^{ldiff}	penalty cost per vu/hour due the lower difference of flow rate compared to the last interval

Cfr^{udiff} penalty cost per vu/hour due the upper difference of flow rate compared to the last interval
$Cint$ penalty cost associated with the contamination volume due the interface between two different products being pumped sequentially

Continuous variables
$v_{p,i}$ volume of product p allocated to be pumped during the interval i (vu)
$v^{dc}_{n,p,i}$ volume of product p to be delivered in the DC n during the interval i (vu)
volume of product p expected to be delivered to DC n during interval i from the volume
$v^{batch}_{p,i}$ batch size of product p being pumped during the interval i (vu).
$vm_{b,e}$ volume of the batch b into the considered pipeline segment at the event e (vu)
$vt^{d}_{b,i,n}$ volume of batch b delivered to DC n during interval i (vu)
$cd^{head}_{b,e}$ volumetric coordinate of the batch's head b at the event e (vu)
$cd^{tail}_{b,e}$ volumetric coordinate of the batch's tail b at the event e (vu)
$fr_{p,\,i}$ pumping flow rate allocated to the product p during the interval i (vu/hour)
$fr^{d}_{b,i,n}$ flow rate of the delivery operation of batch b to DC n during interval i (vu/hour)
$fr^{s}_{b,i}$ flow rate of the batch b that enters into the pipeline segment being considered in SM, which discounts the flow rate of the delivery operation (vu/h)
mfr mean pumping flow rate of the refinery (vu/hour)
$lmfr_i$ lower difference between the flow rate during interval i and the mfr (vu/hour)
$umfr_i$ upper difference between the flow rate during interval i and the mfr (vu/hour)
$inv_{n,p,e}$ inventory level of product p in the DC n at the event e (vu)
$vid^{emp}_{n,p,e}$ volume violated with respect to empty storage of product p in DC n at the event e (vu)
$vid^{cap}_{n,p,e}$ volume violated with respect to capacity level of product p in DC n at the event e (vu)
$vid^{min}_{n,p,e}$ volume violated with respect to minimum operational level of product p in DC n at the event e (vu)
$vid^{max}_{n,p,e}$ volume violated with respect to maximum operational level of product p in DC n at the event e (vu)
$vid^{mintg}_{n,p,e}$ volume violated with respect to minimum target level of product p in DC n at the event e (vu)
$vid^{maxtg}_{n,p,e}$ volume violated with respect to maximum target level of product p in DC n at the event e (vu)
$vfr^{min}_{b,i}$ minimum flow rate violation of the batch b during interval i for the pipeline segment being considered in the SM iteration
$vfr^{max}_{b,i}$ maximum flow rate violation of the batch b during interval i for the pipeline segment being considered in the SM iteration
vfr^{ldiff}_{i} lower difference between the flow rate that passes the upstream DC during interval i and $i-1$ (vu/hour)
vfr^{udiff}_{i} upper difference between the flow rate that passes the upstream DC during interval i and $i-1$ (vu/hour)

Binary variables
$a_{p,\,i}$ 1 if the product p is allocated to be pumped in the interval i; 0 otherwise
$ch_{p,\,i}$ 1 if the product p allocated in the interval $i-1$ ($a_{p,i-1}=1$) changes to other product in the interval i ($a_{p,\,i}=0$); 0 otherwise
$pd_{b,\,i,\,n}$ 1 if there is the possibility to deliver batch b to the DC n during interval i; 0 otherwise
$ss_{b,\,i,\,n}$ 1 if a delivery operation is scheduled for the batch b to the DC n during interval i; 0 otherwise
$ss^{dc}_{b,n}$ 1 if at least one delivery operation is scheduled for batch b to the DC n; 0 otherwise
$hb_{b,\,e,\,n}$ 1 if the head of batch b already reached the DC n at the event e; 0 otherwise
$tb_{b,\,e,\,n}$ 1 if the tail of batch b already passed the DC n at the event e; 0 otherwise

Several approaches have been studied to solve pipeline scheduling problems, as hereafter detailed in this section. In general terms, the approaches are based on heuristics (Kirschstein, 2018; Sasikumar et al., 1997), mathematical programming using mixed integer linear programming (MILP) or mixed integer non-linear programming (MINLP) formulations (Cafaro, Cafaro, Méndez, & Cerdá, 2015a; Zaghian & Mostafaei, 2016), and decomposition strategies (Cafaro, Cafaro, Méndez, & Cerdá, 2011; 2012). Solution approaches that tackle real-world perspectives of the problem, also considering a long-term scheduling plan, tend to apply structural and/or temporal decomposition techniques to achieve solutions in a reasonable computational time, given that pipeline scheduling is an operational planning problem.

Hane and Ratliff (1995) presented one of the first MILP formulations for the scheduling of a multiproduct pipeline connecting a refinery to multiple destinations. The problem was decomposed into subproblems that were solved by a branch-and-bound algorithm. Rejowski and Pinto (2003) presented a discrete MILP model to solve a real-world pipeline network, transporting four products from a refinery to five destinations. The volume between two consecutive DCs was denominated as “segment” and each segment divided into volume packs. Two models were formulated, one with equal pack size and other with different sizes. The time representation was also discrete and the objective was to minimize the storage, pumping, and interface costs. In order to improve the developed model performance, Rejowski and Pinto (2004) extended the previous model adding special constraints and valid cuts. Later on, Rejowski and Pinto (2008) proposed, for the same problem, an MINLP model considering an adjustable time interval duration, however still dividing the pipeline into volume batches. The novel approach allowed changes in the pumping rate and also treating some hydraulics problems involved.

Cafaro and Cerdá (2004) developed an MILP continuous formulation for the scheduling of a multiproduct pipeline from one refinery to multiple DCs, similar to the problem presented by Rejowski and Pinto (2003). The objective function was to minimize the interfaces, pumping and inventory costs attending the demand of each depot on time. Cafaro and Cerdá (2008) generalized the previous work for the long-term scheduling, considering a multi-period rolling horizon with a demand delivery due date at the end of each weekly period, instead of just at the horizon end. Cafaro et al. (2011) introduced a two-level hierarchical decomposition, (a) the aggregate planning and (b) the detailed scheduling. The upper level (a) determines the sequence of batches and the aggregate deliveries to depots during each pumping run. The lower level (b) is a temporization step that determines when each valve and pump should be turned on/off. Their approach involved

MILP and discrete-event simulation methods. Cafaro, Cafaro, Méndez, and Cerdá (2012) presented a continuous-time MILP model for the scheduling of a single-source and multiple destinations pipeline system that allows simultaneous deliveries at two or more distribution centers. Consequently, the number of flow restarts and stoppages reduced considerably. Following, Cafaro et al. (2015a) developed an MINLP continuous formulation, considering friction loss and tracking power consumption with nonlinear equations. Thus, the benefit achieved was a more stable flow rate for every pipeline segment over the planning horizon.

Mirhassani, Moradi, and Taghinezhad (2011) proposed a continuous MILP model for the scheduling of the same problem. They considered interesting aspects such as daily demand due dates, settling periods for quality control, and prespecified shutdown periods for pipeline maintenance. For long-term scheduling, the time horizon was divided into periods, where each one was determined by a heuristic that calculated the first day when a DC would face a shortage.

Based on Cafaro et al. (2012), Mostafaei and Ghaffari-Hadigheh (2014) developed an MILP formulation for the same problem. The model was capable of handling simultaneous deliveries and also permitted lots of different products to be delivered to a DC during the same pumping run. For the short-term scheduling of the same problem, Ghaffari-Hadigheh and Mostafaei (2015) proposed a single-step MILP model with the objective of finding the optimal sequence of input and output operations at minimum total costs. Recently, Zaghian and Mostafaei (2016) also developed a single-step MILP model for short-term scheduling of a multi-product pipeline with a single-source and several DCs. Their approach included the possibility to deliver multiple products to an active DC during the same pumping run, similar to Mostafaei and Ghaffari-Hadigheh (2014) consideration.

The combinatorial characteristic of the problem poses a significant computational challenge as the models approximate to the operational reality including more and more aspects of a real-world pipeline network. In order to overcome this problem some approaches, like Cafaro et al. (2012), experimented a structural and/or temporal (rolling horizon) decomposition. In this work, we addressed a series of operational aspects in an unified computational framework that was able to solve a real pipeline network case with a refinery distributing products to multiple DCs. The following operational aspects, which are commonly found in real-world contexts, were addressed: pipeline maintenance, which can decrease partially or totally the efficiency of the network; tank maintenance; a more rigorous inventory control, taking into account different inventory levels, namely: (i) physical empty/capacity levels; (ii) minimum/maximum operational levels; and, (iii) minimum/maximum target levels as detailed in Boschetto et al. (2010), Magatão et al. (2012) and Polli et al. (2017). Thus, the objective function considers shortage/surplus penalties to each one of the three mentioned levels. We also treat the demand consumption as a parameter, which can daily vary, and has to be strictly respected along the time horizon. Therefore, the proposed system needs to adapt the pipeline scheduling to avoid inventory violations in the DCs. This imposes a significant computational challenge, since the approach cannot determine how the clients are attended, but rather respect a previously established DC demand consumption rate. We propose a structural and temporal decomposition strategy combining heuristic procedures and MILP models, consisting in a matheuristic decomposition approach.

The remainder of this paper is organized as follows. Section 2 describes the problem studied and the problem assumptions. Section 3 describes the decomposition approach developed. Sections 4 and5 detail the two modules of the proposed approach, Allocation and Sequencing Module and the Scheduling Module, respectively. Section 6 discusses the results for two case studies; flexibility and scalability aspects are evidenced. Finally, Section 7 presents a conclusion and future work.

2. Problem definition

The problem consists of a real-world straight multiproduct pipeline network that connects a single-source (refinery) to multiple destinations. The refinery pumps batches of products, oil derivatives, into the pipeline and the DCs act as removal terminals. The product is extracted from a passing batch, stored into a tank and, then, consumed by the local market demand. The proposed approach schedules the refinery pumping and receiving operations for a long-term operational planning period (e.g. 30 days) respecting the operational constraints of the system while minimizing shortage and surplus of inventory for each product in each DC; pumping and segment flow rate variations; and number of product interfaces.

The developed decomposition approach relies on the following assumptions:

- The batches flow unidirectionally from the refinery to the downstream distribution centers.
- The pipeline is always full of incompressible products and the only way to remove a volume of product to one or more DCs is by injecting the same volume into the pipeline. When assuming liquid incompressibility, an input operation propagates instantaneously through the downstream network and the same amount of volume is removed from the system.
- The refinery works in an ideal way, that is, the production and storage capacities are sufficient to supply any pumping need into the pipeline.
- Each DC considers all tanks that store the same product as an aggregate tank. During a tank maintenance period, the volume of the unavailable tank is subtracted from the respective aggregate tank capacity of the corresponding product.
- The interface between two consecutive batches of different products generates a contaminated volume. We consider a fixed penalty cost aggregated to the contamination event and the changeover operation. However, when the contamination between two products may exceed a certain limit in product specification, the sequence is considered a forbidden operation.
- The volume of each product batch to be pumped is limited by known lower and upper bounds.
- The volume to be removed from a batch in a delivery (receiving) operation is limited by a known lower bound.
- The admissible minimum and maximum product pumping rate, pipeline segment flow rate, and DC receiving flow rate are known.
- Tank maintenance due to technical problems may also exist, reducing the aggregate tankage capacity of a product in a DC. The period of each tank maintenance is specified in advance and the details are inputs of the network.
- Pipeline maintenance may occur, reducing partially or totally the pumping rate efficiency of the network. The period of each pipeline maintenance is specified within days in advance and the details are inputs of the system.
- Simultaneous deliveries of products in two or more terminals are allowed.
- Delivery operations of multiple batches into a DC during the same pumping run is allowed.
- Initialization batches, initial inventory level, inventory levels (physical empty/capacity, operational minimum/maximum and target minimum/maximum), demand consumption and horizon length are known parameters of the system. Initialization batches are the ones that are located inside of the pipeline at the beginning of the horizon. Operational limits represent the

Please cite this article as: W.H.T. Meira et al., A matheuristic decomposition approach for the scheduling of a single-source and multiple destinations pipeline system, European Journal of Operational Research (2018), https://doi.org/10.1016/j.ejor.2018.01.032

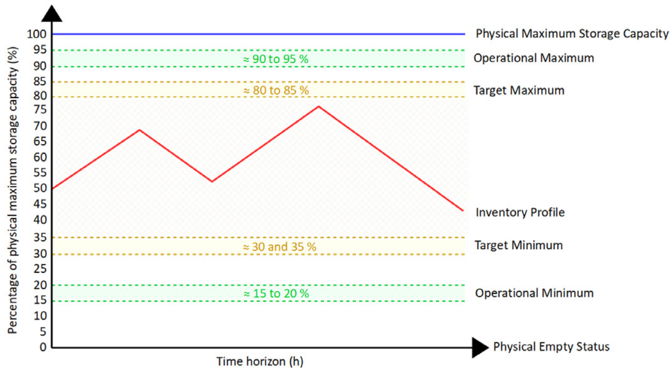


Fig. 1. Typical inventory levels as a percentage of the physical maximum storage capacity.

maximum and minimum recommended inventory levels, which maintain a certain distance to the physical maximum storage capacity or physical empty status. The range between the target maximum and minimum levels frames the adequate inventory profile to be maintained in the distribution center. Each inventory level is an input parameter of the system, which is provided by the oil company, and its value is usually defined as a percentage of the physical maximum storage capacity, which consists of the total aggregate storage capacity of all tanks allocated to a certain product in a distribution center. Fig. 1 illustrates typical inventory levels.

The output of the system is the complete detailed scheduling of batch pumping operations and delivery operations during the specified horizon.

3. Decomposition approach

The proposed solution for the defined problem is based on a decomposition strategy. A macro view of the developed approach is exhibited in Fig. 2.

The instance contains all the input parameters for the proposed approach. The parameters are the network information (volume of each pipeline segment, position of the DCs, tanks volume and associated product); the daily demand per product in each DC; initialization batches; inventory limits of each product in each DC; initial inventory at the beginning of the horizon; prespecified pipeline and tank maintenance periods; and configuration parameters of the approach (e.g. uniform interval size, time horizon length in days).

The Allocation and Sequencing Module (ASM) is responsible for determining the product, volume, flow rate and sequence of the batches to be pumped from the refinery during the considered horizon. The Scheduling Module (SM) solves the detailed scheduling of all the delivery (removal) operations to be executed at each DC during the considered horizon.

Detailed explanation of the processes in ASM and SM are available in Sections 4 and 5, respectively. After running the two modules, the output is the complete pipeline scheduling, which includes the pumping operations from the refinery and all the delivery operations during the complete horizon.

4. Allocation and Sequencing Module (ASM)

The first module to be executed is the Allocation and Sequencing Module (ASM). Fig. 3 exhibits the execution flowchart of this module. This execution considers, apart from the module decomposition, a temporal decomposition, which uses a rolling horizon concept. The module receives the instance as input, preprocesses the received information (Section 4.1), then initiates the MILP rolling horizon processing (Section 4.2), which involves solving a period of the time horizon at each iteration. During an iteration, the input data preprocessing, the discrete-time MILP model (detailed in Section 4.3), and the output data postprocessing are executed. At the end of the horizon, the iterative part ends, and the final obtained result (the scheduling of pumping operations) serves as an input to the next module.

4.1. Allocation and sequencing preprocessing

The Allocation and Sequencing preprocessing receives the instance parameters. Afterwards, a series of heuristics and algorithms address the structure of the network in a different perspective from reality that simplifies some operational aspects. These operational aspects will be better detailed in the SM module presented in Section 5. The heuristics and algorithms are hereafter described.

A refinery perspective of the DCs is assumed, considering an average time spent since the product is pumped from the refinery until the start of the delivery at each DC. In order to obtain the approximated average travel time of each batch, we developed a heuristic that estimates the average flow rate of each segment based on the characteristics of the instance, mainly the demands of the products. Since it is known the average flow rate and volume of each segment, it is straightforward to obtain the average time necessary to an allocated volume to arrive at a certain DC. Algorithm 1 presents the average segments flow rate heuristic (vu stands for volumetric units). Notices that the algorithm also considers the efficiency reduction during each pipeline maintenance period, which affects the mean flow rate of the pipeline segments.

Fig. 4 illustrates the ASM perspective of the network after the pipeline simplification. In this figure, T1, T2,..., TN are the average times estimated to a batch to pass, respectively, the pipeline segment S1, S2,..., SN. The refinery allocates products directly to DC1 with a T1 hours of delay, to DC2 with (T1 + T2) hours of delay, until DCN with a (T1 + T2 + $\cdots$ + TN) hours of delay.

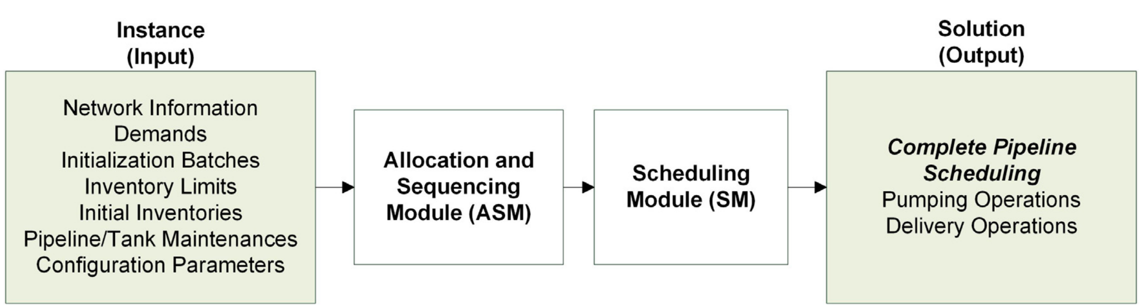


Fig. 2. Macro flowchart of the proposed approach.

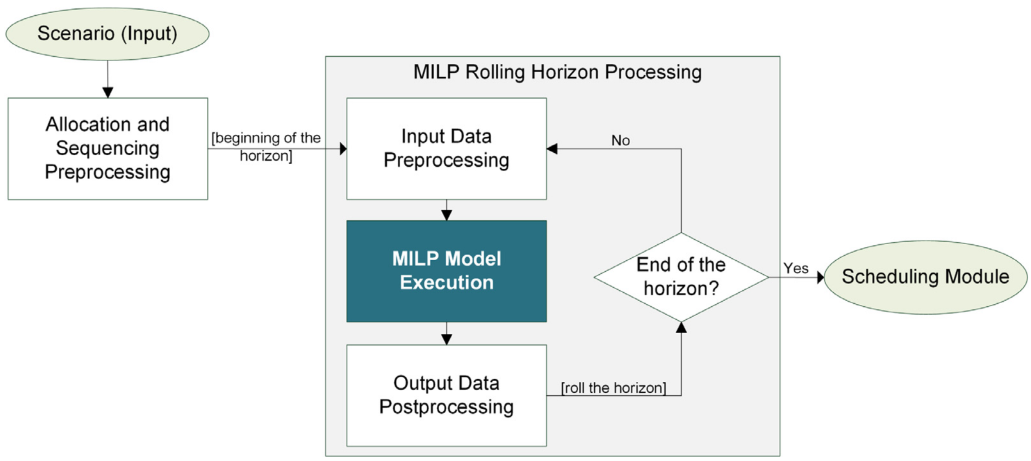


Fig. 3. Flowchart of the Allocation and Sequencing Module.

Algorithm 1 Average segments flow rate heuristic.

function AVERAGESEGMENTFLOWRATE
  totalDemand ← sum of all volume demanded by all products in all DCs (*vu*)
  timeHorizon ← size of the considered horizon (*hour*)
  duration(m) ← duration of pipeline maintenance $m \in M$ (hour)
  $eff(m)$ ← efficiency reduction factor (0 to 1) of pipeline maintenance m
  $time \leftarrow timeHorizon - \sum_{m \in M}(duration(m) \times eff(m))$
  volume ← 0
  ps ← first pipeline segment
  while $ps \neq null$ **do**
    $avgFlowRate(ps) \leftarrow (totalDemand - volume)/time$
    volume ← *volume* + demand of the DC receiving products from segment *ps*
    ps ← next pipeline segment
  end while
  return *avgFlowRate*
end function

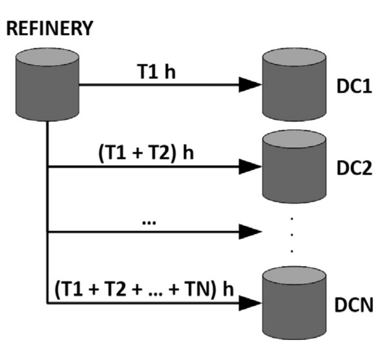


Fig. 4. ASM perspective of the network. TX is the average time estimated for a batch to pass the pipeline segment SX based on its average segment flow rate.

Since the refinery observes each DC delayed in time, the demand curves, of each pair DC-product, must be also shifted forward the same amount of time in order to correctly view the curve at the current time and prevent inventory shortages or surpluses. Fig. 5 shows an example of the demand forward shifting algorithm for the DC2. In this example, let us assume that the average flow rate obtained by the Algorithm 1 was 800 vu/hour and 600 vu/hour for the first (refinery to DC1) and second (DC1 to DC2) segments, respectively. If the volume of the first segment is 4,800 vu and the second segment is 2,400 vu, then the average time to a batch arrive at DC2 is 10 hours: 6 hours to DC1 (T1) plus 4 hours to DC2 (T2). Thus, the demand curve of DC2 needs to be forward shifted 10 hours, which means bringing all the demand of DEM1 period to the beginning of the horizon and DEM2 to be considered by the refinery 10 hours earlier. Most of the accumulated initial demand is fulfilled by the initialization batches existing in the pipeline system at the beginning of the time horizon.

To consider the initialization batches, we formulate a heuristic to simulate the propagation of all these batches until the last DC, defining estimated delivery operations. The heuristic uses the same average segment flow rate obtained before in Algorithm 1 to propagate each initialization batch, estimating when each delivery operation will occur and the amount of volume that is estimated to be delivered to each downstream DC. The batch volume to be delivered at a passing DC depends on the percentage of the total product (same as the batch) demanded by the remaining downstream DCs. For example, consider a network with 3 DCs, a batch of gasoline with 10,000 vu and the total demand of gasoline for the 3 DCs being, respectively, 25,000 (25 %), 40,000 (40 %) and 35,000 (35 %) vu, as shown in Fig. 6. In the first case, Fig. 6(a), the batch starts at the first segment and the heuristic will deliver, proportionally to the downstream remaining demand, 2,500 vu for the DC1, 4,000 vu for the DC2 and 3,500 vu for the DC3. If this batch is initially in the second segment, as indicated in Fig. 6(b), then approximately 5,333 vu (53.33 %, ratio of 40,000 to 75,000) would be delivered for the DC2 and 4,667 vu for the DC3. For the timing of these deliveries, we use the propagation algorithm, which will be better explained in Section 5.3. This algorithm simulates the refinery pumping batches into the first pipeline segment with its average flow rate in order to move all the initialization batches until the last DC. The start and end time of a certain initialization batch passing through a DC are, then, set as the start and end time of the estimated delivery of this batch to that DC. After the execution of the heuristic, a delivery curve for each pair DC-product is obtained. All the delivery curves are then shifted forward, in the same way of the demand curves, to obtain the refinery instantaneously view of the network. After the preprocessing of the network, the module is ready to determine the pumping operations to be executed during the time horizon.

4.2. MILP rolling horizon processing

The iterative part of this module consists of a rolling horizon process. At each iteration, the considered total period is constituted by a current period, which we are solving at the moment, and an

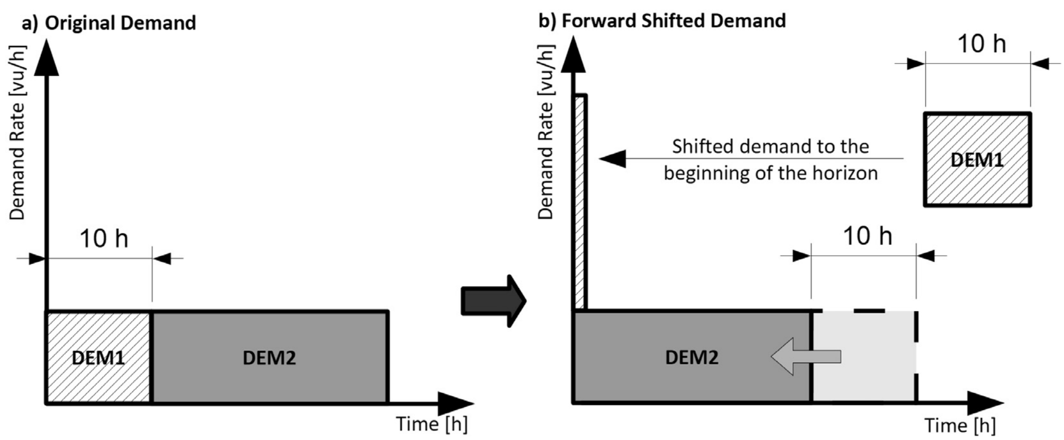


Fig. 5. Example of forward demand shifting to DC2: (a) original demand, (b) 10 hours forward shifted demand.

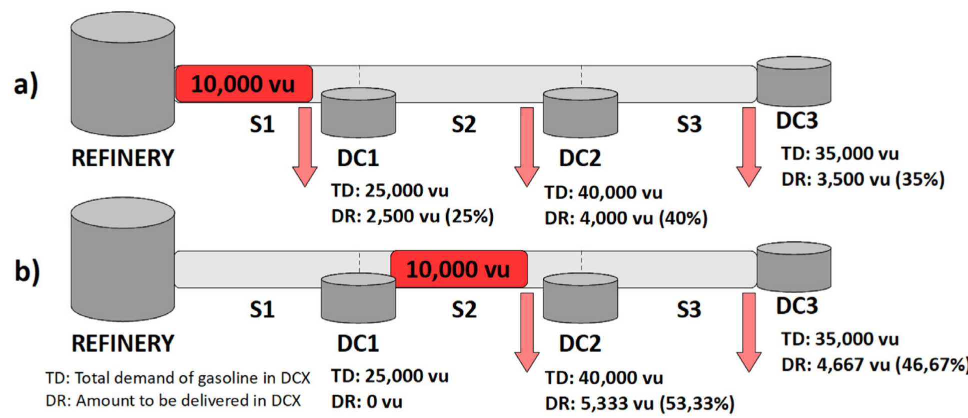


Fig. 6. Example of a 10,000 vu batch of gasoline delivered by the adopted heuristic for initialization batch when it is initially in the first segment (a) and in the second segment (b).

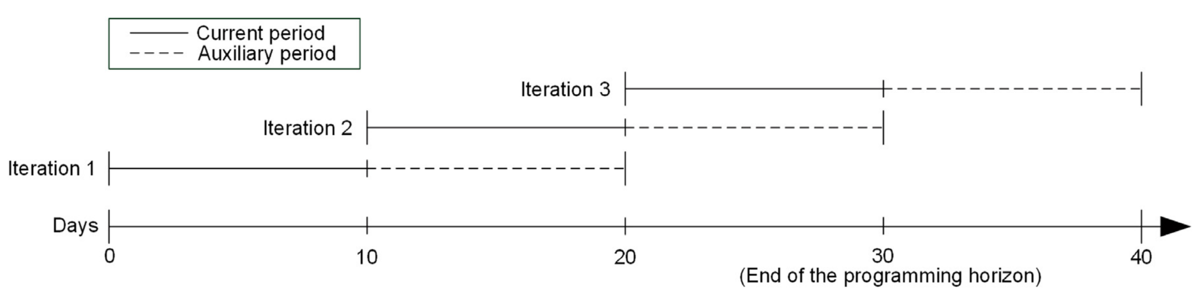


Fig. 7. Example of a rolling horizon execution.

auxiliary period, whose function is to guide the model, helping to anticipate future problems caused, for example, by maintenance periods and demand variations. Fig. 7 illustrates the period division for each iteration, given the following configuration: 30 days of time horizon, 10 days of current period and 10 days of auxiliary period.

Here, before each execution of the ASM MILP model, the input data is organized based on the period being considered. Also, some parameters, such as the initial inventory and initialization batches, which are the final values at the end of the previous iteration must be updated as the initial status of the next iteration. Future demand data of a product in a DC, after the end of the programming horizon, is determined by its average demand flow rate during the programmed horizon. Then, the model (main process, whose mathematical formulation will be detailed in Section 4.3) finds a solution for the current period. The output data processing eliminates batches sequenced during the auxiliary period, preserving the solution obtained for the current period. The considered total period is then “rolled” to the next iteration and fixed.

The iterative part stops when the end of the current period reaches the end of the time horizon. At this point, all the pumping operations for the time horizon have been fully allocated and sequenced by the ASM. The execution proceeds then to the Scheduling Module.

4.3. Allocation and sequencing model formulation

The allocation and sequencing MILP model is the main process of the ASM. The model uses a discrete time approach. At each iteration, the considered total period is divided into events and intervals, where an event is a point of interest and an interval is a period of time between two consecutive events. First, the considered total period is divided into uniformly separated events, creating intervals of few hours (e.g. 480 hours divided into intervals of 12 hours). Afterwards, critical events are added into the considered grid. For instance, for each tank and pipeline maintenance period, the start and end events are also added due to their critical impact to the model. A tank maintenance alters temporarily the capacity of a product in a DC and a pipeline maintenance alters temporarily the flow rate limits. Consequently, some intervals can be shorter than the initially established uniform discrete time grid.

In the proposed ASM formulation, the constraints have been organized into 7 groups: (1) batch allocation constraints for the definition of product, volume and flow rate of a batch during an interval; (2) DC volume allocation constraints, to set the volume to be allocated to each DC during an interval; (3) product change constraints, to identify the intervals where a product change occurs; (4) batch size constraints, to monitor the size of the batches being pumped; (5) forbidden sequences of products constraints, to avoid the pump of incompatible sequences of products; (6) flow rate control constraints, to maintain the pump flow rate with low variations; (7) inventory control constraints for the management of the inventory levels in order to avoid shortages and surpluses.

The nomenclature used in the model is listed at the end of the paper. For readability purposes, the notation to parameters and sets starts with an uppercase letter and decision variables start with a lowercase letter. The MILP model uses sparse sets (e.g. *NP*, *NPE*), as they allow the consideration of only valid indexes for variables and constraints.

4.3.1. Batch allocation

The binary variable $a_{p,\,i}$ denotes if the product p is allocated to be pumped from the refinery during the interval i. Constraint (1) establishes that one product is pumped during an interval.

$$\sum_{p\in P} a_{p,i} = 1, \quad \forall i \in I \tag{1}$$

Continuous variables $fr_{p,\,i}$ and $v_{p,i}$ express, respectively, the flow rate and volume of the allocated product p to be pumped during the interval i. The value of $fr_{p,\,i}$ is bounded by the minimum and maximum pumping flow rate of the refinery ($Fp^{min}_{p,i}$ and $Fp^{max}_{p,i}$), as expressed by the constraints (2) and (3).

$$fr_{p,i} \geqslant Fp^{min}_{p,i} \times a_{p,i}, \quad \forall p \in P, i \in I \tag{2}$$

$$fr_{p,i} \leqslant Fp^{max}_{p,i} \times a_{p,i}, \quad \forall p \in P, i \in I \tag{3}$$

The values of the parameters $Fp^{max}_{p,i}$ and $Fp^{min}_{p,i}$ may not be the same for every interval i. During a full pipeline maintenance, $Fp^{max}_{p,i}$ and $Fp^{min}_{p,i}$ are set to 0. During a partial pipeline maintenance with, for example, a reduction of 60 % of the maximum efficiency compared to the regular operation, both $Fp^{max}_{p,i}$ and $Fp^{min}_{p,i}$ also decrease proportionally, 60 %, during the intervals included in this maintenance period.

The volume of product p pumped during interval i ($v_{p,i}$) is given by the specific duration of interval (T_i) multiplied by its flow rate ($fr_{p,\,i}$), as expressed by Eq. (4).

$$v_{p,i} = T_i \times fr_{p,i}, \quad \forall p \in P, i \in I \tag{4}$$

4.3.2. DC volume allocation

The continuous variable $v^{dc}_{n,p,i}$ represents the volume of product p that is expected to be delivered to DC n during interval i, which is a fraction of the volume of product p to be pumped during the same interval $v_{p,i}$. This variable $v^{dc}_{n,p,i}$ does not fix the value that will be delivered from the batch to that DC since the delivery operations will be better defined during the SM module. Constraint (5) establishes that the volume to be pumped during interval i must be equal to the sum of the volume expected to be delivered to all DCs.

$$\sum_{\{n,p,i\}\in NPI} v^{dc}_{n,p,i} = v_{p,i}, \quad \forall p \in P, i \in I \tag{5}$$

The big-M formulation of inequality (6) expresses that the volume of product p allocated to a DC n during an interval i can only be greater or equal than 0 if p is being pumped during the same interval ($a_{p,i} = 1$), otherwise the value of $v^{dc}_{n,p,i}$ is 0.

$$v^{dc}_{n,p,i} \leqslant a_{p,i} \times U, \quad \forall \{n, p, i\} \in NPI \tag{6}$$

The volume of product p delivered to DC n during interval i must not exceed a maximum volume that is equal to the maximum delivery flow rate of a product p to DC n (parameter $Fd^{max}_{n,p}$) multiplied by the interval size of i, as defined by the constraint (7).

$$v^{dc}_{n,p,i} \leqslant Fd^{max}_{n,p} \times T_i, \quad \forall \{n, p, i\} \in NPI \tag{7}$$

4.3.3. Product change

The binary variable $ch_{p,\,i}$ indicates that a product p that has being pumped during the previous interval $i-1$ is not being pumped in the current interval i. This means that $ch_{p,i} = 1$ when the product p is to be pumped in the interval $i-1$ ($a_{p,i-1} = 1$) and it is not to be pumped in interval i ($a_{p,i} = 0$). For all other cases $ch_{p,i} = 0$. For the first interval, this logical relationship is defined by constraint (8), which states that if product p is being initially pumped ($IP_p = 1$) and is different from the product to be pumped during the first interval ($a_{p,1} = 0$), then a change operation occurred. Each operation is accounted only for the product that stopped being pumped. For instance, if product $p1$ has being initially pumped ($IP_{p1} = 1$) and product $p2$ is to be pumped in the first interval, then $ch_{p1,1} = 1$ and $ch_{p2,1} = 0$.

$$ch_{p,i} = IP_p \times (1 - a_{p,i}), \quad \forall p \in P, i \in I \mid i = 1 \tag{8}$$

For the next intervals ($i > 1$), the previously product is now represented by the variable $a_{p,i-1}$. So, the $ch_{p,\,i}$ binary logical relationship needs to be established by constraints (9)–(11).

$$ch_{p,i} \leqslant a_{p,i-1}, \quad \forall p \in P, i \in I \mid i > 1 \tag{9}$$

$$ch_{p,i} \leqslant (1 - a_{p,i}), \quad \forall p \in P, i \in I \mid i > 1 \tag{10}$$

$$ch_{p,i} \geqslant a_{p,i-1} - a_{p,i}, \quad \forall p \in P, i \in I \mid i > 1 \tag{11}$$

4.3.4. Batch size

For the product p allocated during the first interval ($i = 1$), the batch size ($v^{batch}_{p,i}$) is equal to the volume allocated during i plus the initial batch volume (IV_p), as expressed by the big-M formulations (12) and (13).

$$v^{batch}_{p,i} - (v_{p,i} + IV_p) \leqslant U \times (1 - a_{p,i}), \quad \forall p \in P, i \in I \mid i = 1 \tag{12}$$

$$v^{batch}_{p,i} - (v_{p,i} + IV_p) \geqslant L \times (1 - a_{p,i}), \quad \forall p \in P, i \in I \mid i = 1 \tag{13}$$

For the other intervals ($i > 1$), the batch size is equal to the previous batch size plus the volume allocated during the current interval. Constraints (14) and (15) express the previous assertion.

$$v^{batch}_{p,i} - (v_{p,i} + v^{batch}_{p,i-1}) \leqslant U \times (1 - a_{p,i}), \quad \forall p \in P, i \in I \mid i > 1 \tag{14}$$

$$v^{batch}_{p,i} - (v_{p,i} + v^{batch}_{p,i-1}) \geqslant L \times (1 - a_{p,i}), \quad \forall p \in P, i \in I \mid i > 1 \tag{15}$$

The batch size of a product p is limited to a maximum volume that should not be exceeded during any interval i, as defined by the constraint (16). If the product p is not allocated in the interval i, then the value of $v^{batch}_{p,i}$ is 0.

$$v^{batch}_{p,i} \leqslant Lb^{max}_{p} \times a_{p,i}, \quad \forall p \in P, i \in I \tag{16}$$

If the product being pumped p changes during interval i, the batch size of the same product in the previous interval end $(i-1)$ has to be larger than the minimum volume limit, as stated by constraints (17) $(i=1)$ and (18) $(i>1)$.

$$IV_p \geqslant Lb^{min}_{p} \times ch_{p,i}, \quad \forall p \in P, i \in I \mid i = 1 \tag{17}$$

$$v^{batch}_{p,i-1} \geqslant Lb^{min}_{p} \times ch_{p,i}, \quad \forall p \in P, i \in I \mid i > 1 \tag{18}$$

4.3.5. Forbidden sequences of products

Due to the high contamination of the interface produced by some sequences of product batches, certain sequences are considered forbidden to the system. The binary matrix $PP_{p,p'}$ informs if the product p and p' are incompatible. For the operational characteristic to be considered, constraints (19) and (20) are included.

$$IP_p + a_{p',i} \leqslant 1, \quad \forall p \in P, p' \in P, i \in I \mid PP_{p,p'} = 1, \ i = 1 \tag{19}$$

$$a_{p,i-1} + a_{p',i} \leqslant 1, \quad \forall p \in P, p' \in P, i \in I \mid PP_{p,p'} = 1, \ i > 1 \tag{20}$$

4.3.6. Flowrate control

The mean flow rate is calculated as expressed by Eq. (21). It is expressed as the sum of all established flow rates for each product p during interval i divided by the sum of all intervals duration (hours of the actual period).

$$mfr = \frac{\sum_{p \in P, i \in I} T_i \times fr_{p,i}}{\sum_{i \in I} T_i} \tag{21}$$

Constraints (22) and (23) are related, respectively, to the upper and lower differences to the mean flow rate (mfr) during each interval. These differences are considered as violations and are, then, minimized by the model objective function, Eq. (27).

$$\sum_{p \in P} fr_{p,i} - mfr - umfr_i \leqslant 0, \quad \forall i \in I \tag{22}$$

$$\sum_{p \in P} fr_{p,i} - mfr + lmfr_i \geqslant 0, \quad \forall i \in I \tag{23}$$

4.3.7. Inventory control

Eq. (24) defines that the initial inventory of every pair DC-product $\{n, p\}$ at the event $e = 0$ equals to the parameter $ID_{n,\,p}$. Eq. (25) calculates the volumetric balance of the inventory level for the next events $(e > 0)$. The volume of a product p in DC n at an event e is the inventory level of the previous event minus the consumed demand during the interval i (period between $e-1$ and e) plus the initialization batch received during i and plus the volume allocated for the DC n during the same interval.

$$inv_{n,p,e} = ID_{n,p}, \quad \forall \{n, p, e\} \in NPE \mid e = 0 \tag{24}$$

$$\begin{aligned} inv_{n,p,e} &= inv_{n,p,e-1} - Dem_{n,p,i} + Rec_{n,p,i} + v^{dc}_{n,p,i}, \forall \{n, p, e\} \\ &\in NPE, \{n, p, i\} \in NPI \mid e > 0, \ i = e \end{aligned} \tag{25}$$

The set of inequalities in (26) defines the surplus and shortage violations of all considered inventory limits for every pair DC-product $\{n, p\}$ at every event e. Violations are minimized by the model objective function, Eq. (27).

$$\left.\begin{aligned} inv_{n,p,e} - vid^{cap}_{n,p,e} &\leqslant ID^{cap}_{n,p,e} \\ inv_{n,p,e} - vid^{max}_{n,p,e} &\leqslant ID^{max}_{n,p,e} \\ inv_{n,p,e} - vid^{maxtg}_{n,p,e} &\leqslant ID^{maxtg}_{n,p,e} \\ inv_{n,p,e} + vid^{mintg}_{n,p,e} &\geqslant ID^{mintg}_{n,p,e} \\ inv_{n,p,e} + vid^{min}_{n,p,e} &\geqslant ID^{min}_{n,p,e} \\ inv_{n,p,e} + vid^{emp}_{n,p,e} &\geqslant ID^{emp}_{n,p,e} \end{aligned}\right\} \forall \{n, p, e\} \in NPE \tag{26}$$

4.3.8. Objective function

The ASM model objective function (Eq. (27)) minimizes the sum of 3 groups of penalty costs: (1) inventory levels violation; (2) difference to the mean flow rate; and, (3) product change. The first group is responsible for controlling the inventory of each pair DC-product, minimizing surplus and shortage volume of the physical, operational and target inventory levels. The second group controls the flow rate of the network, diminishing the upper and lower difference to the mean flow rate. This objective is considered in order to maintain the flow rate with small variations, as operationally desired. The third group minimizes the number of changeover operations, allocating a cost per product change operation, that is also related to the contamination volume generated between two batches of different products.

The order of magnitude established for the cost factors of each ASM objective function term was established in conjunction with specialists of the oil company and the weights represent their preference over each term. Therefore, the values should be guided by follows: $Cint \gg Cid^{emp} \geq Cid^{cap} \gg Cid^{min} \geq Cid^{max} \geq (Cfr^{lmean}, Cfr^{umean}) \gg Cid^{mintg} \geq Cid^{maxtg}$. For the inventory levels, the minimum unitary costs (Cid^{emp}, Cid^{min}, and Cid^{mintg}) are higher than their respective maximum unitary costs (Cid^{cap}, Cid^{max}, and Cid^{maxtg}). Thus, a higher priority is associated to avoid inventory shortages over surpluses.

$$\begin{aligned} &\text{Minimize } z_{ASM} \\ &= \underbrace{\begin{aligned} &Cid^{cap} \cdot \sum_{\{n,p,e\} \in NPE} vid^{cap}_{n,p,e} + Cid^{emp} \cdot \sum_{\{n,p,e\} \in NPE} vid^{emp}_{n,p,e} \\ &+ Cid^{max} \cdot \sum_{\{n,p,e\} \in NPE} vid^{max}_{n,p,e} + Cid^{min} \cdot \sum_{\{n,p,e\} \in NPE} vid^{min}_{n,p,e} \\ &+ Cid^{maxtg} \cdot \sum_{\{n,p,e\} \in NPE} vid^{maxtg}_{n,p,e} + Cid^{mintg} \cdot \sum_{\{n,p,e\} \in NPE} vid^{mintg}_{n,p,e} \end{aligned}}_{\text{Group 1: inventory levels violation}} \\ &+ \underbrace{Cfr^{umean} \cdot \sum_{i \in I} umfr_i + Cfr^{lmean} \cdot \sum_{i \in I} lmfr_i}_{\text{Group 2: mean flow rate difference}} \\ &+ \underbrace{Cint \cdot \sum_{p \in P, i \in I} ch_{p,i}}_{\text{Group 3: product change}} \end{aligned} \tag{27}$$

5. Scheduling Module (SM)

The second module – Scheduling Module (SM) – is responsible for determining the delivery operations to all DCs during the horizon. A delivery operation consists in the receiving of an amount of volume from a batch passing along a DC. Fig. 8 shows the flowchart of this module. The module applies a structural decomposition, where the delivery operations are sequentially defined per pipeline segment, from the first to the last.

The scheduling preprocessing applies a network simplification through the use of heuristics and algorithms, which represent the

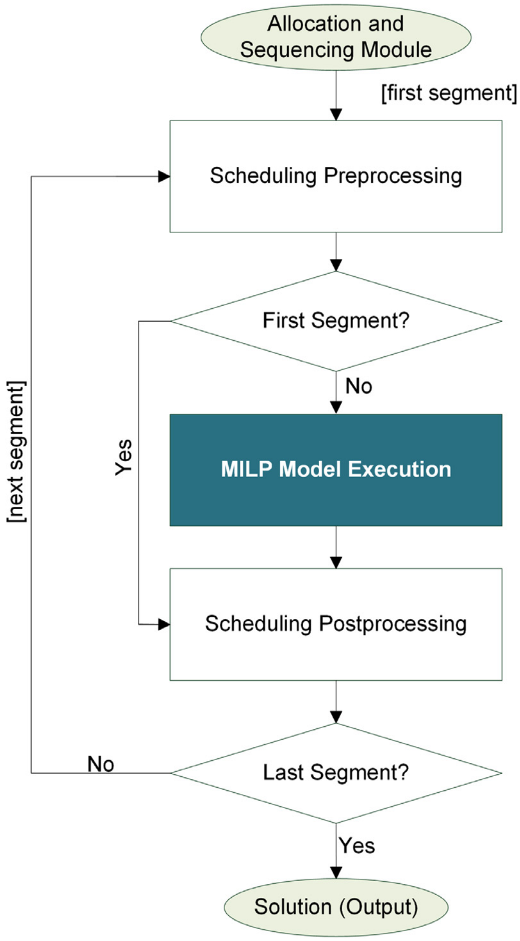


Fig. 8. Flowchart of the Scheduling Module.

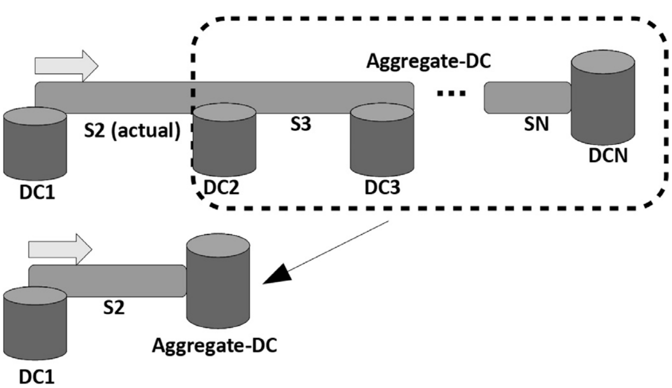


Fig. 9. Example of aggregate-DC constructed during the second segment (S2) iteration.

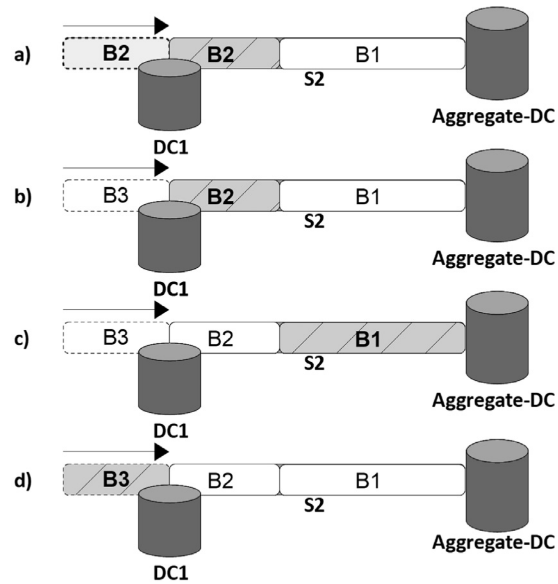


Fig. 10. Possible cases to calculate the size of a batch at an event. The hatched area represents the batch being considered. The cases are: (a) hatched batch is the same as the next batch part; (b) hatched batch is different from the next batch part; (c) same as (b), but hatched batch is not the closest batch; (d) hatched batch is the first batch part entering the segment.

real structure into a modified problem structure (Section 5.1 explains the concept and the processes involved). The MILP model, whose formulation is available in Section 5.2, is responsible for determining the delivery operations of the upstream DC of the segment being treated, as illustrated in Figs. 9 and 10. The scheduling postprocessing, detailed in Section 5.3, propagates all the batches to the next segment, timing when each batch part arrives at the next DC.

The execution flow initiates at the first segment (S1), which connects the refinery to the first DC (DC1). After finishing the iteration, the procedure addresses the next segment, until the last segment is postprocessed. Each iteration determines the delivery operations of the upstream DC of the current segment, which justifies the non execution of the MILP model during the first segment iteration, as in this case the upstream DC is the refinery and no delivery operations are needed. In this particular case, the postprocessing is performed directly, propagating and timing the initialization and allocating batches to the first DC. This is possible as the pumping times and flow rates of each batch are known. Then, for the second segment (S2), it is known the moment that each batch is starting to pass along the DC1, but the flow rate variation profile entering in S2 is not known, as this depends on delivery operations to be scheduled by the MILP model for the DC1 (upstream DC). After the MILP model execution, the algorithm is able to calculate the flow rate profile; making possible to know when each batch will start to pass along the DC2 (downstream DC). The process is repeated for S3 until SN. At the last segment (SN), the DCN receives all the remaining batches. As result the complete pipeline schedule for the considered horizon is obtained. It is important to notice that the approach is designed for addressing a generic number of segments (SN) and distribution centers (DCN). Thus, within the configuration of a single-source and multiple destinations the approach can serve to different n–values. This fact is further exploited in Section 6 (Results and Discussion).

5.1. Scheduling preprocessing

The scheduling preprocessing is responsible for simplifying the considered network and organizing the data required for the MILP model execution. For each iteration the preprocessing constructs a simplified network containing only the segment being considered, connecting its upstream DC to an aggregate-DC, which represents, in a simplified way, the remaining network, including all the following segments and DCs. Fig. 9 illustrates an example of the simplified network for the second segment. The last segment execution does not need a simplification, however still a data preprocessing is necessary.

The aggregate-DC is defined as the combination of the subsequent DCs and pipeline segments from the downstream DC of the actual segment until the last DC. The network simplification applies most of the heuristics used during the preprocessing of the ASM. The tanks of the same product in the aggregate-DC are combined as only one tank, summing the initial inventory and their inventory limits. The demand curve of each involved DC is forward shifted the average time in hours to transport from the first downstream DC to the respective DC using the average flow rate heuristic, as performed during the allocation and sequencing preprocessing. A tank maintenance period in an involved DC has to be perceived by the first downstream DC also forward shifted the same average transportation time before the start of the operation. The initialization batches of the segments of the aggregate-DC, existing at the beginning of the horizon, are propagated using the same estimation heuristic applied during the ASM. These deliveries are considered only during the actual iteration and treated as a receiving curve. The receiving curves are also forward shifted as the demand curves.

The input of the SM module considers the sequence of the batches reaching the upstream DC and their volume and flow rate. A batch may vary its flow rate for a period of time while passing through a segment, due to a delivery operation occurring in a previous DC or a pumping flow rate variation. So, for model representation purposes, the batches influenced by a delivery operation are broken into smaller parts with continuous flow rate during the preprocessing. For example, take the network of Fig. 6 and a 10,000 vu batch of gasoline reaching DC2 at 1,000 vu/hour, but in the last 4,000 vu a 500 vu/hour delivery operation starts at DC1. The batch will be split into two parts, one of 6,000 vu reaching DC2 at 1,000 vu/hour and other of 4,000 vu at 500 vu/hour. The MILP model considers each part of batch entering the simplified network as an interval period, so the start and end of a batch reaching the upstream DC are considered additional events.

After the network simplification and preprocessings, the module is ready to start the execution of the scheduling MILP model.

5.2. Scheduling model formulation

The MILP scheduling model is the main SM process and each iteration considers only one pipeline segment. The execution order follows the downstream sequence of segments: S2, S3,..., SN. Remembering, that the model is not executed for the first segment S1 since delivery operations are not needed for the refinery. The model considers as an interval the period corresponding to a batch part reaching the upstream DC with a constant flow rate. Hence, two events are assigned to the beginning and end of each interval. This information is obtained from the propagation algorithm (further explained in Section 5.3) of the last iteration. Since the delivery operations of the DC before the simplified network are already set, the batches are propagated to the upstream DC, obtaining intervals of constant flow rate. Therefore, the number of intervals of the SM model depends directly on the flow rate variation of the last pipeline segment.

In the proposed formulation, the constraints are organized into 6 groups: (1) batch size constraints, to identify the size of a batch inside the segment at an event; (2) constraints of batch volumetric coordinates, for monitoring the position of the batches inside the segment and also if the batch reached or passed a DC; (3) possibility of delivery constraints, to identify if it is possible to deliver a batch to a DC during a certain interval; (4) delivery operation constraints to consider operational limits, such as delivery flow rate and minimum delivery volume; (5) pipeline segment passage constraints, controlling the segment flow rate limit and variation; (6) inventory control constraints, to manage the inventory levels. The nomenclature used in the model is listed at the end of the paper. The same notation pattern as in the ASM MILP model is followed: parameters and sets start with an uppercase letter and decision variables start with a lowercase letter.

5.2.1. Batch size

The volume of the batches existing inside the segment in the beginning of the horizon ($e = 0$) is equivalent to the initialization batches volume, as established by constraint (28).

$$vm_{ib,e} = V_{ib}^{init}, \quad \forall ib \in B^{init}, e \in E \mid e = 0 \tag{28}$$

For the events after the first ($e > 0$), the batch size $vm_{b,e}$ depends on the volume entering the system during interval i (from $e = i - 1$ to $e = i$) and the volume delivered of batch b during the same interval. The volume entering the simplified network during interval i is represented by the parameter V_i^{pass}. To calculate the size of each batch inside the segment, we evaluate four possible cases during an event e, as illustrated in Fig. 10. The hatched area represents the batch being considered in each case.

Case (a) is when the hatched batch ($B2$) is the closest to the upstream DC ($DC1$) at the previous event ($e - 1$) and the next batch part to enter the network is another part of the same batch ($B2$). Hence, the batch size (constraint (29)) is equal to the size existing at the previous event plus the volume of the new part minus the volume delivered to the upstream and downstream DC during the interval from $e - 1$ to e. Case (b) is when the hatched batch was the closest batch to the upstream DC ($B2$) at the previous event and the next part is a different batch ($B3$). Case (c) is similar to (b), but the batch considered ($B1$) is not the last batch of the segment in the event $e - 1$. The batch size for the cases (b) and (c) are defined by the Eq. (30), where the batch size is equal to the previous batch size minus the volume delivered to both DCs during the interval. Case (d) is the first volume of a batch ($B3$) entering the system, being the size at the previous events equal to zero. Constraint (31) considers the case (d), where the batch size b is equal to the volume of the first part minus the volume delivered to both DCs during the interval. The equation includes the aggregate-DC since the volume of the batch part can be large enough to reach the aggregate-DC during the same interval.

$$vm_{b,e} = V_i^{pass} + vm_{b',e-1} - \sum_{\{b,i,n\} \in BIN} vt_{b,i,n}^{d}, \quad \forall \{b,e\} \in BE, \{b',e\} \in BE, i \in I \mid b = B_e^{last}, b' = B_{e-1}^{last}, b = b', i = e, e > 0 \tag{29}$$

$$vm_{b,e} = vm_{b,e-1} - \sum_{\{b,i,n\} \in BIN} vt_{b,i,n}^{d}, \quad \forall \{b,e\} \in BE, i \in I \mid b < B_e^{last}, i = e, e > 0 \tag{30}$$

$$vm_{b,e} = V_i^{pass} - \sum_{\{b,i,n\} \in BIN} vt_{b,i,n}^{d}, \quad \forall \{b,e\} \in BE, i \in I \mid b = B_{e-1}^{last} + 1, i = e, e > 0 \tag{31}$$

5.2.2. Batch volumetric coordinate

The farthest volumetric coordinate of a batch is called *head* coordinate and the closest coordinate to the upstream DC is called *tail* coordinate. So, for the first event ($e = 0$) the head coordinate of the initial batches are given by the parameter $HPos_{id}^{init}$, as presented by the constraint (32).

$$cd_{ib,e}^{head} = HPos_{ib}^{init}, \quad \forall ib \in B^{init}, e \in E \mid e = 0 \tag{32}$$

For the next events ($e > 0$), two cases are considered. The first (Eq. (33)) is when the batch b is the last batch at the event e, then the head coordinate is equal to the batch size $vm_{b,e}$. The second case (Eq. (34)) is when the batch b entered the segment but it is not the last batch at the event e, then the head is the coordinate of the next batch ($b + 1$, closer to the upstream DC) plus the batch size of b. For the batches that already left the segment, the head and tail are equal to the maximum volumetric coordinate (segment volume).

$$cd_{b,e}^{head} = vm_{b,e}, \quad \forall \{b,e\} \in BE \mid b = B_e^{last}, e > 0 \tag{33}$$

$$cd_{b,e}^{head} = vm_{b,e} + cd_{b',e}^{head}, \quad \forall \{b,e\} \in BE, \{b',e\} \in BE \mid b = b' - 1, b \neq B_e^{last}, e > 0 \tag{34}$$

The tail coordinate relates to the head coordinate as expressed in Eq. (35). Where the tail coordinate is equal to the head coordinate minus the batch size.

$$cd_{b,e}^{tail} = cd_{b,e}^{head} - vm_{b,e}, \quad \forall \{b,e\} \in BE \tag{35}$$

The head and tail coordinates of batch b must be greater or equal to the previous event, in other words, the batches move unidirectionally, they cannot move backward. This property is defined by the inequalities (36) and (37).

$$cd_{b,e}^{head} \geqslant cd_{b,e-1}^{head}, \quad \forall \{b,e\} \in BE \mid e > 0 \tag{36}$$

$$cd_{b,e}^{tail} \geqslant cd_{b,e-1}^{tail}, \quad \forall \{b,e\} \in BE \mid e > 0 \tag{37}$$

If the head of a batch b already passed the DC n at the event e, which means $hb_{b,e,n} = 1$, then the head coordinate of the batch is greater or equal to the volumetric coordinate of the DC n, otherwise $hb_{b,e,n} = 0$. The big-M formulation that relates the binary variable $hb_{b,e,n}$ with the continuous variable $cd_{b,e}^{head}$ is defined by the constraints (38) and (39). Notice that the parameter ϵ is a dimensionally small value.

$$cd_{b,e}^{head} - VC_n \geqslant L \times (1 - hb_{b,e,n}), \quad \forall \{b,e,n\} \in BEN \tag{38}$$

$$cd_{b,e}^{head} - VC_n \leqslant (U + \epsilon) \times hb_{b,e,n} - \epsilon, \quad \forall \{b,e,n\} \in BEN \tag{39}$$

If the tail of a batch b already passed the DC n at the event e, which means $tb_{b,e,n} = 1$, then the tail coordinate of the batch is greater or equal to the volumetric coordinate of the DC n, otherwise $tb_{b,e,n} = 0$. The constraints (40) and (41) express this relation.

$$cd_{b,e}^{tail} - VC_n \geqslant L \times (1 - tb_{b,e,n}), \quad \forall \{b,e,n\} \in BEN \tag{40}$$

$$cd_{b,e}^{tail} - VC_n \leqslant (U + \epsilon) \times tb_{b,e,n} - \epsilon, \quad \forall \{b,e,n\} \in BEN \tag{41}$$

If the tail of batch b passed the DC n at the event e, then the head already reached n, as expressed in constraint (42).

$$tb_{b,e,n} \leqslant hb_{b,e,n}, \quad \forall \{b,e,n\} \in BEN \tag{42}$$

5.2.3. Possibility of delivery

The binary variable $pd_{b,i,n}$ is equal to 1 when there is the possibility to deliver the batch b during interval i to the DC n, otherwise is 0. So, for the upstream DC ($n = 0$), it is only possible to deliver the batch reaching this DC during the same interval (parameter B_i^{pass}), as indicated by constraints (43) and (44).

$$pd_{b,i,n} = 1, \quad \forall \{b,i,n\} \in BIN \mid b = B_i^{pass}, n = 0 \tag{43}$$

$$pd_{b,i,n} = 0, \quad \forall \{b,i,n\} \in BIN \mid b \neq B_i^{pass}, n = 0 \tag{44}$$

For the aggregate-DC ($n = 1$), there is the possibility to deliver batch b during interval i (from event $e = i - 1$ to $e = i$) when the head of batch b at the end of this interval ($e = i$) reaches the aggregate-DC and the tail of the same batch still has not passed the aggregate-DC at the start of this interval ($e = i - 1$). The logic relation between the binary variables is expressed by the inequalities (45)–(47).

$$pd_{b,i,n} \leqslant hb_{b,e,n}, \quad \forall \{b,e,n\} \in BEN, \{b,i,n\} \in BIN \mid i = e, e > 0, n = 1 \tag{45}$$

$$pd_{b,i,n} \leqslant 1 - tb_{b,e-1,n}, \quad \forall \{b,e,n\} \in BEN, \{b,i,n\} \in BIN \mid i = e, e > 0, n = 1 \tag{46}$$

$$pd_{b,i,n} \geqslant hb_{b,e,n} - tb_{b,e-1,n}, \quad \forall \{b,e,n\} \in BEN, \{b,i,n\} \in BIN \mid i = e, e > 0, n = 1 \tag{47}$$

5.2.4. Delivery operation

Constraint (48) expresses that the total volume delivered during an interval i is equal to the volume reaching the upstream DC, maintaining the volume balance of the network.

$$\sum_{\{b,i,n\} \in BIN} vt_{b,i,n}^{d} = V_i^{pass}, \quad \forall i \in I \tag{48}$$

The binary variable $ss_{b,i,n}$ is equal to 1 if a volume greater than 0 is delivered by the batch b during interval i to DC n, otherwise is 0, which means $vt_{b,i,n}^{d} = 0$. The volume of $vt_{b,i,n}^{d}$ should also not be less than the volume entering the system V_i^{pass} during the same interval. This relation is expressed by the inequalities (49) and (50).

$$vt_{b,i,n}^{d} \geqslant (L - \epsilon) \times (1 - ss_{b,i,n}) + \epsilon, \quad \forall \{b,i,n\} \in BIN \tag{49}$$

$$vt_{b,i,n}^{d} \leqslant V_i^{pass} \times ss_{b,i,n}, \quad \forall \{b,i,n\} \in BIN \tag{50}$$

A delivery operation can happen only if the delivery possibility exists, as indicated by the constraint (51).

$$ss_{b,i,n} \leqslant pd_{b,i,n}, \quad \forall \{b,i,n\} \in BIN \tag{51}$$

The binary variable $ss_{b,n}^{dc}$ is 1 if at least one delivery operation is scheduled for the batch b to DC n, otherwise is 0, meaning that the DC does not received any amount of batch b. Constraints (52) and (53) certify this relation between $ss_{b,n}^{dc}$ and $ss_{b,i,n}$.

$$\sum_{\{b,i,n\} \in BIN} ss_{b,i,n} \geqslant (L - \epsilon) \times (1 - ss_{b,n}^{dc}) + \epsilon, \quad \forall b \in B, n \in N \tag{52}$$

$$\sum_{\{b,i,n\} \in BIN} ss_{b,i,n} \leqslant U \times ss_{b,n}^{dc}, \quad \forall b \in B, n \in N \tag{53}$$

When a delivery operation of a batch b to the upstream DC ($n = 0$) is scheduled, at least a minimum volume must be respected, as an operation constraint of the system. This constraint is expressed by the inequality (54).

$$\sum_{\{b,i,n\} \in BIN} vt_{b,i,n}^{d} \geqslant Ld_p^{min} \times ss_{b,n}^{dc}, \quad \forall b \in B, \{n,p\} \in NP \mid p = P_b, n = 0 \tag{54}$$

The delivery flow rate of a batch b during an interval i to a DC n is equal to the delivery volume divided by the duration of the considered interval, as given by the Eq. (55).

$$fr_{b,i,n}^{d} = \frac{vt_{b,i,n}^{d}}{T_i}, \quad \forall \{b,i,n\} \in BIN \tag{55}$$

A delivery operation is also constrained by the maximum and minimum delivery flow rate specified for a particular product p to the DC ($n = 0$). The delivery flow rate must be less or equal the maximum flow rate (constraint (56)). And greater than or equal the

Please cite this article as: W.H.T. Meira et al., A matheuristic decomposition approach for the scheduling of a single-source and multiple destinations pipeline system, European Journal of Operational Research (2018), https://doi.org/10.1016/j.ejor.2018.01.032

minimum flow rate, when a delivery is scheduled for the interval (constraint (57)).

$$fr^{d}_{b,i,n} \leqslant Fd^{max}_{n,p}, \quad \forall \{b, i, n\} \in BIN, \{n, p\} \in NP \mid p = P_b, n = 0 \quad (56)$$

$$fr^{d}_{b,i,n} \geqslant Fd^{min}_{n,p} \times ss_{b,i,n}, \quad \forall \{b, i, n\} \in BIN, \{n, p\} \in NP \mid p = P_b, n = 0 \quad (57)$$

5.2.5. Pipeline segment passage

The flow rate of a batch entering the pipeline segment during an interval i is equal to the flow rate of the batch part reaching the upstream DC minus the flow rate of the delivering operation of the same interval. This flow rate balance is expressed by the Eq. (58).

$$fr^{s}_{b,i} = F^{pass}_{i} - fr^{d}_{b,i,n}, \quad \forall \{b, i, n\} \in BIN \mid b = B^{pass}_{i}, n = 0 \quad (58)$$

The inequalities (59) and (60) are constraints of pipeline segment maximum and minimum flow rate, respectively. The flow rate limits should be respected. The model allows, however, flow rate violations, which are minimized in the objective function. The flow rate limits may also vary, depending on the product.

$$fr^{s}_{b,i} \leqslant Fs^{max}_{p} + vfr^{max}_{b,i}, \quad \forall \{b, i\} \in BI \mid p = P_b \quad (59)$$

$$fr^{s}_{b,i} \geqslant Fs^{min}_{p} - vfr^{min}_{b,i}, \quad \forall \{b, i\} \in BI \mid p = P_b \quad (60)$$

One of the operational objectives of the model is to maintain the segment flow rate as constant as possible with the purpose of generating fewer parts of batches. Constraints (61) and (62) count, respectively, the upper and lower flow rate difference between the actual and the previous interval. The flow rate difference is then weighted and minimized by the model objective function. The lower the number of flow rate fluctuations, the lower is the number of batch parts and, consequently, fewer intervals are created in the next iteration.

$$fr^{s}_{b,i} - fr^{s}_{b',i'} + vfr^{udiff}_{i'} \geqslant 0, \ \forall \{b, i\} \in BI, \{b', i'\} \in BI \mid i = i' - 1, \quad (61)$$
$$i' > 1, b = B^{pass}_{i}, b' = B^{pass}_{i'}$$

$$fr^{s}_{b,i} - fr^{s}_{b',i'} - vfr^{ldiff}_{i'} \leqslant 0, \ \forall \{b, i\} \in BI, \{b', i'\} \in BI \mid i = i' - 1, \quad (62)$$
$$i' > 1, b = B^{pass}_{i}, b' = B^{pass}_{i'}$$

5.2.6. Inventory control

The initial inventory (event $e = 0$) is defined by the input parameter $ID_{n,p}$ for every DC n and product p, as indicated by the Eq. (63). In a complementary manner, Eq. (64) calculates the volumetric balance of the inventory level for the other events ($e > 0$). The inventory volume of a product p in DC n at an event e is the previous inventory volume minus the consumed demand plus the initialization batch receiving and plus the total volume delivered during interval i (from $e - 1$ to e).

$$inv_{n,p,e} = ID_{n,p}, \quad \forall \{n, p, e\} \in NPE \mid e = 0 \quad (63)$$

$$inv_{n,p,e} = inv_{n,p,e-1} - Dem_{n,p,i} + Rec_{n,p,i} + \sum_{b \in B \mid p = P_b} vt^{d}_{b,i,n}, \quad \forall \{n, p, e\} \in NPE, \{n, p, i\} \in NPI \mid e > 0, i = e \quad (64)$$

The set of inequalities in (65) defines the surplus and shortage violations of all considered inventory limits for every pair DC-product $\{n, p\}$ at every event e. The violations are then minimized by the model objective function, Eq. (66).

$$\left.\begin{array}{l} inv_{n,p,e} - vid^{cap}_{n,p,e} \leqslant ID^{cap}_{n,p,e} \\ inv_{n,p,e} - vid^{max}_{n,p,e} \leqslant ID^{max}_{n,p,e} \\ inv_{n,p,e} - vid^{maxtg}_{n,p,e} \leqslant ID^{maxtg}_{n,p,e} \\ inv_{n,p,e} + vid^{mintg}_{n,p,e} \geqslant ID^{mintg}_{n,p,e} \\ inv_{n,p,e} + vid^{min}_{n,p,e} \geqslant ID^{min}_{n,p,e} \\ inv_{n,p,e} + vid^{emp}_{n,p,e} \geqslant ID^{emp}_{n,p,e} \end{array}\right\} \forall \{n, p, e\} \in NPE \quad (65)$$

5.2.7. Objective function

The SM model objective function (Eq. (66)) minimizes the sum of 3 groups of penalty costs: (1) inventory levels violation; (2) pipeline segment flow rate limits; (3) pipeline segment flow rate difference (or variation). The first group is responsible for controlling the inventory of each pair DC-product, minimizing surplus and shortage of physical, operational, and target levels. Moreover, the penalty for a violation at the upstream DC ($n = 0$) is higher (double) than the aggregate-DC ($n = 1$), prioritizing the first DC in analysis during the current iteration. On the other hand, the aggregate-DC, that is, the second DC considered in the current iteration, is a simplification that will be later better addressed. Thus, there is a possibility of adjusting eventual problems regarding the aggregate-DC in the next iterations.

The second group minimizes the maximum and minimum pipeline segment flow rate violations. Where the maximum flow rate is a physical limit of the network and the minimum a desired operational limit. Thus, the maximum flow rate penalty cost is higher than the minimum. The last group minimizes the segment flow rate difference between two sequential intervals, decreasing the flow rate variation of the batches reaching the next DC, a consideration that implies fewer batch parts and intervals to be considered in the next iterations.

The order of magnitude of the cost factors of each term of the SM objective function was obtained as the cost factors of the ASM objective function (Eq. (27)), considering the analysis of the oil company specialists. As a result, cost factor values should be guided as the following order of magnitude: $Cfr^{max} \gg Cfr^{min} \gg Cid^{emp} \geq Cid^{cap} \gg (Cfr^{ldiff}, Cfr^{udiff}) \geq Cid^{min} \geq Cid^{max} \gg Cid^{mintg} \geq Cid^{maxtg}$. As in the ASM MILP model, the minimum levels costs are weighted more than the maximum level costs.

$$\begin{aligned} \text{Minimize } z_{SM} \\ = \underbrace{\begin{aligned} & Cid^{cap} \cdot \sum_{\{n,p,e\} \in NPE} vid^{cap}_{n,p,e}/(n+1) + Cid^{emp} \cdot \sum_{\{n,p,e\} \in NPE} vid^{emp}_{n,p,e}/(n+1) \\ & + Cid^{max} \cdot \sum_{\{n,p,e\} \in NPE} vid^{max}_{n,p,e}/(n+1) + Cid^{min} \cdot \sum_{\{n,p,e\} \in NPE} vid^{min}_{n,p,e}/(n+1) \\ & + Cid^{maxtg} \cdot \sum_{\{n,p,e\} \in NPE} vid^{maxtg}_{n,p,e}/(n+1) + Cid^{mintg} \cdot \sum_{\{n,p,e\} \in NPE} vid^{mintg}_{n,p,e}/(n+1) \end{aligned}}_{\text{Group 1: inventory levels violation}} \\ \underbrace{+ Cfr^{max} \cdot \sum_{\{b,i\} \in BI} vfr^{max}_{b,i} + Cfr^{min} \cdot \sum_{\{b,i\} \in BI} vfr^{min}_{b,i}}_{\text{Group 2: flow rate limits}} \\ \underbrace{+ Cfr^{udiff} \cdot \sum_{i \in I} vfr^{udiff}_{i} + Cfr^{ldiff} \cdot \sum_{i \in I} vfr^{ldiff}_{i}}_{\text{Group 3: flow rate difference}} \quad (66) \end{aligned}$$

5.3. Scheduling postprocessing

After the execution of each model, we obtain the delivery operations scheduled for the upstream DC during the entire horizon. A delivery operation has an interval, volume, and flow rate associated. With this set of information it is possible to obtain the flow

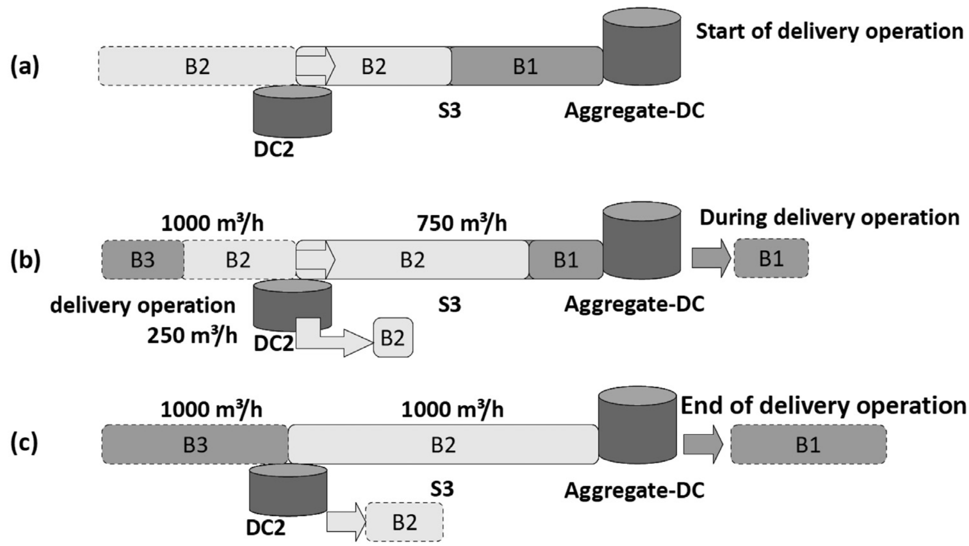


Fig. 11. Example of the propagation algorithm.

rate profile reaching the downstream DC of the considered segment. The proposed postprocessing algorithm subtracts from the flow rate of the batches, reaching the upstream DC the delivery operation flow rate during the scheduled period. Thus, we can propagate each batch through the pipeline segments until it reaches the downstream DC, making possible to obtain timing details. Then, the information obtained is used in the next iteration.

An example of the postprocessing propagation algorithm is illustrated in Fig. 11. The batch B2 propagates part of the passing period at 1,000 vu/hour. Then, while a scheduling delivery operation occurs (b), its flow rate and volume decreases. The algorithm considers the flow rate variations and determines the timing of each batch reaching the next DC.

For the last segment (SN), the postprocessing also determines all the delivery operations to be realized in DCN; the distribution center must receive all the batches that reach it. After the last segment postprocessing, all the delivery operations for the time horizon and a complete solution for the given instance is obtained.

6. Results and discussion

The proposed decomposition approach is validated by solving instances of two case studies, where the first one considers illustrative networks and the second one a real-world network. The approach was developed using Java as programming language and the IBM ILOG CPLEX *Optimization Studio* 12.6. All the obtained results were executed on a PC platform with an Intel i7-6500 @ 2.50 GHz processor, 16 GB of RAM running Windows 10 64-bits as operational system.

6.1. Case Study 1: illustrative network

The case study 1 (CS1) has the purpose of validating the proposed decomposition approach against several instances of illustrative networks with different numbers of products and taking into account different operational aspects. Fig. 12 shows the illustrative network with 3 DCs. Additionally, we considered also variations of the same network with 2 and 1 DCs.

All the instances were run with the same configuration parameters. The time horizon was set to 21 days. The current and auxiliary periods of each rolling horizon iteration of ASM (see Fig. 7) were set to 3 and 5 days, respectively. For the current period, a

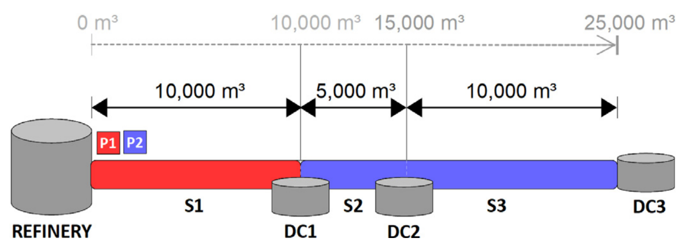


Fig. 12. CS 1 - Illustrative pipeline network with 3 DCs.

smaller value would mean more iterations, and a larger number would increase the model dimensions. For the auxiliary period, the value should be large enough to allow the model to analyze the behavior of the system for the scheduled batches of the current period and avoid, in advance, problems due to maintenance periods or demand variations. With the defined configurations, the ASM model runs 7 iterations during the rolling horizon processing. On the other hand, the SM model depends on the structure of the network: excluding the first segment, the model executes once for each next segment. For the illustrative network with just one destination, only the post-processing of the SM is executed. The interval size of the ASM was set to 8 hours. For the SM models, as explained in Section 5.2, intervals are defined as start and end periods of batch parts with constant flow rate arriving at the upstream DC of the actual segment. Further, Table A.1 in the Appendix presents all the cost factors set for the two MILP models. These values follow the order of magnitude established for the ASM and SM models, as defined in Sections 4.3.8 and 5.2.7, respectively.

Data for the instances are defined from Table A.2 to A.6 in the Appendix. Table A.2 contains the initialization batches existing at the beginning of the horizon. Table A.3 contains the minimum and maximum batch sizes of each product. Due to high contamination, forbidden sequences are defined, as shown by the incompatibility matrix in Table A.4. Table A.5 contains the total demand of each pair DC-product for the 21-days of time horizon. As explained in Section 2, the demand consumption occurs in a constant flow rate fashion during the time horizon. Table A.6 presents all the pair DC-product characteristics: minimum and maximum delivery flow

Table 1
CS1 - CPU time in seconds for each of the 24 different instances.

	Without maintenance			With maintenance		
	DC1	DC1,DC2	DC1,DC2,DC3	DC1	DC1,DC2	DC1,DC2,DC3
P1,P2	1.2	2.2	3.4	1.4	2.4	4.6
P1,P2,P3	6.7	11.9	19.4	5.0	12.9	19.9
P1,P2,P3,P4	24.0	177.6	78.7	27.1	128.5	68.3
P1,P2,P3,P4,P5	91.7	475.4	544.5	94.2	614.5	565.0

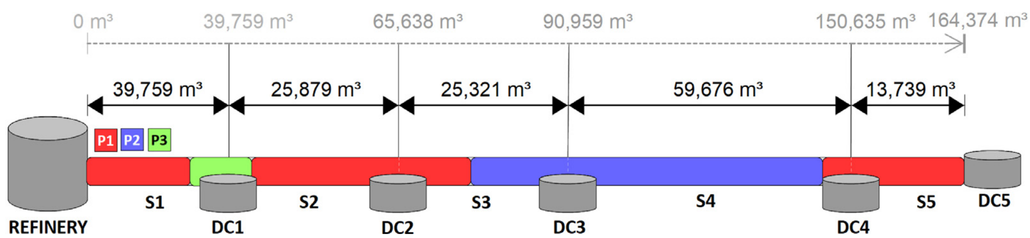


Fig. 13. CS2 - Real-world pipeline network.

rate, minimum volume of a delivery operation, initial inventory and all inventory levels (physical, operational and target levels). The pumping and segment (S1-S3) flow rate limit is between 10 and 500 vu/hour.

The instances tested were a combination of the number of active DCs of the illustrative network in Fig. 12: from 1 to 3 DCs; number of products from 2 to 5; and with or without maintenance periods. When considering instances with maintenance periods, we added a tank maintenance period and a total pipeline maintenance period with 24 hours of duration each. Thus, 24 different instances, representing different network topologies and conditions, were analyzed within case study 1. Table 1 presents the CPU time for each complete instance run. The number of products increases the CPU time rapidly and a similar behavior is observed as more DCs are operational. Also, almost all the instances with maintenance periods demonstrated an increase in CPU time, when compared against their similar instances, but without maintenance periods. This fact occurs due to critical events – start and end of the maintenance periods – also added to the event list of the ASM models, increasing the number of intervals to be treated.

The proposed approach obtained solutions in a reasonable CPU time for all the different combinations of instances for the case study 1. We validated the capability of the approach solving different instances with different aspects and observed the computational behavior of the system as more DCs and products are added to the straight pipeline network. In relation to the quality of the solution, no physical inventory and flow rate limit violation were observed in none of the solutions. Based on the validated results for case study 1 for different instances, Section 6.2 addresses a real-world network with 5 DCs and 3 products: the case study 2.

6.2. Case Study 2: real-world network

Case study 2 (CS2) solves a real-world system formed by a multiproduct pipeline that connects a refinery to five destinations. The network distributes products to part of the southeast and central-west regions of Brazil. The pipeline length and volume are 964 kilometers and 164,374 cubic meters respectively. Fig. 13 represents the considered network and the volumetric dimension of each segment (S1–S5). On the gray dashed line it is indicated the accumulated volume of the network at each DC (DC1–DC5).

The proposed decomposition approach is used for solving a real instance considering three examples: (1) the base instance; (2) an extension of this base instance where one tank maintenance and two pipeline maintenance periods are considered; and (3) model performance tests were developed as the uniform interval size of the ASM decreases. The instance was provided by the oil company that owns the studied network, however, the input data had to be modified in order to preserve industrial confidentiality. The studied network operates by distributing 3 oil products (P1, P2 and P3).

The first two examples were run using the same configuration parameters. The time horizon to be scheduled is set to 30 days. The current and auxiliary periods of each rolling horizon iteration of ASM (see Fig. 7) were set to 5 and 10 days, respectively, meaning 6 iterations during the ASM rolling horizon processing. The interval size of the ASM was set to 8 hours. For the third example, the tests were executed decreasing progressively the interval size parameter in order to observe the performance of the approach and the solution quality. The cost factors set for the MILP models were the same used for the case study 1 (Table A.1).

6.2.1. Example 1: base instance

For the base instance, the pumping rate limits at the refinery are between 700 and 1,200 vu/hour, which is the same flow rate limit of the pipeline segment S1. The other segments (S2-S5) have a minimum/maximum flow rate of 400/1,200 vu/hour, 400/1,200 vu/hour, 300/850 vu/hour and 100/450 vu/hour, respectively, values established based on the pipeline operations. The remaining instance data are defined from Table A.7 to A.11 in the Appendix.

Table A.7 contains the initialization batches existing at the beginning of the horizon (also illustrated in Fig. 13). Table A.8 contains the minimum and maximum batch size of each product. Forbidden sequences are defined in Table A.9. Since the interface between a batch of product P2 and P3 is prohibited, the product P1 will always be in between these two. Table A.10 contains the total demand of each pair DC-product for the complete time horizon. Lastly, Table A.11 exhibits all the pair DC-product characteristics.

The complete execution of the solution approach for the base instance took a total of 279.6 seconds, being 274.7 seconds during ASM models, 4.7 seconds during SM models and around 0.2 seconds spent with input reading, output writing, and execution of the heuristics and algorithms of the approach. Table 2 presents the dimension of each model executed during the ASM and SM. The ASM model is executed 6 times (30-day horizon divided by 5 –

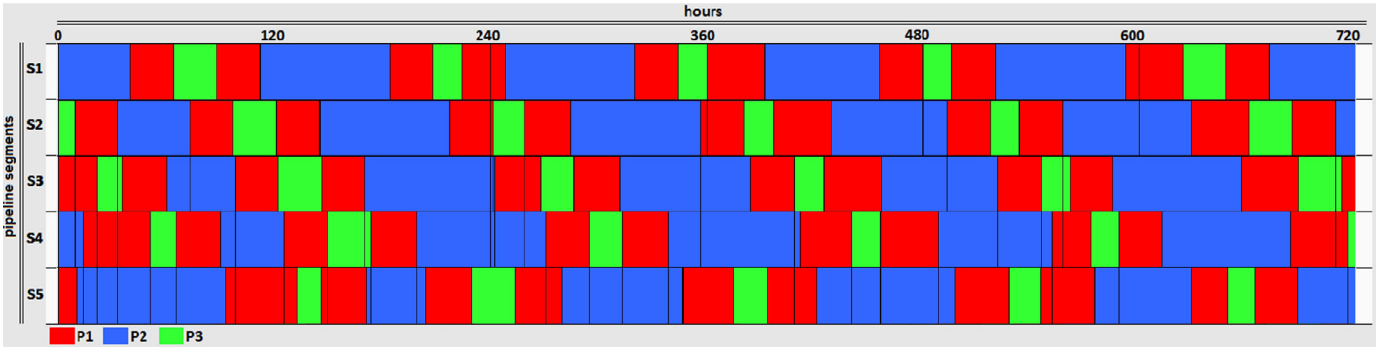


Fig. 14. CS2 - Gantt chart of the obtained solution for the base instance (example 1).

Table 2
CS2 - MILP models statistics for the base instance run (example 1).

Module	Iteration	Constraints	Variables	Bin. Var.	CPU time (s)	Gap (%)
ASM	1	7,885	6,272	270	24.1	0
ASM	2	7,885	6,272	270	12.1	0
ASM	3	7,885	6,272	270	49.6	0
ASM	4	7,885	6,272	270	70.6	0
ASM	5	7,885	6,272	270	41.1	0
ASM	6	7,885	6,272	270	77.2	0
SM	1	3,713	2,566	740	0.3	0
SM	2	4,612	3,143	912	0.6	0
SM	3	7,235	4,541	1,630	1.8	0
SM	4	6,427	4,288	1,314	2.0	0

the size of the current period) and the SM model is executed for the last 4 segments. For the SM model dimensions, we observe a tendency of a model growth as it moves downstream the network. This fact is explained by the higher number of batch parts generated due to deliveries happening in the upstream DCs, which implies a less constant segment flow rate. Also, the CPU time for the SM models seems to increase as the dimension increases. All models were executed until optimality, which means that all optimality gaps reached 0%.

The obtained solution scheduled 21 batches to be pumped by the refinery during the 30-day time horizon. Fig. 14 shows the obtained Gantt chart of the solution. The pipeline segments are organized on the vertical axis and the time horizon (from 0 hour until 720 hour) in the horizontal axis with different colors assigning different products. The scheduled batches to be pumped are represented in the first horizontal line, entering the segment S1.

Fig. 15(a–e) present some of the inventory profiles along the 30-day time horizon. The selected profiles are the storage of product P1 in all DCs (DC1–DC5). The profiles indicate that the inventory levels tend to go down and up along the scheduling horizon, but the curves remain inside inner levels. A negative slope happens due to the demand consumption and a positive slope is caused by a delivery operation only when the delivery flow rate is superior to the demand flow rate, otherwise, the slope will be just less negative. The inventory level of product P1 in DC4 and DC5 (Fig. 15 d–e) shows the existence of some operational level violations. The model is not able to maintain the inventory among the recommended target levels at the start of the horizon. This fact occurred due to the initial status of the system (initialization batches positions and initial inventory), therefore, it was not possible to avoid such temporary violations. However, as soon as the scheduled batches adjust the system to a favorable position, the inventory levels are better controlled. Important to mention that, for the obtained solution, no shortage/surplus of physical empty/capacity levels occur during the entire horizon for any pair DC-product and also no physical flow rate violation in any pipeline segment occurred. The pumping flow rate profile during the time horizon is illustrated in Fig. 16.

This execution illustrates the capability of the approach to solve a real-life instance in few minutes without major problems, as shown in the Gantt chart (Fig. 14), inventory profiles (Fig. 15) and pumping flow rate profile (Fig. 16). In Section 6.2.2, we increase the complexity of the instance, including tank and pipeline maintenance periods, in order to evaluate the capability of the approach when treating these aspects.

6.2.2. Example 2: extended instance

This extended instance aims at representing practical conditions where maintenance periods are imposed to the scheduling system and may occur more than once during a scheduling horizon. The base instance was extended so as to include a tank maintenance and two pipeline maintenance periods scheduled to be performed during the time horizon. The tank maintenance is scheduled to a P1 tank of 8,600 vu located in DC4 and it starts at 00h00 of day 20 and ends at 00h00 of day 22 (2 days), which is from 480 hour until 528 hour. The first pipeline maintenance is total (pumping stops) and it is scheduled to start at 12h00 of day 8 and end at 04h00 of day 9 (16 hours of duration, from 204 hour until 220 hour). The second pipeline maintenance is partial, decreasing 50 % of the pumping efficiency, starting at 00h00 of day 13 and ending at 00h00 of day 14 (24 hours of duration, from 312 hour until 336 hour).

The complete execution of the extended instance took a total of 255.4 seconds, being 241.7 seconds during ASM models, 13.5 seconds during SM models and around 0.2 seconds spent with input reading, output writing, and execution of the heuristics and algorithms of the approach. Table 3 presents the statistics of each ASM and SM model executed. The ASM model dimensions of the extended instance are equal to the base instance dimensions, except for the two first iterations due to the total pipeline mainte-

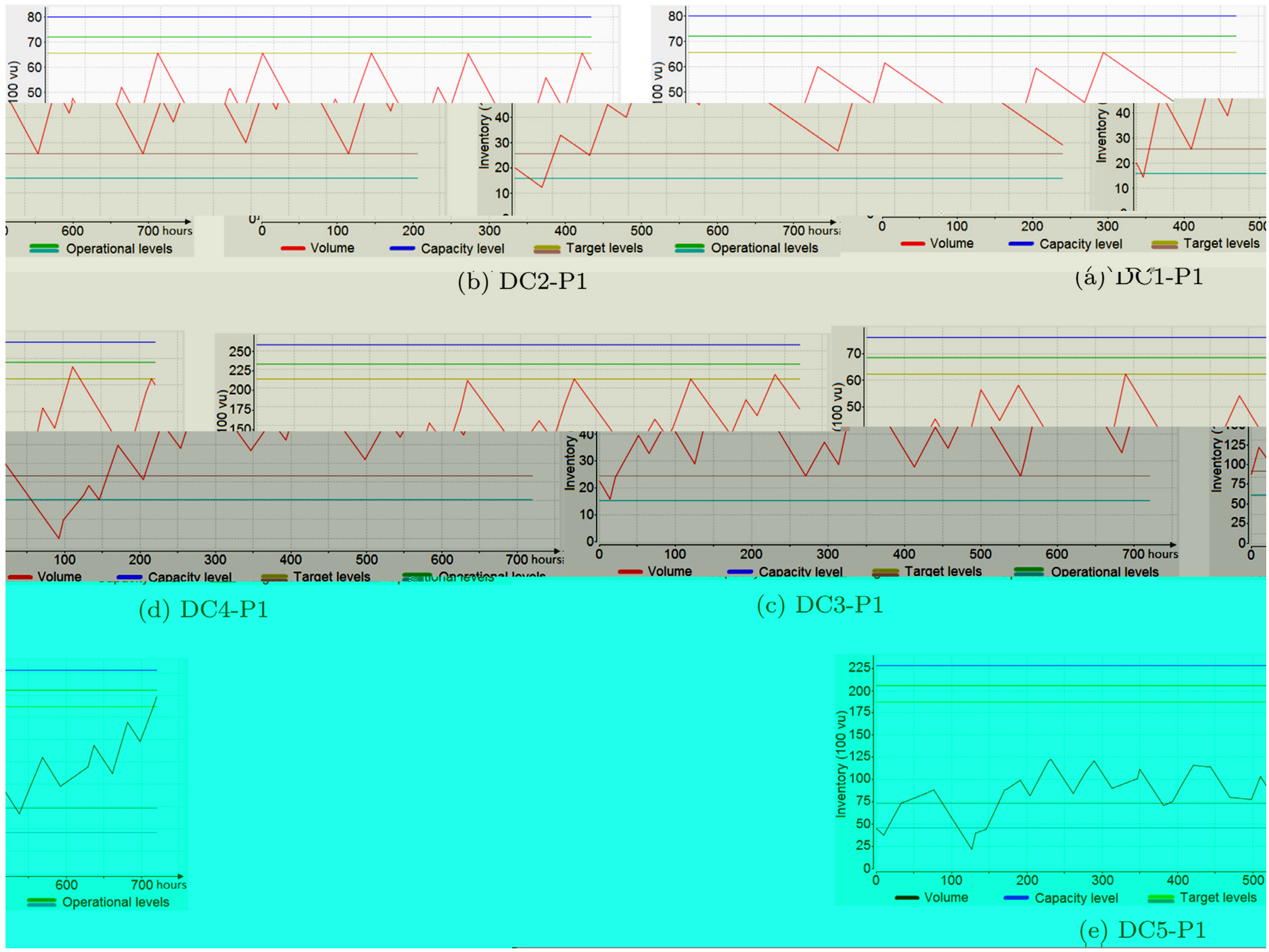


Fig. 15. CS2 - Inventory profile of product P1 in all DCs (DC1–DC5) for the entire time horizon of the base instance (example 1). Where the *X*-axis is the time horizon in hours (from 0 hour to 720 hour) and *Y*-axis is the volume in 100 vu.

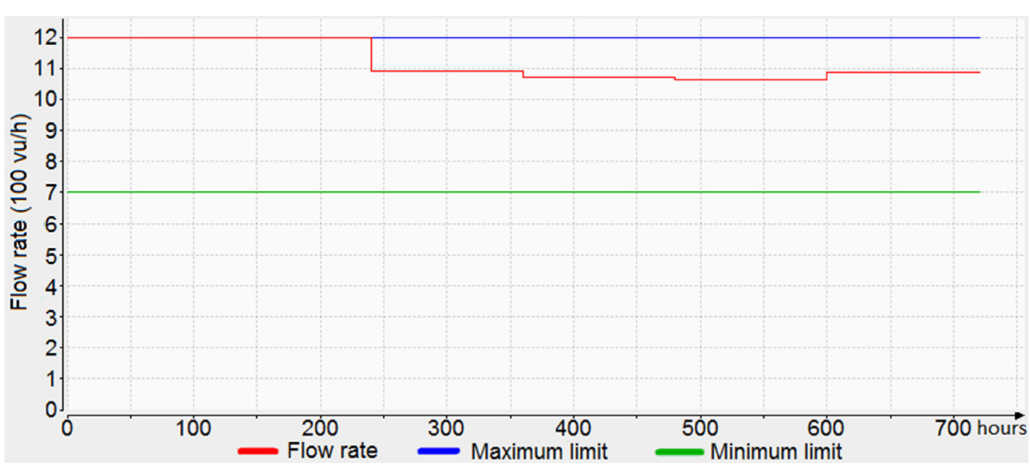


Fig. 16. CS2 - Pumping flow rate profile for the base instance (example 1).

nance period (204–220 hour) existing in their interval range. So, for the first two iterations, the critical maintenance events (start and end) complement the uniform events set by the configuration parameter, increasing the dimension of these models. The start and end times of the tank maintenance (312 hour and 336 hour) and partial pipeline maintenance (480 hour and 528 hour) were already existing in the 8-hour interval length, thus not changing the ASM model dimension. On the other hand, the growth of the SM model dimensions, in relation to the base instance run, is caused by the increase of batch parts allocated by the first module.

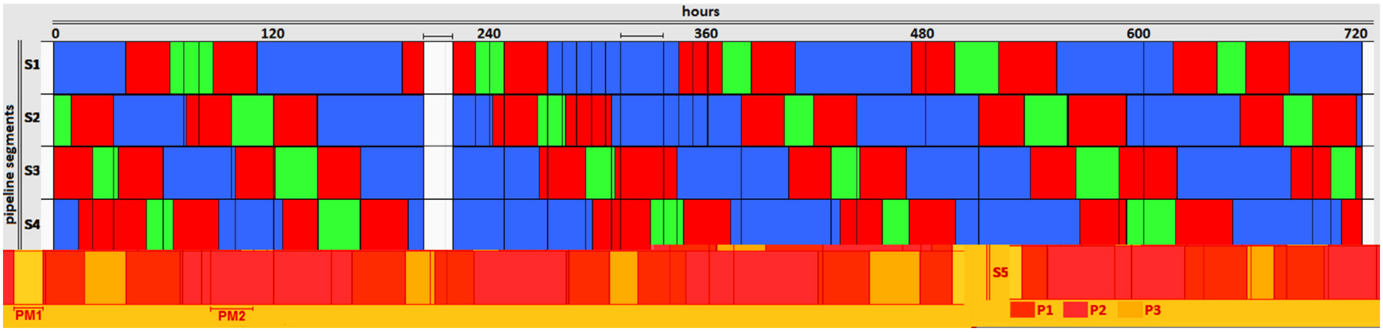


Fig. 17. CS2 - Gantt chart of the obtained solution for the extended instance. The pipeline maintenance periods (PM1 and PM2) are highlighted (example 2).

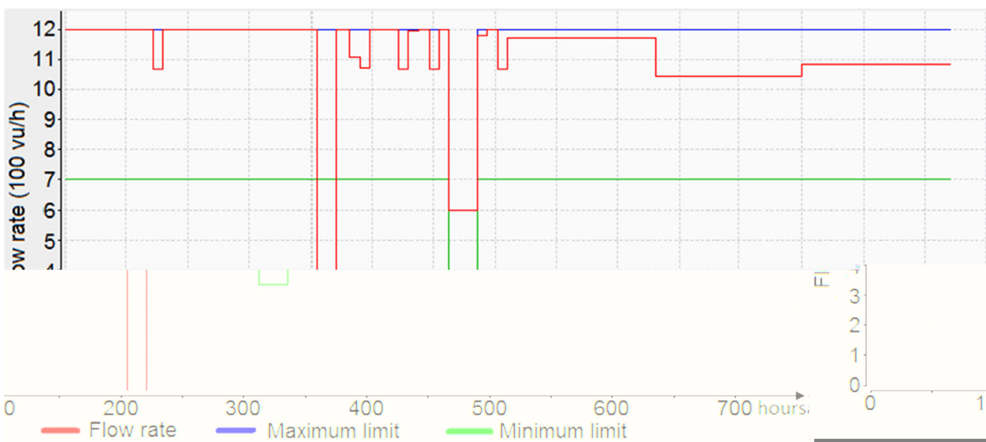


Fig. 18. CS2 - Pumping flow rate profile for the extended instance (example 2).

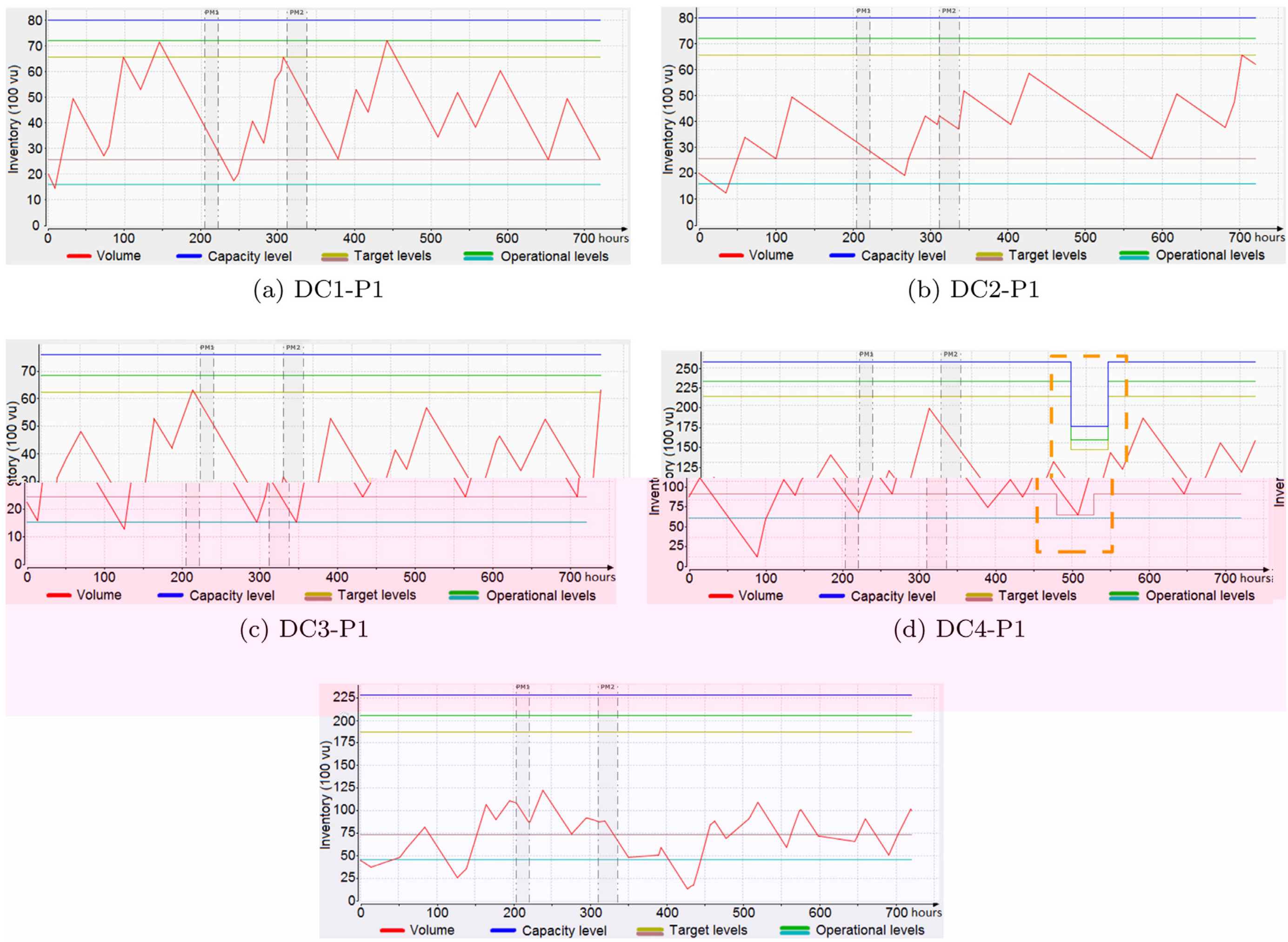
Table 3
CS2 - MILP models statistics for the extended instance run (example 2).

Module	Iteration	Constraints	Variables	Bin. Var.	CPU time (s)	Gap (%)
ASM	1	8,231	6,546	282	15.7	0
ASM	2	8,231	6,546	282	11.3	0
ASM	3	7,885	6,272	270	27.5	0
ASM	4	7,885	6,272	270	35.9	0
ASM	5	7,885	6,272	270	118.0	0
ASM	6	7,885	6,272	270	33.3	0
SM	1	5,573	3,844	1,076	0.6	0
SM	2	5,486	3,721	1,086	1.0	0
SM	3	8,959	5,613	2,011	6.2	0
SM	4	6,465	4,306	1,325	5.7	0

Table 4
CS2 - CPU time of each model execution for 8–3 hours of interval size (example 3).

Module	Iteration	Interval duration (hour)					
		8	7	6	5	4	3
ASM	1	*15.7 (0)	54.5 (0)	38.6 (0)	66.8 (0)	126.6 (0)	417.6 (0)
ASM	2	11.3 (0)	31.2 (0)	24.4 (0)	53.9 (0)	124.5 (0)	462.9 (0)
ASM	3	27.5 (0)	52.1 (0)	41.3 (0)	155.7 (0)	97.5 (0)	923.3 (0)
ASM	4	35.9 (0)	151.2 (0)	44.0 (0)	73.7 (0)	247.0 (0)	2329.5 (0)
ASM	5	118.0 (0)	129.5 (0)	94.0 (0)	471.8 (0)	539.2 (0)	5112.9 (0)
ASM	6	33.3 (0)	72.9 (0)	510.9 (0)	586.2 (0)	1249.4 (0)	3846.7 (0)
SM	1	0.6 (0)	0.7 (0)	0.6 (0)	1.0 (0)	0.7 (0)	0.7 (0)
SM	2	1.0 (0)	1.2 (0)	0.6 (0)	0.8 (0)	0.6 (0)	0.9 (0)
SM	3	6.2 (0)	4.8 (0)	1.5 (0)	2.2 (0)	3.1 (0)	2.9 (0)
SM	4	5.7 (0)	4.4 (0)	2.5 (0)	1.0 (0)	2.8 (0)	5.6 (0)
Total CPU time		**255.2**	**502.5**	**758.4**	**1413.1**	**2391.4**	**13103.0**

*CPU time in seconds (integrality gap in %).



(a) DC1-P1

(b) DC2-P1

(c) DC3-P1

(d) DC4-P1

(e) DC5-P1

Fig. 19. CS2 - Inventory profile of product P1 in all DCs (DC1–DC5) for the entire time horizon of the extended instance (example 2). Where the *X*-axis is the time horizon in hours (from 0 hour to 720 hour) and *Y*-axis is the volume in 100 vu.

Pipeline maintenance periods also contribute to pumping flow rate variations.

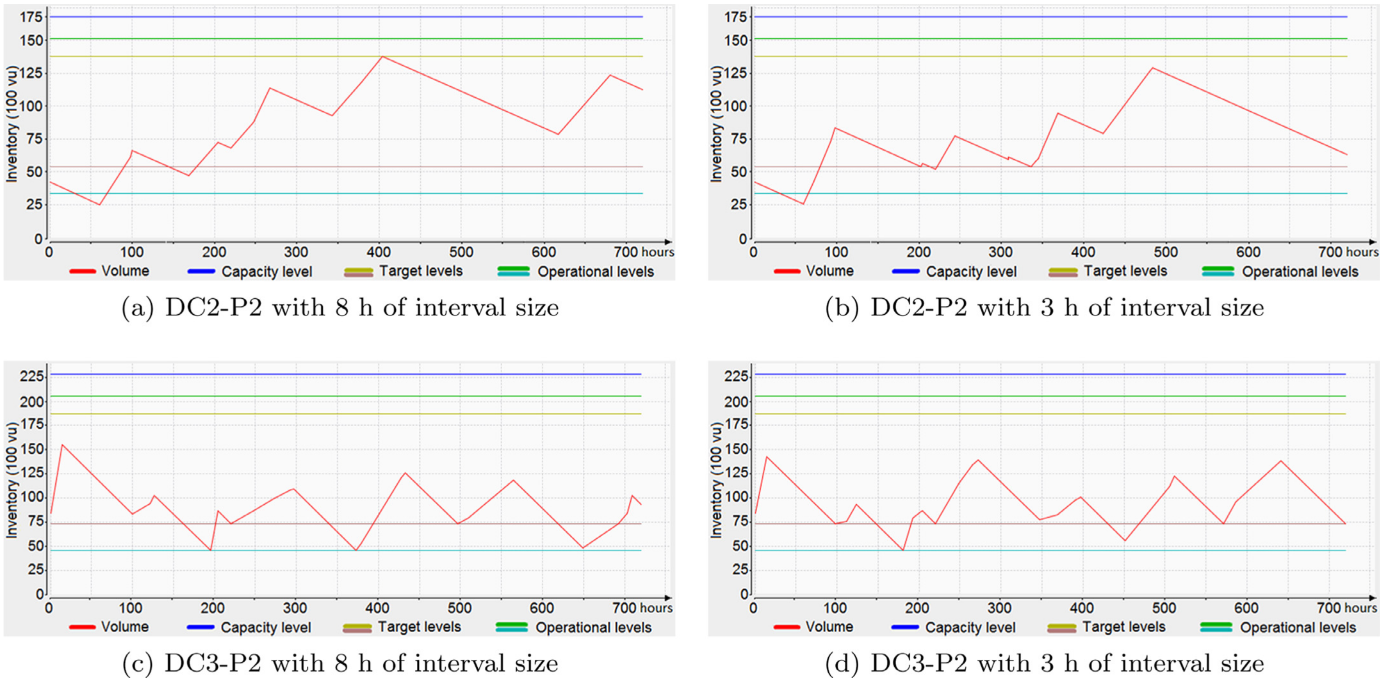
The obtained solution for the extended instance scheduled 21 batches to be pumped during the time horizon. The Gantt chart of this solution is shown in Fig. 17. It is important to notice that the entire network stops during the total pipeline maintenance, occurring from 204 hour to 220 hour. By Fig. 18 it is possible to observe the variations on pumping flow rate during the pipeline maintenance periods (total and partial). In case of pipeline partial maintenance, for the pumping system consistency, it is not just the maximum flow rate that decreases during maintenance periods, but also proportionally the same factor is applied to the minimum flow rate limit during these periods. This procedure prevents the maximum flow rate being lower than the minimum.

A consequence of a pipeline maintenance to the network is the tendency of increasing both before the maintenance period: the pumping flow rate and the inventory levels. This fact resembles a predictive control behavior, since the receiving will be affected by the maintenance. During a pipeline maintenance, the demand consumption does not stop, so the inventory levels are still decreasing with the network operating at a lower flow rate or stopped during the maintenance period. Therefore, the approach increases the inventory levels before maintenance periods to anticipate supplying problems. On the other hand, during a tank maintenance, the inventory level of the affected aggregate-tank of the DC-product needs to stay low enough, to avoid a surplus when the capacity level reduces (one of the tanks of the aggregate-tank becomes unavailable for the maintenance period).

Fig. 19(a–e) shows the inventory profiles for the obtained extended instance solution (product P1 for DC1 to DC5) in order to illustrate the differences obtained by the approach to adapt the solution to the three included maintenance periods. The inventory of product P1 in DC4 (Fig. 19(d)) presents a profile where the tank maintenance took place. The inventory levels alter during the tank maintenance, however, the inventory curve still avoided surpluses and shortages (even in target levels) during the maintenance period. The approach maintains the inventory level lower than the first execution (Fig. 15(d)) to perform the operation avoiding violations. Also, the increase in inventory before a pipeline maintenance may be noticed in Fig. 19 mainly before the full pipeline maintenance period (first shadowed area).

For the extended instance, the obtained solution also has no shortage/surplus of physical empty/capacity levels during the complete time horizon and no physical flow rate violation in any pipeline segment.

Please cite this article as: W.H.T. Meira et al., A matheuristic decomposition approach for the scheduling of a single-source and multiple destinations pipeline system, European Journal of Operational Research (2018), https://doi.org/10.1016/j.ejor.2018.01.032



(a) DC2-P2 with 8 h of interval size
(b) DC2-P2 with 3 h of interval size
(c) DC3-P2 with 8 h of interval size
(d) DC3-P2 with 3 h of interval size

Fig. 20. CS2 - Inventory profile comparison for 8 hour and 3 hour of uniform interval size in ASM (example 3).

This second example shows that the proposed decomposition approach is capable of solving complex instances with a considerable small variation of CPU time when compared to the execution without maintenance periods (first 346.7 seconds and second 353.3 seconds). The inventory profiles for both examples (Figs. 15 and 19) demonstrate the ability of the developed approach to control the volume in between the inventory levels. In addition, the Gantt charts (Figs. 14 and 17) and the pumping flow rate profiles (Figs. 16 and 18) complement the proposed approach validation and its capability of solving a real-world pipeline network scheduling problem.

6.2.3. Example 3: model performance tests

Model performance tests were developed using the extended instance. The uniform interval size of ASM was decreased progressively, in order to analyze the impact of this configuration parameter on the performance and on the quality of the obtained solution. A smaller interval size implies a larger number of events and intervals to be considered by the ASM MILP model and, consequently, increasing its dimension. In relation to the quality of the solution, more intervals mean an increase of the “observation capacity” of the model, which, consequently, tends to reduce operational problems, such as inventory violations.

The configuration parameter interval size was decreased from the initial 8 hours until 3 hours. Table 4 presents the CPU time and the final integrality gap of each model executed using 8, 7, 6, 5, 4 and 3 hours of uniform interval size. The results emphasize that the CPU time significantly increases as the interval size decreases. For instance, when the interval size is decreased from 4 hours to 3 hours, the total execution time increased more than 5 times. However, a better inventory control is obtained, as observed, for example, in the two side-by-side comparisons in Fig. 20. In the first comparison, the inventory profile of DC2-P2 obtained with 3-hour interval size (b) was able to recover fast from the first operational and target minimum violations and almost avoided the second meta minimum violation that occurred for a longer period in the 8 hour-interval solution (a). The second comparison presents the inventory profile of DC3-P2, where the 8-hour interval solution (c) could not avoid the last target minimum violation around 650 hour, whereas the 3-hour interval (d) obtained a better inventory control and avoided the third target minimum violation.

7. Conclusion

In this paper, a matheuristic decomposition approach has been proposed for the long-term schedule of a multiproduct pipeline network with a single-source and multiple destinations. As final result, the approach obtains the sequence of pumping operations and delivery operations to be performed during the considered time horizon, attending an established demand, while respecting the operational constraints of the system. Several aspects of the studied network were considered, such as forbidden sequences, simultaneous deliveries, rigorous inventory management, total pipeline maintenance, partial pipeline maintenance, tank maintenance, and flow rate variation control. To the best knowledge of authors, the last three aspects were not solved in a detailed manner by any former literature approach, given the context of a single-source with multiple destinations network.

The developed approach involves two main modules. The first module is responsible for allocating and sequencing the batches to be pumped. This module also applies a temporal decomposition using a rolling horizon process in order to deal with the long-term horizon. The second module details the delivery operations of each DC. In the last module, a structural decomposition is considered, solving the scheduling segment by segment.

In order to validate the capabilities of the proposed approach, tests on two case studies were performed. In case study 1, we solved several instances of an illustrative network with different numbers of products and with maintenance periods. For case study 2, a real-world network with a real instance from a Brazilian oil company is solved. Three examples were performed for the second case study: (1) base instance; (2) extended instance where a tank maintenance and a total and a partial pipeline maintenance peri-

Table A.1
Cost factors adopted for the case studies 1 and 2.

Cost factor	Value
Cint	10,000
Cid^{emp}	5,000
Cid^{cap}	4,000
Cid^{min}	500
Cid^{max}	400
Cid^{mintg}	50
Cid^{maxtg}	40
Cfr^{lmean}	200
Cfr^{umean}	200
Cfr^{min}	10,000
Cfr^{max}	1,000,000
Cfr^{ldiff}	550
Cfr^{udiff}	550

Table A.2
CS1 - Initialization batches.

Code	Product	Segment	Partial Vol. (vu)	Total Vol. (vu)
B2	P1	S1	10,000	10,000
B1	P2	S2	5,000	15,000
		S3	10,000	

Table A.3
CS1 - Minimum and maximum batch size.

Product	Min. Vol. (vu)	Max. Vol. (vu)
P1	7,000	25,000
P2	6,000	20,000
P3	3,000	10,000
P4	1,000	8,000
P5	1,000	8,000

ods were considered; and (3) model performance tests decreasing the interval size parameter. The purpose of the tests was to analyze the capabilities of the developed approach in solving different instances, as well as the performance and quality of the solutions when different parameters and aspects are considered. Feasible solutions were obtained for each execution in a reasonable time, attending the operational constraints of the system.

Although the presented approach is quite detailed and is able to deal with real systems, further improvements can still be devised. In particular the proposed approach can be improved by including other operational aspects of real networks such as: the use of plug products to allow forbidden sequences; settling period for quality control; and re-pumping operation to add flexibility to the system.

Acknowledgment

The authors acknowledge the Erasmus Mundus SMART[2] support (Project Reference: 552042-EM-1-2014-1-FR-ERA MUNDUS-EMA2) coordinated by CENTRALESUPLEC. The authors would also like to acknowledge financial support from the Brazilian Oil Company PETROBRAS (grant nos. 0050.0066666.11.9 and 5850.0102354.16.9), CAPES - DS and CNPq (grant nos. 305816/2014-4, 309119/2015-4, and 406507/2016-3).

Appendix A

Data Tables A.1–A.11 for the case studies 1 (CS1) and 2 (CS2).

Table A.4
CS1 - Product incompatibility matrix.

	P1	P2	P3	P4	P5
P1	–		x		
P2		–		x	
P3	x		–		
P4		x		–	x
P5				x	–

Table A.5
CS1 - DC-product demand for 21 days.

Product	DC	Dem. (vu)	Total Dem. (vu)
P1	DC1	24,500	58,100
	DC2	17,500	
	DC3	16,100	
P2	DC1	17,500	57,400
	DC2	22,400	
	DC3	17,500	
P3	DC1	12,600	25,900
	DC2	9,100	
	DC3	4,200	
P4	DC1	7,000	16,800
	DC2	6,300	
	DC3	3,500	
P5	DC1	4,900	12,600
	DC2	4,900	
	DC3	2,800	
		TOTAL	**170,800**

Table A.6
CS1 - DCs characteristics.

DC	Prod.	Delivery operation			Inventory						
		Min. Fr. (vu/hour)	Max. Fr. (vu/hour)	Min. Vol. (vu)	Initial (vu)	Cap. (vu)	Max. Op. (vu)	Max. Tg. (vu)	Min. Tg. (vu)	Min. Op. (vu)	Empty (vu)
DC1	P1	50	500	50	8,000	20,000	18,000	16,400	6,400	4,000	0
DC1	P2	50	500	50	6,000	15,000	13,500	12,300	4,800	3,000	0
DC1	P3	50	500	50	5,000	12,000	10,800	9,840	3,840	2,400	0
DC1	P4	50	500	50	4,000	8,000	7,200	6,560	2,560	1,600	0
DC1	P5	50	500	50	3,000	8,000	7,200	6,560	2,560	1,600	0
DC2	P1	10	500	50	8,000	15,000	13,500	12,300	4,800	3,000	0
DC2	P2	10	500	50	6,000	10,000	9,000	8,200	3,200	2,000	0
DC2	P3	10	500	50	5,000	10,000	9,000	8,200	3,200	2,000	0
DC2	P4	10	500	50	4,000	8,000	7,200	6,560	2,560	1,600	0
DC2	P5	10	500	50	4,000	8,000	7,200	6,560	2,560	1,600	0
DC3	P1	10	500	50	8,000	15,000	13,500	12,300	4,800	3,000	0
DC3	P2	10	500	50	6,000	12,000	10,800	9,840	3,840	2,400	0
DC3	P3	10	500	50	6,000	10,000	9,000	8,200	3,200	2,000	0
DC3	P4	10	500	50	4,000	5,000	4,500	4,100	1,600	1,000	0
DC3	P5	10	500	50	3,000	5,000	4,500	4,100	1,600	1,000	0

Please cite this article as: W.H.T. Meira et al., A matheuristic decomposition approach for the scheduling of a single-source and multiple destinations pipeline system, European Journal of Operational Research (2018), https://doi.org/10.1016/j.ejor.2018.01.032

Table A.7
CS2 - Initialization batches.

Code	Product	Segment	Partial Vol. (vu)	Total Vol. (vu)
B5	P1	S1	28,036	28,036
B4	P3	S1	11,723	16,223
		S2	4,500	
B3	P1	S2	21,379	32,926
		S3	11,547	
B2	P2	S3	13,774	67,804
		S4	54,030	
B1	P1	S4	5,646	19.385
		S5	13,739	

Table A.8
CS2 - Minimum and maximum batch size.

Product	Min. Vol. (vu)	Max. Vol. (vu)
P1	25,000	60,000
P2	45,000	135,000
P3	15,000	50,000

Table A.9
CS2 - Product incompatibility matrix.

	P1	P2	P3
P1	–		
P2		–	x
P3		x	–

Table A.10
CS2 - DC-product demand for 30 days.

Product	DC	Dem. (vu)	Total Dem. (vu)
P1	DC1	40,000	284,000
	DC2	15,000	
	DC3	34,000	
	DC4	100,000	
	DC5	95,000	
P2	DC1	50,000	375,000
	DC2	20,000	
	DC3	60,000	
	DC4	190,000	
	DC5	55,000	
P3	DC1	11,000	89,000
	DC2	5,000	
	DC3	15,000	
	DC4	45,000	
	DC5	13,000	
		TOTAL	**748,000**

Table A.11
CS2 - DCs characteristics.

DC	Prod.	Delivery operation			Inventory						
		Min. Fr. (vu/hour)	Max. Fr. (vu/hour)	Min. Vol. (vu)	Initial (vu)	Cap. (vu)	Max. Op. (vu)	Max. Tg. (vu)	Min. Tg. (vu)	Min. Op. (vu)	Empty (vu)
DC1	P1	100	600	2,000	2,000	8,000	7,200	6,560	2,560	1,600	0
DC1	P2	100	600	2,000	7,600	22,800	20,520	18,696	7,296	4,560	0
DC1	P3	100	600	2,000	6,000	20,000	18,000	16,400	6,400	4,000	0
DC2	P1	100	600	2,000	2,000	8,000	7,200	6,560	2,560	1,600	0
DC2	P2	100	600	2,000	4,200	16,800	15,120	13,776	5,376	3,360	0
DC2	P3	100	600	2,000	2,000	8,000	7,200	6,560	2,560	1,600	0
DC3	P1	100	600	2,000	2,233	7,600	6,840	6,232	2,432	1,520	0
DC3	P2	100	600	2,000	8,410	22,800	20,520	18,696	7,296	4,560	0
DC3	P3	100	600	2,000	5,000	16,000	14,400	13,120	5,120	3,200	0
DC4	P1	100	600	2,000	7,896	25,800	23,220	21,156	8,256	5,160	0
DC4	P2	100	600	2,000	20,531	60,400	54,360	49,528	19,328	12,080	0
DC4	P3	100	600	2,000	9,728	23,800	21,420	19,516	7,616	4,760	0
DC5	P1	100	450	2,000	4,513	22,800	20,520	18,696	7,296	4,560	0
DC5	P2	100	450	2,000	6,096	15,800	14,220	12,956	5,056	3,160	0
DC5	P3	100	450	2,000	5,345	15,800	14,220	12,956	5,056	3,160	0

Please cite this article as: W.H.T. Meira et al., A matheuristic decomposition approach for the scheduling of a single-source and multiple destinations pipeline system, European Journal of Operational Research (2018), https://doi.org/10.1016/j.ejor.2018.01.032

References

Boschetto, S. N., Magatão, L., Brondani, W. M., Neves-Jr, F., Arruda, L. V. R., Barbosa-Póvoa, A. P. F. D., & Relvas, S. (2010). An operational scheduling model to product distribution through a pipeline network. *Industrial and Engineering Chemistry Research, 49*(12), 5661–5682. doi:10.1021/ie900685v.

Cafaro, D. C., & Cerdá, J. (2004). Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation. *Computers and Chemical Engineering, 28*(10), 2053–2068. doi:10.1016/j.compchemeng.2004.03.010.

Cafaro, D. C., & Cerdá, J. (2008). Dynamic scheduling of multiproduct pipelines with multiple delivery due dates. *Computers and Chemical Engineering, 32*(4–5), 728–753. doi:10.1016/j.compchemeng.2007.03.002.

Cafaro, D. C., & Cerdá, J. (2009). Optimal scheduling of refined products pipelines with multiple sources. *Industrial and Engineering Chemistry Research, 48*(14), 6675–6689. doi:10.1021/ie900015b.

Cafaro, D. C., & Cerdá, J. (2011). A rigorous mathematical formulation for the scheduling of tree-structure pipeline networks. *Industrial and Engineering Chemistry Research, 50*(9), 5064–5085. doi:10.1021/ie101462k.

Cafaro, V. G., Cafaro, D. C., Méndez, C. A., & Cerdá, J. (2011). Detailed scheduling of operations in single-source refined products pipelines. *Industrial and Engineering Chemistry Research, 50*(10), 6240–6259. doi:10.1021/ie200007a.

Cafaro, V. G., Cafaro, D. C., Méndez, C. A., & Cerdá, J. (2012). Detailed scheduling of single-source pipelines with simultaneous deliveries to multiple off-take stations. *Industrial and Engineering Chemistry Research, 51*(17), 6145–6165. doi:10.1021/ie202520a.

Cafaro, V. G., Cafaro, D. C., Méndez, C. A., & Cerdá, J. (2015a). MINLP model for the detailed scheduling of refined products pipelines with flow rate dependent pumping costs. *Computers and Chemical Engineering, 72*, 210–221. doi:10.1016/j.compchemeng.2014.05.012.

Cafaro, V. G., Cafaro, D. C., Méndez, C. A., & Cerdá, J. (2015b). Optimization model for the detailed scheduling of multi-source pipelines. *Computers and Industrial Engineering, 88*, 395–409. doi:10.1016/j.cie.2015.07.022.

Ghaffari-Hadigheh, A., & Mostafaei, H. (2015). On the scheduling of real world multiproduct pipelines with simultaneous delivery. *Optimization and Engineering, 16*, 571–604. doi:10.1007/s11081-014-9263-9.

Hane, C., & Ratliff, H. (1995). Sequencing inputs to multi-commodity pipelines. *Annals of Operations Research, 57*(1), 73–101. doi:10.1007/BF02099692.

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

Kirschstein, T. (2018). Planning of multi-product pipelines by economic lot scheduling models. *European Journal of Operational Research, 264*(1), 327–339. doi:10.1016/j.ejor.2017.06.014.

Magatão, S. N. B., Magatão, L., Neves-Jr, F., & Arruda, L. V. R. (2015). Novel MILP decomposition approach for scheduling product distribution through a pipeline network. *Industrial and Engineering Chemistry Research, 54*(18), 5077–5095. doi:10.1021/ie5046796.

Magatão, S. N. B., Magatão, L., Polli, H. L., Neves-Jr, F., Arruda, L. V. R., Relvas, S., & Barbosa-Póvoa, A. P. F. D. (2012). Planning and sequencing product distribution in a real-world pipeline network: An MILP decomposition approach. *Industrial and Engineering Chemistry Research, 51*(12), 4591–4609. doi:10.1021/ie2010252.

MirHassani, S. A., & Jahromi, H. F. (2011). Scheduling multi-product tree-structure pipelines. *Computers and Chemical Engineering, 35*(1), 165–176. doi:10.1016/j.compchemeng.2010.03.018.

Mirhassani, S. A., Moradi, S., & Taghinezhad, N. (2011). Algorithm for long-term scheduling of multiproduct pipelines. *Industrial and Engineering Chemistry Research, 50*(24), 13899–13910. doi:10.1021/ie200101a.

Mostafaei, H., & Castro, P. M. (2017). Continuous-time scheduling formulation for straight pipelines. *AIChE Journal, 63*(6), 1923–1936. doi:10.1002/aic.15563.

Mostafaei, H., Castro, P. M., & Ghaffari-Hadigheh, A. (2016). Short-term scheduling of multiple source pipelines with simultaneous injections and deliveries. *Computers and Operations Research, 73*, 27–42. doi:10.1016/j.cor.2016.03.006.

Mostafaei, H., & Ghaffari-Hadigheh, A. (2014). A general modeling framework for the long-term scheduling of multiproduct pipelines with delivery constraints. *Industrial and Engineering Chemistry Research, 53*, 7029–7042. doi:10.1021/ie4038032.

Polli, H. L., Magatão, L., Magatão, S. N. B., Neves-Jr, F., & Arruda, L. V. R. (2017). Collaborative approach based on heuristic algorithm and MILP model to assignment and sequencing of oil derivative batches in pipeline networks. *Industrial and Engineering Chemistry Research, 56*(9), 2492–2514. doi:10.1021/acs.iecr.6b03516.

Rejowski, R., & Pinto, J. M. (2003). Scheduling of a multiproduct pipeline system. *Computers and Chemical Engineering, 27*, 1229–1246. doi:10.1016/S0098-1354(03)00049-8.

Rejowski, R., & Pinto, J. M. (2004). Efficient MILP formulations and valid cuts for multiproduct pipeline scheduling. *Computers and Chemical Engineering, 28*(8), 1511–1528. doi:10.1016/j.compchemeng.2003.12.001.

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*(4–5), 1042–1066. doi:10.1016/j.compchemeng.2007.06.021.

Sasikumar, M., Ravi Prakash, P., Patil, S. M., & Ramani, S. (1997). PIPES: A heuristic search model for pipeline schedule generation. *Knowledge-Based Systems, 10*(3), 169–175. doi:10.1016/S0950-7051(97)00026-9.

Zaghian, A., & Mostafaei, H. (2016). An MILP model for scheduling the operation of a refined petroleum products distribution system. *Operational Research, 16*(3), 513–542. doi:10.1007/s12351-015-0212-y.