Optimization
Iranian Journal of Optimization
Volume 11, Issue 2, 2019, 243-259
Research Paper
Online version is available on: www.ijo.iaurasht.ac.ir
Islamic Azad University
Rasht Branch
ISSN: 2588-5723
E-ISSN:2008-5427

An Improved Optimization Model for Scheduling of a Multi-Product Tree-Like Pipeline

Mehrnoosh Taherkhani*

Department of Industrial Engineering, Zanjan Branch, Islamic Azad University, Zanjan, Iran

Received: 28 January 2019
Accepted: 20 July 2019

Keywords:
Transportation
Tree-like pipeline
Scheduling
Simultaneous injections and deliveries

Abstract

In the petroleum supply chain, oil refined products are often delivered to distribution centers by pipelines since they provide the most reliable and economical mode of transportation over large distances. This paper addresses the optimal scheduling of a complex pipeline network with multiple branching lines. The main challenge is to find the optimal sequence and time of product injections/deliveries at input /output nodes in order to satisfy product demands with minimum costs. We propose a mixed integer linear problem (MILP) approach that is capable of detecting the interface volumes in any pipeline and managing the simultaneous deliveries to distribution depots. Numerical examples are solved to validate the proposed model.

* **Correspondence E_mail**: taherkhani.mehrnoosh@gmail.com

INTRODUCTION

Distribution of oil derivatives (kerosene, gasoil, gasoline, and jet fuel) is perhaps the core component of oil supply chain. Transfer operations of oil derivatives (hereafter as products) can be carried out by road, railroad, vessel and pipeline, among which the latter represents the most reliable and cost-effective way of transportation. A measure of the pipeline importance is the fact that about 70% of crude oil and its derivatives in the US are shipped by them (Cafaro & Cerdá, 2004).

Refineries can be connected to local distribution centers through different pipeline configurations, which range from straight to tree-like and mesh structures. Most pipelines transport a variety of products back to back without any physical barrier separating the in-transit products. This issue leads to a contamination volume (interface volume) between two adjacent products. The interface volume, depending on the type of adjacent products, needs to be reprocessed at a refinery and the cost concerning this process is very high. To this end, the pipeline operators should sequence products inside the pipelines with the minimum number of interfaces.

Lot sizing and scheduling problems have received a great deal of attention over the last decades. Main decisions on pipeline scheduling problem concern the optimal sequence, length and starting time of pumping operations at a refinery (aggregated schedule) and the optimal sequence, size and time of delivery operations at depots (detailed schedule) to meet product demands during a known scheduling horizon at minimum total cost. Most scheduling models for pipelines are based on discrete or continuous time MILP frameworks. In discrete time models the horizon length is divided into a number of time slots of equal and fixed durations (Rejowski & pinto, 2003; Magato et al., 2004) whereas in continuous time representations the time-slot length is selected by the optimization (Boschetto et al., 2010; Cafaro & Cerda, 2004; Relvas et al., 2006; Cafaro & Cerda, 2008; Cafaro & Cerda., 2009; Cafaro & Cerda, 2012a; Cafaro & Cerda, 2012b; Cafaro et al., 2012; Ghaffari-Hadigheh & Mostafaei, 2015; Cafaro et al., 2015; Mostafaei et al., 2016; Mostafaei & Castro, 2017; Castro & Mostafaei, 2017; Mostafaei et al., 2015; Mostafaei & Castro, 2018; Castro & Mostafaei, 2019; Taherkhani, 2018; Castro, 2017; Cafaro & Cerda, 2011; Taherkhani et al., 2017).

Aggregate scheduling models for pipelines generally have two major limitations: (1) they do not determine in which sequence to transfer products to distribution depots, and (2) they do not consider flow rate limitations in pipeline segments. In turn, detailed scheduling models overcome these limitations and reduce the energy consumed for restarting flow in inactive pipeline segments by optimally determining the sequence of product deliveries. In past few years, several papers have addressed the detailed scheduling of pipelines with different structures involving straight pipelines with a single refinery and multiple depots (Cafaro et al ., 2012; Ghaffari-Hadigheh & Mostafaei, 2015), straight pipelines with multiple refineries and depots (Cafaro et al., 2015; Mostafaei et al., 2016; Mostafaei & Castro 2017; Castro & Mostafaei, 2017) tree-like pipelines (Mostafaei et al., 2015; Mostafaei & Castro, 2018; Castro & Mostafaei, 2019; Taherkhani, 2018) and mesh structure pipelines (Castro 2017).

Cafaro and Cerda (Cafaro & Cerda, 2011) developed the first continuous time MILP model for the aggregate scheduling of a multi-level tree like pipeline, where products injected to the mainline can be branched to secondary lines and re-branched to lower level pipelines (split lines). The pipeline system connects a single refinery at the origin of the mainline to the multiple distribution centers. The problem goal is to satisfy product demand through a single due date scheduling horizon at minimum pumping, interface, idle transport capacity, and inventory carrying costs. In contrast, Taherkhani et al. (2017) introduced a MILP framework for a multi-level tree like pipeline which can be fed by multiple refineries located on the mainline. Moreover, the approach considers dual purpose nodes on the mainline, and multiple due dates for product demand at distribution centers.

Previous models on the scheduling of two-level treelike pipelines (Cafaro & Cerda, 2011; Taherkhani et al., 2017)

(Cafaro & Cerda, 2011; Taherkhani et al., 2017) neither consider simultaneous deliveries to distribution depots nor handle flow rate limitations in pipeline segments of lower diameters. In this paper, we relax these assumptions by developing a continuous time mixed integer linear programming (MILP) model. The model can be regarded as an upgrade version of the MILP approaches in (Mostafaei, 2015; Castro, 2017; Taherkhani et al., 2017). The proposed model rigorously handles flow rate limitations and optimally sequences oil products in the pipeline branches. Moreover, in contrast to the recent works on the detailed scheduling of pipeline, the proposed model rigorously considers the interface volume constraints and penalizes the interface cost in the objective function. The problem aims to find the optimal sequence of injection and delivery operations that meet product demand at minimum costs including pumping, interface, backordered demand and pump operating costs.

The rest of the paper is structured as follows. The next section briefly describes the problem under investigation and presents an MILP representation for the problem. In section 3, two case studies are solved to illustrate the proposed mathematical model. The last section provides the conclusions and some guidelines for future work.

PROBLEM STATEMENT AND OPTIMIZATION MODEL

This paper deals with the detailed scheduling of a tree-like pipeline featuring a mainline (pipeline n0), multiple branching lines (pipelines n_1, n_2,…), refineries and distribution depots (see Fig. 1). The pipeline is fed by refineries and supplies products to depots. Some nodes can both supply and receive products to/from the pipeline (known as dual purpose nodes). Throughout this paper, we use the terms secondary lines (pipelines n_1, n_2 and n_4 in Fig. 1) and split lines (pipeline n_3 in Fig. 1) to refer to the pipelines branched from the mainline and the secondary lines, respectively. We use the term "branching line" for a pipeline that branches from another. In Fig. 1, the pipeline system features 4 branching lines n_1, n_2, n_3 and n_4.

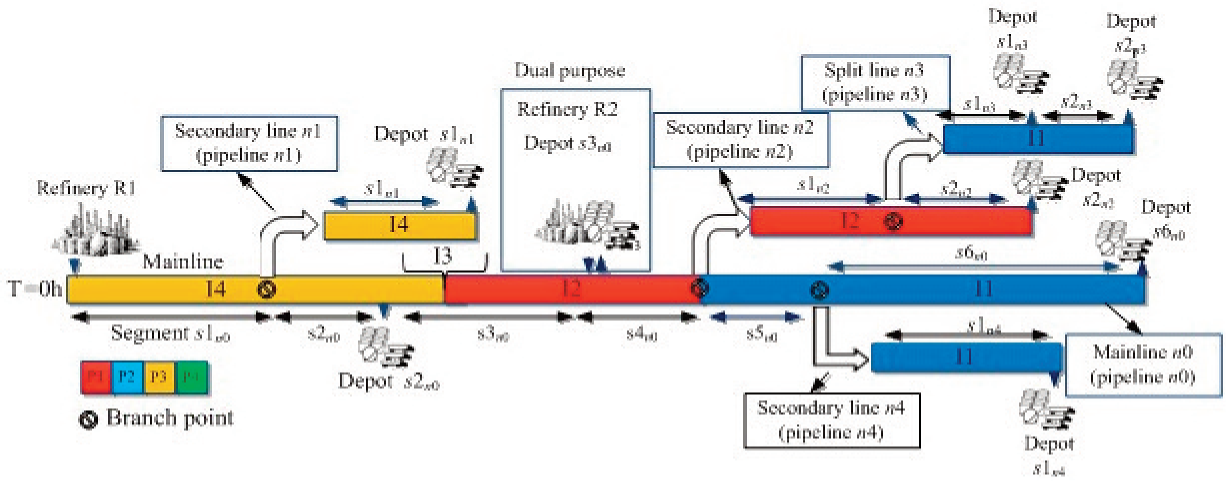

Fig. 1. A multi-product tree-like pipeline with four branching lines (Taherkhani et al., 2017).

As can be observed from Fig. 1, the pipeline system composes of a set of interconnected segments, which end with a depot or a branch point. For example, the secondary line n2 in Fig.1 contains two segments $s1_{n2}$ and $s2_{n2}$. Note that depot s_n always locates at the end of segment s_n. All pipeline segments remain full of products at any time and operate in a single flow direction when they are operative. To track the location of products inside the pipeline over time, the products are assigned to batches (lots), with each batch conveying a single product. The batches inside pipeline n at time $T=0$ h are called old batches (I_n^{old}). In Fig.1, for example, the set of mainline's old batches is I_{n0}^{old}= {I1, I2, I3, I4}. Note that the old batch I3 in the mainline is empty, which has been left by the user for a new product injection at intermediate refineries (Taherkhani et al., 2017).The batches that will be transferred to pipeline n after $T>0$ h are called new batches (I_n^{new}). Always a batch $i+1$ will follow a batch i inside pipeline n. For more information about the problem sets, refer to our previous work in (Taherkhani et al., 2017).

The problem goal is to meet product demand at distribution depots with minimum pumping, interface and pump operating costs subject to the following assumptions:

1. Input nodes are considered on the mainline and can only pump products into the mainline.
2. Flow rate in an operative segment should be kept in the given domain.
3. Pump rate in an active input node should be kept in the given range.
4. Product demands requested by depots are deterministic data and should be satisfied before the horizon end.
5. Due to quality reasons, some product sequences are forbidden.
6. Inventory of products at refinery tanks is known during planning horizon.

Motivating example

In contrast to previous works on multi-level tree-like pipelines (Castro, 2017; Cafaro & Cerda, 2011; Taherkhani et al., 2017), the proposed model in this paper allows simultaneous deliveries at depots. To illustrate the problem under study, let us consider a motivating example containing a treelike pipeline with two refineries (R1-R2), five depots (D1-D5), a secondary line and a split line. Fig. 2 shows the pipeline system at time $T=0$ h, which is filled with 4 old batches I1 (P2), I2 (P1), I3 (P2) and I4 (P3). Note that batch I3 is an empty batch and will be filled by refinery R2. The flow rate ranges at pipeline segments, product inventory at refineries and product demand at depots for the next two days (scheduling horizon length = 48 h) are given in Fig. 2. For example, R12000P3 indicates that the refinery R1 can inject 2000 m^3 of product P3 into the pipeline over next 2 days. In turn, D11000P3 denotes that depot D1 should receive 1000 m^3 of P3 during the scheduling horizon. The aim is to meet product demand at distribution depots at minimum number of pumping runs (cost of executing a pumping run = 100$).

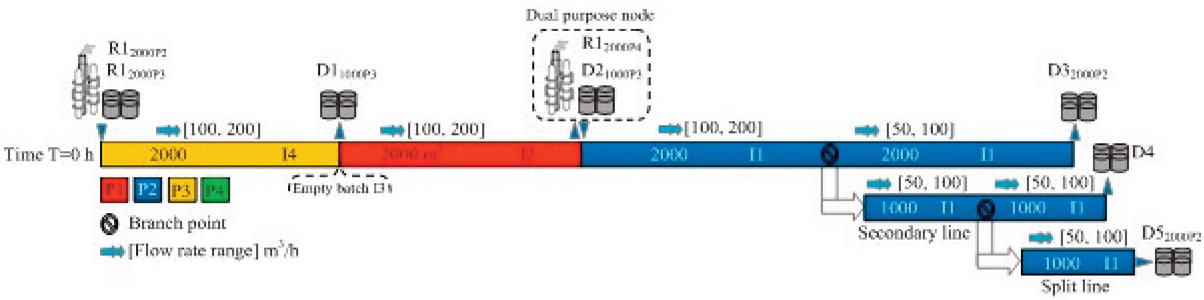

Fig. 2. Pipeline system for motivating example

Fig. 3 shows the optimal detailed schedule for the motivating example without simultaneous deliveries to depots. The optimal cost is 400 $ and the solution features 4 pumping runs *k1*, *k2*, *k3* and *k4*. The first pumping run *k1* adds 1000 m^3 of product P3 from refinery R1 into the old batch I4 and transfers 1000 m^3 of batch I1 (P2) to the secondary line to direct the same amount of I2 to split line and then to depot D5. In run *k2*, the injection of product P3 in R1 is resumed, the secondary line becomes idle and the flow goes in the last segment of mainline to transfer 1000 m^3 of P2 to depot D3. Run *k3* involves two simultaneous pumping operations at R1 and R2. Refinery R1 injects 1000 m^3 of new batch I5 (I2) and at the same time 1000 m^3 of product P3 are transferred to depot D1. Refinery R2 fills the empty batch I3 with 1000 m³ of product P4 to push 1000 m³ of batch I1 (P2) to depot D3.

The last run *k4* also contains two simultaneous injections at R1 and R2 and lasts from time 30.00 to 40.00 h. During this time interval, the following input and output operations take place: (1) refinery R1 adds 1000 m³ of product P5 to batch I5 and directs the same volume of product P3 to depot D1 and (b) refinery R2 reinserts 1000 m³ of product P4 into the batch I3 to divert 1000 m³ of product P1 (I2) to the secondary line and then to the split line. From Fig. 3, the depot requirements are satisfied within 40 h (makespan= 40 h) with four pumping runs each lasting 10 h.

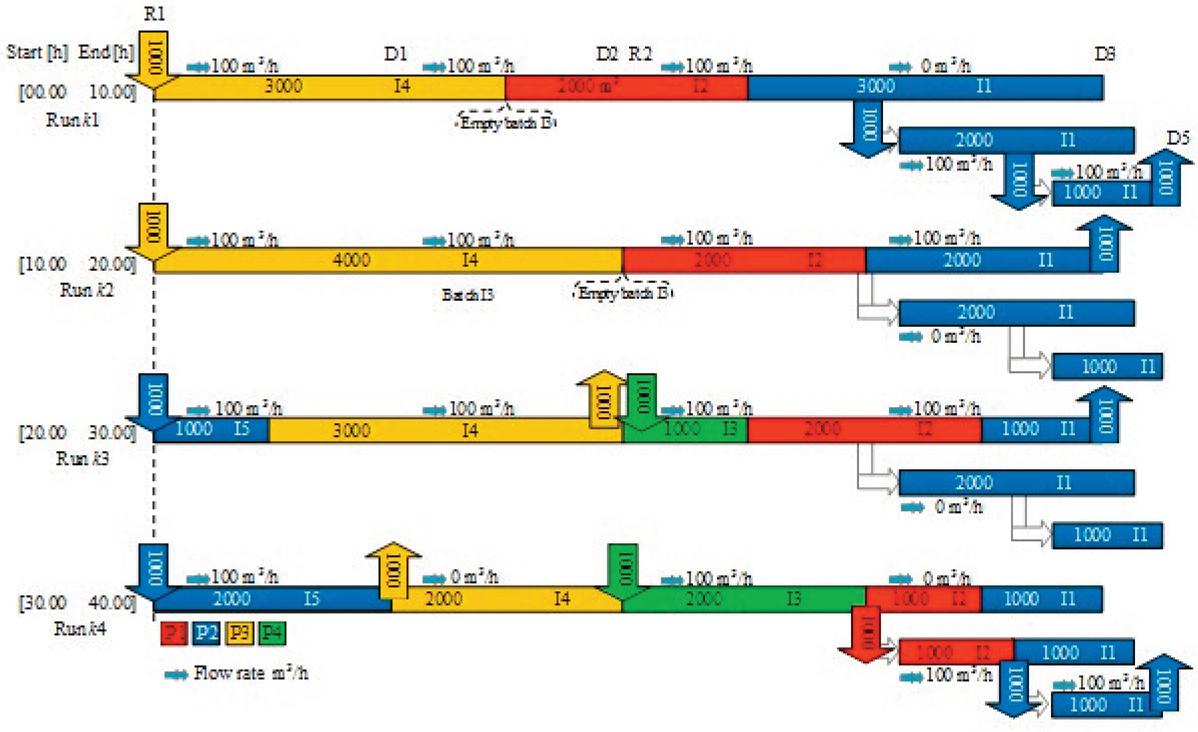

Fig. 3. Optimal schedule for the motivating example without simultaneous deliveries

The best detailed schedule with simultaneous deliveries to depots for the motivating example is depicted in Fig. 4. In the first pumping run, 2000 m³ of product P3 are added to batch I3 at a pump rate of 200 m³/h and at the same time two depots D3 and D5 start receiving product P2 at a flow rate of 100 m³/h. Pumping run k2 involves the following operations: (a) 2000 m³ of product P2 (I5) are injected from R1 to simultaneously deliver 1000 m³ of P3 to depot D1 and 1000 m³ of P3 to depot D3, and (b) empty batch I3 receives 2000 m³ of product P3 from refinery R2 and pushes 1000 m³ of batch I2 to the last segment of the mainline and the same volume of I2 to the secondary line. The total cost is 200 $ and the solution features a makespan of 20 h. This paper aims to develop a MILP model to consider simultaneous deliveries to depots while reducing pipeline operational costs.

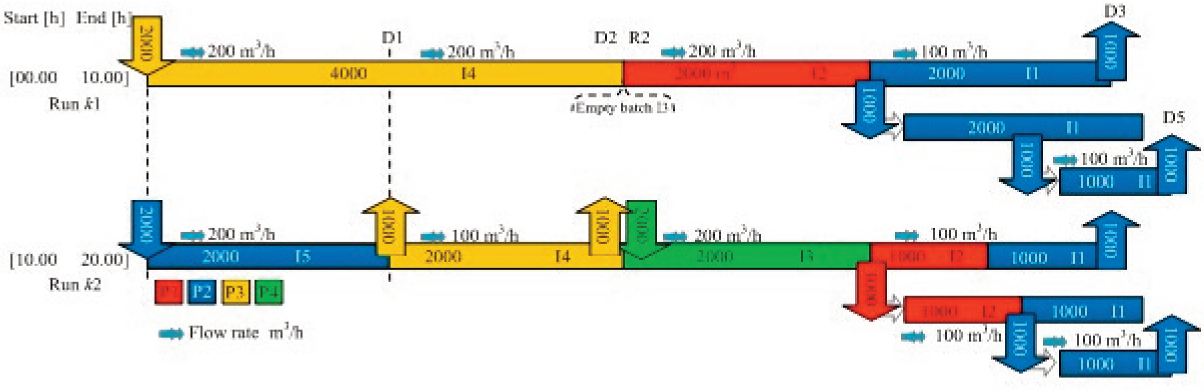

Fig. 4. Optimal schedule for the motivating example with simultaneous deliveries

Mathematical model

In this subsection, we introduce a mixed integer linear programming (MILP) model for the detailed scheduling of a tree-like pipeline with multiple branching lines, refineries and distribution depots. The model requires the following sets, parameters, and decision variables:

Sets

K	Pumping runs indexed by $k=0, 1,\dots, \|K\|$, where $k=0$ describes pipeline at the start time of scheduling horizon
N	Pipelines indexed by $n, n'=0, 1,\dots, \|N\|$
SP	Secondary lines (lines emerged from the mainline) ($SP\subset N$)
SL_n	Split lines on secondary line n ($SL_n \subset N$)
R	Refineries indexed by $r, r'= 1,\dots, \|R\|$
I	Product batches (lots) indexed by $i, i', j= 1,\dots, \|I\|$
S_n	Segments of pipeline n indexed by s, s'
I_r	Batches that can receive product from refinery r ($I_r \subset I$)
I_n^{old}	Old batches of pipeline n ($I_n^{old} \subset I$)
I_n^{new}	New batches of pipeline n ($I_n^{new} \subset I$)
I_n	Batches to be transported by pipeline n ($I_n=I_n^{old}\subset I_n^{new}$)
I_n^S	Batches that can be transferred to depot s_n ($I_n^s \subset I$)
P	Products indexed by p, p'

Parameters

h_{max}	Planning horizon (h)
stf	Starting time of the first pumping run (h)
CR	Cost of performing a pumping run ($/run)
$CP_{p,r}$	Cost of pumping a unit of product p from refinery r ($/m^3)
$CIF_{p,p'}$	Cost of reprocessing a unit of mixing volume between two products p,p'($/m3)
CB	Cost of backorder of a unite product p at depot s_n ($/m^3)
vr_r^{min}/ vr_r^{max}	Min / max pumping rate at refinery r (m^3/h)
$vs_{s,n}^{min}$/ $vs_{s,n}^{max}$	Min / max flow rate in segment s_n (m^3/h)
IPV^{min}/IPV^{max}	Min / max batch injected size in mainline in each run (m^3)
IRV_n^{min}/IRV_n^{max}	Min / max batch size transferred to branch line n in each run (m^3)
$DPV_{s,n}^{min}$/ $DPV_{s,n}^{max}$	Min / max batch size transferred to depot s_n in each run (m^3)
θ_r	Coordination of refinery r on the mainline (m^3)
σ_n	Coordination of secondary line n on the mainline (m^3)
$\delta_{n,n'}$	Coordination of split line n' one the secondary line n (m^3)
$\tau_{s,n}$	Coordinate of depot s_n on pipeline n (m^3)
$\rho_{s,n}$	Volume of pipeline n between its origin and end of segment s_n (m^3)
PV_n	Capacity of pipeline n (m^3)
$reft_{p,r}$	Volume of product p at refinery r (m^3)
$Demand_{s,n,p}$	Demand of product p at depot s_n (m^3)
$MIX_{p,p',n}$	Mixing volume between products p,p' in pipeline n (m^3)
$Touch_{p,p'} \in \{0, 1\}$	1 if products p and p' can touch each other in pipeline, otherwise 0 (m^3)
$ISPV_{i,n}$	Volume of old batch i in pipeline n (m^3)

Variables

ST_k	Start time of run k (h)
$LR_{r,k}$	Activity length of refinery r at run k (h)
L_k	Length of run k (h)
$Back_{p,s,n}$	Backordered demand of product p in depot s_n (m^3)
$IPV_{i, r, K}$	Volume of lot i pumped from refinery r at run k (m^3)
$PPV_{i, p, r, k}$	Volume of lot i of product p pumped from refinery r at run k (m^3)
$IRV_{i, k, n}$	Volume of lot $i \in I_n$ transferred to branching line n during run k (m^3)
$DPV_{i, s, k, n}$	Volume of lot $i \in I_n^s$ transferred to depot s_n at run k (m^3)
$PDPV_{i, p, s, k, n}$	Volume of lot $i \in I_n^s$ of product p transferred to depot s_n at run k (m^3)
$LPV_{i, k, n}$	Upper coordinate lot i in pipeline n at the end of run k (m^3)
$SPV_{i,k,n}$	Volume of lot i in pipeline n at the end of run k (m^3)
$INFT_{i, p, p',n0}$	Interface volume between batches i_p and $(i-1)_{p^{\wedge\prime}}$ in pipeline n (m^3)
$\lambda_{i, r,k} \in \{0,1\}$	1 if lot $i \in I_r$ is pumped at refinery r during run k
$u_{i, k, n} \in \{0, 1\}$	1 if lot i is transferred to secondary line n during run k
$ur_{i, k, n, n'} \in \{0, 1\}$	1 if lot i is transferred from secondary line n to split line n' during run k
$x_{i, s, k, n} \in \{0,1\}$	1 if lot $i \in I_n^s$ is transferred to depot s_n during run k
$y_{i, p} \in \{0, 1\}$	1 if lot i is of product p
$z_{i, n} \in \{0, 1\}$	1 if lot i exists in branching line n
$v_{s, n, k} \in \{0, 1\}$	1 if segment s_n is active in run k

The MILP formulation of the problem is as follows (Eq. 1-59).

$$\min\ z_1 = \sum_{k\in K}\sum_{n\in N}\sum_{r\in R_n}\sum_{i\in I_n^r}\sum_{p\in P_{r,n}} CP_{p,r,n}.PPV_{i,p,r,k} + \sum_{n\in N}\sum_{i\in I_n}\sum_{p\in P}\sum_{p'\in P} CIF_{p,p'}.INFT_{i,p,p',n} + \sum_{n\in N}\sum_{s\in S_n}\sum_{p\in P}\sum_{t\in T} CB.Back_{p,s,n,t} + \sum_{k\in K} CR.R_k \quad (1)$$

Subject to:

$$R_k \geq \frac{1}{|R|}\sum_{r\in R}\sum_{i\in I_r}\lambda_{i,r,k}, \quad \forall k\in K \quad (2)$$

$$LPV_{i,k,n} = \sum_{i'\geq i} SPV_{i',k,n}, \quad \forall i\in I_n, k\in K, n\in N. \quad (3)$$

$$\sum_{i\in I_r}\lambda_{i,r,k} \leq 1, \quad \forall k\geq 1, r\in R, \quad (4)$$

$$LPV_{i,k-1,n0} \geq \theta_r.\lambda_{i,r,k}, \quad \forall i\in I_r, k\in K, r\in R, \quad (5)$$

$$LPV_{i+1,k-1,n0} \leq \theta_r + (PV_{n0}-\theta_r).(1-\lambda_{i,r,k}), \quad \forall i\in I_r, k\in K, r\in R, \quad (6)$$

$$\lambda_{i,r,k}.IPV^{min} \leq IPV_{i,r,K} \leq \lambda_{i,r,k}.IPV^{max}, \quad \forall i\in I_r, k\in K, r\in R, \quad (7)$$

$$\sum_{k\in K}\sum_{r\in R}\lambda_{i,r,k} \geq 1, \quad \forall i\in I_{n0}^{new}, \quad (8)$$

$$\sum_{p\in P} y_{i,p} = 1, \quad \forall i\in I, \quad (9)$$

$$\sum_{k}\sum_{r} PPV_{i,p,r,k} \leq M.y_{i,p}, \forall i\in I, p\in P, r\in R, \quad (10)$$

$$\sum_{p} PPV_{i,p,r,k} = IPV_{i,r,k}, \quad \forall i\in I_r, r\in R, k\in K, \quad (11)$$

$$\sum_{k\in K}\sum_{i\in I_r} PPV_{i,p,r,k} \leq reft_{r,p}, \forall r\in R, p\in P, \quad (12)$$

$$\frac{\sum_{i\in I_r} IPV_{i,r,K}}{vr_r^{\max}} \leq LR_{r,k}, \quad \forall k\in K, r\in R, \quad (14)$$

$$LR_{rk} \leq L_k \leq LR_{rk} + h\left(1-\sum_{i\in I_r}\lambda_{i,r,k}\right), \quad \forall k\in K, r\in R, \quad (15)$$

$$\sum_{i\in I_s^n} x_{i,s,k,n} \leq 1, \quad \forall i\in I_s^n, k\in K, s\in S_n, n\in N, \quad (16)$$

$$LPV_{i,k-1,n} \geq \tau_{s,n}.x_{i,s,k,n}, \forall i\in I_s^n, k\in K, s\in S, n\in N, \quad (17)$$

$$LPV_{i,k,n} - SPV_{i,k,n} \leq \tau_{s,n} + (PV_n-\tau_{s,n})(1-x_{i,s,k,n}), \forall i\in I_s^n, k\in K, s\in S_n, n\in N, \quad (18)$$

$$x_{i,s,k,n}DPV^{min} \leq DPV_{i,s,k,n} \leq x_{i,s,k,n}DPV^{max}, \forall i\in I_s^n, k\in K, s\in S_n, n\in N, \quad (19)$$

$$\sum_{p\in P} PDPV_{i,p,s,k,n} = DPV_{i,s,k,n}, \quad \forall i\in I_n^s, k\in K, n\in N, s\in S_n, \quad (20)$$

$$\sum_{k\in K}\sum_{n\in N}\sum_{s\in S_n} PDPV_{i,p,s,k,n} \leq M.y_{i,p}, \quad \forall i\in I, p\in P, \quad (21)$$

$$\sum_{k\in K}\sum_{i\in I_n^s} PDPV_{i,p,s,k,n} \geq Demand_{s,n,p,t} - Back_{p,s,n}, \quad \forall p\in P, s\in S_n, n\in N, \quad (22)$$

$$LPV_{i,n0,k-1} \geq \sigma_n u_{i,n,k}, \forall i\in I_n, n\in SP, k\in K\ (k\geq 1), \quad (23)$$

$$LPV_{i+1,n0,k} \leq \sigma_n + (PV_n0-\sigma_n)(1-u_{i,n,k}), \forall i\in I_n, n\in SP, k\in K\ (k\geq 1), \quad (24)$$

$$IRV_n^{min} u_{i,n,k} \leq IRV_{i,n,k} \leq IRV_n^{max} u_{i,n,k}, \forall i\in I_n, n\in SP, k\in K\ (k\geq 1), \quad (25)$$

$$LPV_{i,k-1,n} \geq \delta_{n,n'} ur_{,k,n,n'}, \quad \forall i\in I_n, k\in K, n\in SP, n'\in SL_n, \quad (26)$$

$$LPV_{i+1,k,n} \leq \delta_{n,n'} + (PV_n - \delta_{n,n^{\wedge}})(1 - ur_{i,k,n,n'}),$$
$$\forall i \in I_n,\ k \in K,\ n \in SP,\ n' \in SL_n, \quad (27)$$

$$ur_{i,k,nn'}\, IRV_{n'}^{min} \leq IRV_{i,k,n'} \leq ur_{i,k,n'}\, IRV_{n'}^{max},$$
$$\forall i \in I_n,\ k \in K,\ n \in SP,\ n' \in SL_n, \quad (28)$$

$$SPV_{i,k,n0} = SPV_{i,k-1,n0} + \sum_r IPV_{i,r,k} + - \sum_{s \in S_{n0}} DPV_{i,s,k,n0}$$
$$- \sum_{n \in SP} IRV_{i,k,n}, \quad \forall i \in I_{n0}, k \in K, k \geq 1, \quad (29)$$

$$SPV_{i,k,n} = SPV_{i,k-1,n} + IRV_{i,k,n} - \sum_{s \in S_n} DPV_{i,s,k,n}$$
$$- \sum_{n' \in SL_n} IRV_{i,k,n'}, \quad \forall i \in I_n, k \in K, n \in SP, \quad (30)$$

$$SPV_{i,k,n} = SPV_{i,k-1,n} + IRV_{i,k,n} - \sum_{s \in S_n} DPV_{i,s,k,n},$$
$$\forall i \in I_n, k \in K, n \in N \backslash \{SP \cup n0\}, \quad (31)$$

$$\sum_{i \in I_n} SPV_{i,k,n} = PV_n, \quad \forall n \in N, k \in K, \quad (32)$$

$$\sum_{i \in I_n} IRV_{i,k,n} = \sum_{s \in SL_n} \sum_{i \in I_n} DPV_{i,s,k,n},$$
$$\forall k \in K, n \in N - \{n0 \cup SP\}, \quad (33)$$

$$\sum_{i \in I_n} IRV_{i,k,n} = \sum_{s \in S_n} \sum_{i \in I_n} DPV_{i,s,k,n} + \sum_{n' \in SL_n} \sum_{i \in I_{n'}} IRV_{i,k,n'}$$
$$\forall k \in K, n \in SP, \quad (34)$$

$$\sum_{i \in I_r} \sum_{r \in R} IPV_{i,r,k} = \sum_{s \in S_{n0}} \sum_{i \in I_{n0}} DPV_{i,s,k,n0}$$
$$+ \sum_{n \in SP} \sum_{i \in I_n} IRV_{i,k,n}, \forall k \in K, \quad (35)$$

$$INFT_{i,p,p',n0} \geq MIX_{p,p',n}(y_{i+1,p} + y_{i,p'} - 1),$$
$$\forall i \in I_{n0}, p, p', \quad (36)$$

$$INFT_{i,p,p',n} \geq MIX_{p,p',n}\left(y_{i,p} + y_{i',p'} + z_{i,n} + z_{i',n} + \sum_{j \geq i'+1}^{i-1} z_{j,n} - Touch_{p,p'} - 2\right), \forall i, i'$$
$$\in I_n (i > i'), p, p', n \neq n0, \quad (37)$$

$$z_{i,n} \leq \sum_k u_{i,k,n} \leq |K| z_{i,n},$$
$$\forall i \in I_n^{new}, n \in N, , n \neq n0, \quad (38)$$

$$z_{i,n'} \leq \sum_k ur_{i,k,n,n'} \leq |K| z_{i,n'},$$
$$\forall i \in I_n^{new}, n \in SP, n' \in SL_n, \quad (39)$$

$$y_{i,p} + y_{i+1,p'} \leq Touch_{p,p'} + 1, \quad \forall i \in I, p, p' \in P, \quad (40)$$

$$z_{i,n} + z_{i',n} \leq \sum_{j \in I_n, j \geq i'+1}^{i-1} z_{j,n} - y_{i,p} - y_{i',p'} + Touch_{p,p'}$$
$$+ 3, \ \forall i \in I_n^{new}, i' \in I_n, p, p, n \neq n0, \quad (41)$$

$$\sum_{i \in I_n^S} x_{i,s,n,k} \leq v_{s,n,k}, \quad \forall s \in S_n, n \in N, k \in K(k \geq 1), \quad (42)$$

$$\sum_{\substack{r \in R \\ \tau_{s-1,n0} \leq \theta_r < \tau_{s,n0}}} \sum_{i \in I_r} \lambda_{i,r,k} \leq v_{s,n0,k}, \forall s \in S_{n0}, k \in K(k \geq 1), \quad (43)$$

$$v_{s,n0,k} \leq \sum_{r \in R} \sum_{\substack{i \in I_r \\ \tau_{s-1,n0} \leq \theta_r < \tau_{s,n0}}} \lambda_{i,r,k} + v_{s-1,n0,k},$$
$$\forall s \in S_{n0}, s > 1, k \in K(k \geq 1), \quad (44)$$

$$v_{s,n,k} \le v_{s-1,n,k}, \forall s \in S_n, s>1, n \in N(n \ne n0), k \in K(k \ge 1), \quad (45)$$

$$v_{s,n,k} = \sum_{i \in I_n} u_{i,n,k}, \quad s = first(S_n), n \in SP, k \in K(k \ge 1), \quad (47)$$

$$\sum_{i \in I_n} u_{i,n,k} \le v_{s,n0,k}|_{\sigma_n = \rho_{s,n0}}, \forall s \in S_{n0}, n \in SL, k \in K(k \ge 1), \quad (48)$$

$$\sum_{i \in I_{n'}} ur_{i,k,n,n'} \le v_{s,n,k}|_{\delta_{n,n'} = \rho_{s,n'}}, \quad \forall s \in S_{n0}, n \in SL, k \in K(k \ge 1), \quad (49)$$

$$L_k vs_{s,n0}^{\min} - M(1 - v_{s,k,n0}) \le \sum_{s' \ge s} \sum_{i \in I_n^s} DPV_{i,s',k,n0} + \sum_{\substack{n \in SL \\ \rho_{s,n0} \le \sigma_n}} \sum_{i \in I_n} IRV_{i,k,n} - \sum_{i \in I_r} \sum_{\substack{r \in R_{n0} \\ \rho_{s,n0} \le \theta_r}} IPV_{i,r,k} \le L_k vs_{s,n0}^{\max}, \forall s \in S_{n0}, k(k \ge 1), \quad (50)$$

$$L_k vs_{s,n}^{\min} - M(1 - v_{s,k,n}) \le \sum_{s' \ge s} \sum_{i \in I_n^s} DPV_{i,s',k,n} + \sum_{\substack{n' \in SL_n \\ \rho_{s,n} \le \delta_{n,n'}}} \sum_{i \in I_n} IRV_{i,k,n'} \le L_k vs_{s,n}^{\max}, \quad \forall s \in S_n, n \in SP, k \in K(k \ge 1), \quad (51)$$

$$L_k vs_{s,n}^{\min} - M(1 - v_{s,k,n}) \le \sum_{s' \ge s} \sum_{i \in I_n^s} DPV_{i,s',k,n} \le L_k vs_{s,n}^{\max}, \quad \forall s \in S_n, n \in N - \{n0 \cup SP\}, k \in K, \quad (52)$$

$$\sum_{\substack{s \in S_{n0} \\ \tau_{s,n0} \le \theta_r}} \sum_{i \in I_{n0}^s} DPV_{i,s,k,n0} + \sum_{\substack{n \in N \\ \delta_n \le \theta_r}} \sum_{i \in I_{n0}} IRV_{i,k,n} \ge \sum_{\substack{r' \in R \\ \theta_{r'} \le \theta_r}} \sum_{i \in I_{r'}} IPV_{i,r',k} - IRV_{n0}^{\max}(1 - \sum_{i \in I_r} \lambda_{i,r,k}), \quad \forall k \in K, r \in R, \quad (53)$$

$$\sum_{\substack{s \in S_{n0} \\ \delta_n \ge \tau_{s,n0}}} DPV_{i,s,k,n0} + \sum_{n' \in SP}^{n} IRV_{i,k,n} \le \delta_n - (LPV_{i,k-1,n0} - SPV_{i,k-1,n0}) + \sum_{\theta_r \le \delta_n}^{r} IPV_{i,r,k} + PV_{n0}(1 - u_{i,k,n}), \quad \forall i \in I_{n0}, k, n \in SP, \quad (54)$$

$$\sum_{s' \in S_{n0}}^{s} DPV_{i,s',k,n0} + \sum_{\substack{n \in SP \\ \delta_n \le \tau_{s,n0}}} IRV_{i,k,n} \le \tau_{s,n0} - (LPV_{i,k-1,n0} - SPV_{i,k-1,n0}) + \sum_{\theta_r \le \tau_{s,n0}}^{r} IPV_{i,r,K} + PV_{n0}(1 - x_{i,s,k,n0}), \forall i \in I_{n0}, k, s \in S_{n0}, \quad (55)$$

$$\sum_{\substack{s \in S_n \\ \delta_{n,n'} \ge \tau_{s,n}}} DPV_{i,s,k,n} + \sum_{n'' \in SP_n}^{n'} IRV_{i,k,n''} \le \delta_{n,n'} - (LPV_{i,k-1,n} - SPV_{i,k-1,n}) + IRV_{i,k,n} + PV_n(1 - ur_{i,k,n,n'}), \forall i \in I_n, k \in K, n \in SP, n' \in SL_n, \quad (56)$$

$$\sum_{\substack{s' \in S_n \\ s' \le s}} DPV_{i,s,k,n} + \sum_{\substack{n' \in SP_n \\ \delta_{n,n'} \le \tau_{s,n}}} IRV_{i,k,n'} \le \tau_{s,n} - (LPV_{i,k-1,n} - SPV_{i,k-1,n}) + IRV_{i,k,n} + PV_n(1 - x_{i,s,k,n}), \forall i \in I_n, s \in S_n, k \in K, n \in SP, n' \in SL_n, \quad (57)$$

$$ST_k = L_{k-1} + ST_{k-1}, \quad \forall k \in K(k \ge 2), \quad (58)$$

$$ST_k = stf, \quad k = 1. \quad (59)$$

In the model presented above, Eq.1 is the objective function which minimizes product pumping cost, interface cost between two products, backordered demand cost and on/off pump operating cost. Eq.2 states that the pipeline network receives material if at least one of the refineries is injecting material. Eq.3 is to detect the upper coordinate of lot i inside pipeline n at the end of run k. In fact, the upper coordinate of batch i at run k is the size of lots i' (i'>i). In Fig. 2, the upper coordinate of batch I2 in the mainline is 4000 m^3. The upper coordinates of batch I1 are

8000, 4000 and 2000 m^3 in the mainline, secondary line and split line, respectively.

Eq.4-12 are required to manage lot injections at refineries. Eq.4 states that at run k, at most one lot can be pumped from refinery r. A batch i in the mainline can receive material from the refinery r when its coordinate variables satisfy $LPV_{i+1, n0, k-1} \leq \theta_r \leq LPV_{i, n0, k-1}$, see Eq.5 and Eq.6. Eq.7 states that the injection volume should be in the admissible range. Eq.8 states that all new batches should be pumped during the planning horizon. Each lot transports only a single product, as imposed by Eq.9. Due to Eq.10 and Eq.11, the volume p injected to batch i from refinery r at run k is zero if the batch does not convey product p ($y_{i,p}=0$) and is equal to $IPV_{i, r, k}$ if the batch is injected. Eq.12 states that the volume p injected from refinery r cannot exceed its stock reft $_{r, p}$.

The activity length of refinery r during run k ($LR_{r, k}$) is computed through Eq.13 and Eq.14. Eq.15 states that the activity length of all active refineries is the same and determines the duration of pumping run (L_k). Eq.16-22 are needed to control the product deliveries to distribution depots. Eq.16 states that at most on batch can be delivered to a distribution depot during any run k. A lot i in pipeline n can be transferred to depot s if its coordinate variables satisfy $LPV_{i+1, n, k} \leq \tau_{s, n} \leq LPV_{i, n, k-1}$, as imposed by Eq.17 and Eq.18. Eq.19 states that the delivery volume to an active depot should be in the feasible domain. By Eq.20 and Eq.21, the volume p transferred from batch i to depot s_n at run k will be zero if the batch is not of product p ($y_{i,p}=0$) and will be $DPV_{i, s, k, n)}$ if the batch is delivered to depot s_n. Demand should be fully met during the planning horizon; otherwise backordered demand cause penalty cost, as imposed by Eq.22.

Eq.23-25 aim to transfer a lot i in the mainline to secondary line n. When a batch $i \in I_n0$ is transferred to the secondary line n, its coordinate variables satisfy $LPV_{i+1, n0, k} \leq \sigma_n \leq LPV_{i, n0, k-1}$, where σ_n is the volumetric coordinate of secondary line n on the mainline, Eq.23 and Eq.24. A positive volume of batch $i \in I_n0$ belonging to the interval $[IRV_n^{min}, IRV_n^{max}]$ will be transferred from mainline to the secondary line n, if the secondary line is receiving material from batch i through run k, Eq.25. The similar constraints can be given for a batch transferred from a secondary line n to its split line n', i.e., Eq.26-27

Eq.29-31 computes the size of a batch i in pipeline n during a run k. For instance, Eq.29 states that the size of batch i in the mainline increases when it receives product from a refinery and decreases when the batch i is transferred to depots and secondary lines. Each pipeline n remain full with product at the end of any pumping run k (Eq.30) and consequently when some material is input in the pipeline n, the same volume is discharged from that line, Eq.31-33.

As stated before, there is no separating device between two products inside the pipeline and consequently there will be a mixing volume (interface) between them. Eq.36-37 are to detect the interface volume between two products in the mainline and branching lines. Lot i will exist in branching line n if it is transferred through a pumping run k to that line, Eq.38 and Eq.39. Some products cannot touch each other inside the pipeline (due to the huge interface). If product p allows to touch p' then the binary parameter $Touch_{p,p'}$ is one. Eq.39-37 are to prevent products p and p' to touch each other inside a pipeline n if $Touch_{p,p'}=0$.

Eq.42-49 are to determine whether a segment s of pipeline n (segment s_n) is active during run k. If depot s_n is active and consequently segment s_n will be active, Eq.42. Eq.43 states that a segment s of the mainline will be active when refinery r at the start of that segment pumps into the mainline. Through Eq.44, segment s-1 of mainline will be active if segment s is active and there is no injection at refinery r located at the start of segment s. Since there is no refinery on branching lines, segment s of branching line n will be idle if segment s-1 is idle. Due to Eq.46 and Eq.47, the first segment of a branch line will be active when it is receiving material from another pipeline. Eq.48 states that segment s of mainline featuring secondary line n is active if the secondary line is receiving material from the mainline. Eq.49 expresses that segment s of secondary line n having split line n' will be active if the split line is receiving material from the secondary line.

Eq.50-52 are introduced to control the flow rate restrictions in segments of mainline, secondary lines and split lines, respectively. Eq.53 states that no batch can input from segment $(s\text{-}1)_n0$ to s_n0 when refinery r at the start of segment s_n0 pumps into the mainline. Through run k, a lot i in the mainline may be diverted to several secondary lines and distribution depots on the mainline. If so, the volume transferred from the lot $i \in I_n0$ to mainline's secondary lines and depots is limited through Eq.54 and Eq.55. In turn, Eq.56 and Eq.57 are to manage the volume moved from a lot i in secondary line n to its depots and its split lines. Eq.58 states that the start time of pumping run k is equal to the start time of pumping run k-1 plus the length of pumping run k-1. The start time of the first pumping run is a known parameter (stf), see Eq.59.

RESULTS

In this section, we consider two case studies. The examples are solved with non-simultaneous and simultaneous delivery modes. In non-simultaneous mode (NSIM), there is no simultaneous product delivery to distribution centers (by adding $\sum_{i,s,n} x_{i,s,n,k} = \sum_{i,r} \lambda_{i,r,k}$ to the model in Section 2, we call this equation as Eq. (60)) while in simultaneous mode (SIM) multiple depots can receive product from the pipeline during a pumping run. The model for the SIM mode consists of Eqs. (1-59) and for the NSIM mode consists of Eqs. (1)- (60). The MILP models are implemented in GAMS /CPLEX and solved on an Intel (R) core(TM) i5 CPU M430 2.27 GHZ with 4 GB (3.86 GB usable) and 64 bit operation system. We adopt as stopping criterion a maximum computational time of 18,000 CPUs or a relative optimality tolerance of 10-9.

Example 1

The pipeline structure for this example is depicted in Fig. 5, which is similar to the one considered by Taherkhani et al. (Taherkhani et al 2017). The pipeline system should transport four products from two refineries (R1-R2) to five depots (D1-D5) over next 240 h. At time t=ST=0 h, there are three old batches I1 (P2), I2 (P1) and I3 (P3), with batch I1 re-branching into the pipelines N1 and N2. The flowrate in pipeline segments can vary between 20 and 200 m^3/h.

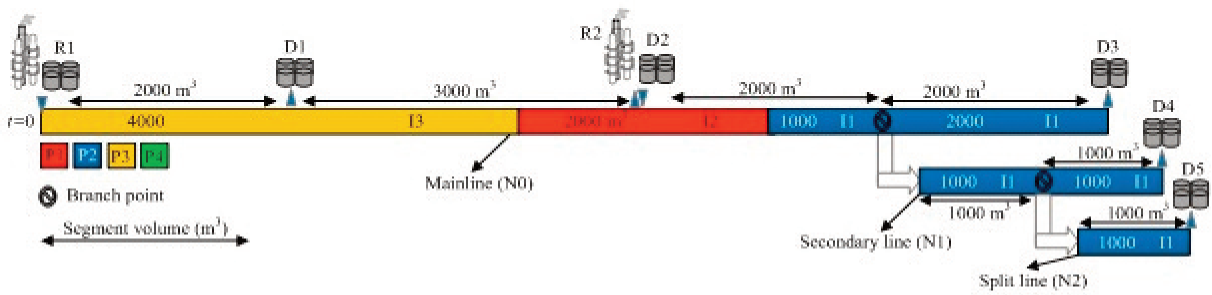

Fig. 5. Pipeline system for Example 1.

The injection rate domain (given in m^3/h) at refineries is [20, 200]. The maximum and minimum volumes transferred to mainline and branching lines during each run are 4000 and 500 m^3, respectively. The same upper and lower bounds are given for min/max volumes transferred from a pipeline to an active distribution center. Other data of the example including inventory and pumping cost at refineries, demand and interface volume (vol.) and cost can be found in Table 1. To solve the problem we consider I_n0^{new}={I4, I5}, CB=200 \$/ m^3 and CR=100 \$/run.

Table 1: Data for Example 1

P	Inventory ($10m^3$)		Pump cost ($\$/m^3$)		Demand ($10m^3$)					Interface vol. (m^3)/cost($)			
	R1	R2	R1	R2	D1	D2	D3	D4	D5	P1	P2	P3	P4
P1	1000	2000	3	4	0	0	180	200	100	0	1/15	1/30	1/26
P2	-	1000	-	3	0	0	300	200	100	1/15	0	1/20	1/18
P3	1200	-	1	-	500	100	0	100	0	1/30	1/20	0	1/30
P4	700	600	2	1	200	0	300	0	100	1/25	1/18	1/30	0

Similar to previous works in [17,23], we adopt an iterative procedure on the elements of set K (starting with $k = \left\lceil \sum_{p,s,n} \frac{demand_{p,s,n}}{IPV^{max}} \right\rceil$), to deter mine the optimal solution. When the solution quality for two consecutive elements k and k+1 is the same, k is the optimal number of pumping runs. Table 2 shows the computational results for Example 1. We start with $k = \left\lceil \sum_{p,s,n} \frac{demand_{p,s,n}}{IPV^{max}} \right\rceil = \left\lceil \frac{23800}{4000} \right\rceil$ =6 (i.e.,K_6={0,1,2,…6}) to solve the NSIM and SIM models. With six pumping runs, SIM faces 4.2% backordered demands while NSIM encounters 33.61% product shortages at depots.

With K_7={0,1,2,…7} product demands are met in SIM and the solution is found at just 16 s, whereas NSIM again faces backordered demands (25.21%) and needs 557s to generate a solution of \$1220825. For K_8={0,1,2,…8} and K_9={0,1,2,…9}, SIM generates the same solution of \$38505 and so the optimal number of pumping runs for the SIM model is 8. With 9 pumping runs, SIM just spends 183s to confirm the solution obtained using 8 pumping runs, while NSIM needs a CPU time of 4387s to get the solution of \$824525. From Table 2, one can see that NSIM cannot meet demand even with 11 pumping runs up to the maximum computational time of 18000 CPUs. Note that for the same number of pumping runs, both SIM and NSIM models are roughly similar in terms of problem size and generate the same linear relaxed solution, but SIM reaches the optimum in lower CPU times due to the lower integrality gaps.

Table 2: Computational results of Example 1

	N.R[a]	CPU(s)	Cont. v[b]	Bin. v[c]	Eqs.	Bd (%)[d]	P.c ($)[e]	I.c ($)[f]	z ($)	Relax z ($)	In.gap[i]
SIM[h]	6	17	1400	352	2892	4.2	39500	221	240321	29687.5	0.876
SIM	7	16	1556	404	3276	0	40500	221	41421	29682.187	0.283
SIM	8	108	1712	456	3660	0	37500	221	38521	29678.055	0.229
SIM	9	183	1868	508	4044	0	37500	221	38521	29674.75	0.229
NSIM[g]	6	79	1400	352	2898	33.61	16000	125	1616725	29687.5	0.981
NSIM	7	557	1556	404	3283	25.21	20000	125	1220825	29682.187	0.975
NSIM	8	903	1712	456	3668	21	22500	125	1023425	29678.055	0.971
NSIM	9	4387	1868	508	4053	16.8	23500	125	824525	29674.75	0.964
NSIM	10	6220	2024	560	4438	12.6	25000	125	626185	29672.045	0.952
NSIM	11	18000	2180	612	4823	8.4	32500	180	433780	29669.791	0.931

[a]Number of pumping runs; [b]Continuous variables; [c]Binary variables; [d]Backordered demand; [e]Pumping cost; [f]Interface cost; [g]Non-simultaneous delivery mode; [h]Simultaneous delivery mode; [i]Integrality gap=(z-Relax z)/z

The optimal detailed schedule for Example 1 in simultaneous mode (SIM) with 8 pumping runs is given in Table 3. From the first row of this table, for instance, 2000 m^3 of product P4 are injected from refinery R1 during time interval [0.00-25.00] h and 1500 m^3 of P3 and 500 m^3 of P2 are transferred to depots D1 and D3, respectively. From this table, product demands are met within time interval [00.00-210.00] h.

Table 3: Amount of products transferred to pipelines and depots of Example 1 (in 10 m^3).

Time Interval [h]	Output nodes		Branchline		Output nodes				
	R1	R2	N1	N2	D1	D2	D3	D4	D5
00.00-25.00	200_{P4}	-	-	-	150_{P3}	-	50_{P2}	-	-
25.00-35.00	200_{P4}	-	-	-	150_{P4}	-	50_{P2}	-	-
35.00-60.00	100_{P3}	300_{P1}	100_{P1}	-	50_{P4}	50_{P3}	200_{P2}	100_{P2}	-
60.00-110.00	200_{P3}	-	200_{P1}	100_{P1}	-	-	-	100_{P2}	100_{P2}
110.00-157.50	200_{P3}	-	200_{P3}	-	-	-	-	200_{P1}	-
157.50-175.00	350_{P3}	300_{P4}	100_{P4}	-	300_{P3}	50_{P3}	200_{P1}	100_{P3}	-
175.00-190.00	100_{P3}	300_{P4}	100_{P4}	100_{P4}	50_{P3}	50_{P3}	200_{P4}	-	100_{P1}
190.00-210.00	200_{P3}	-	100_{P4}	100_{P4}	-	-	100_{P4}	-	100_{P4}

Example 2

This example concerns a large scale pipeline distribution system (Taherkhani et al 2017), in which a pipeline with three branching lines should convey four products from two refineries (R1-R2) to seven distribution depots (D1-D7) over 10-day, see Fig 6. At time t=0, there are four old batches I1 (P1), I2 (P2), I3 (P3) and I4 (P1).The pump rate at refineries should be kept between 300 and 800 m^3/h. Products P1 and P2 are forbidden to touch together inside the pipeline.

The flow rate ranges in pipeline segments are given in Fig 6. The maximum and minimum volumes transferred to each pipeline during each run are 16000 and 500 m^3, respectively. The same upper and lower bounds are for min/max volumes transferred from a pipeline to an active distribution center. Product inventory, pumping cost at refineries and demand for next 10 days are given in Table 4. The interface volume and cost can be found in [23]. We also consider I_n0^{new}={I5, I6}, CB=200 \$/m^3 and CR=0 \$/run.

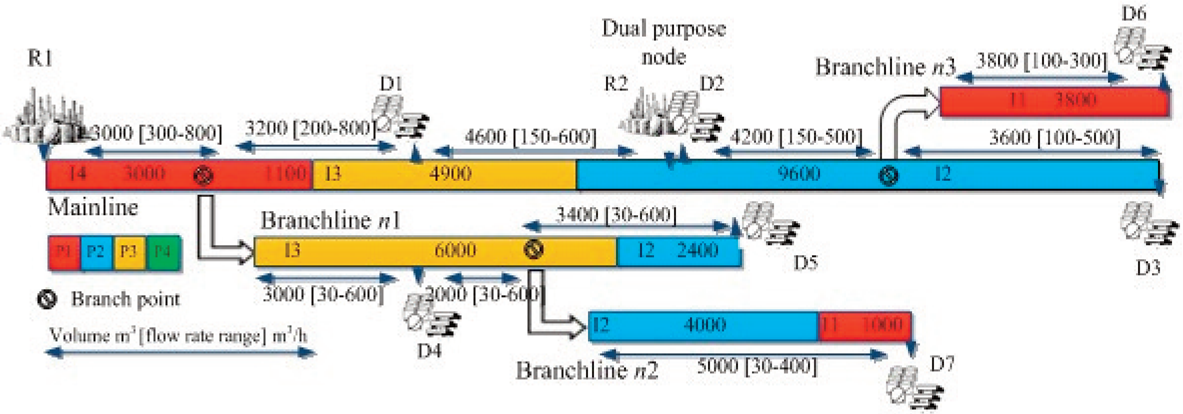

Fig. 6.Pipeline structure of Example 2

Table 4: Inventory, pumping cost and demand of Example 2

P	Inventory ($10m^3$)		Pump cost ($/m^3$)		Initial inventory of depots ($10m^3$)						
	R1	R2	R1	R2	D1	D2	D3	D4	D5	D6	D7
P1	3000	1500	8	6.5	1100	-	1520	-	700	580	100
P2	-	-	-	-	-	-	960	-	240	-	400
P3	4200	-	8	-	870	750	780	240	600	200	200
P4	3800	-	7	-	400	-	800	440	-	-	-

We start with $k = \left\lceil \sum_{p,s,n} \frac{\text{demand}_{p,s,n}}{\text{IPV}^{\max}} \right\rceil = \left\lceil \frac{109600}{16000} \right\rceil = 7$ to solve Example 2. Table 5 shows the computational results for Example 2. From this table, the SIM model needs 13 pumping runs to find the optimal detailed schedule in 10393s. NSIM faces 31.98% backordered demands and returns a poor solution of $7702200 up to the maximum computational time of 18000 CPUs.

The best detailed schedule for Example 2 with 13 pumping runs for the SIM model is given in Table 6. From this table, 36 product deliveries are accomplished at distribution depots through the time interval [187.0-207.0]. For example, during time interval [81.5-92.7] h, 4240 m^3 of P3 and 5600 m^3 of P1 are injected from refineries R1 and R2, respectively. During this time interval, 2000 m^3 of P3 are transferred to the branching line N1 to divert the same volume of P3 to depot D5.

Table 5: Computational results of Example 2

	N.R	CPU(s)	Cont. v	Bin. v	Eqs.	Bd (%)	P.c ($)	I.c ($)	z ($)	Relax z ($)	In.gap
SIM	12	7727	2877	820	7448	0	812000	55700	867700	831900	0.041
SIM	13	10393	3081	884	7972	0	809900	55700	865600	831900	0.038
SIM	14	10156	3285	948	8496	0	809900	55700	865600	831900	0.038
NSIM	7	706	1857	500	4835	42	553400	30200	9723600	831900	0.914
NSIM	8	3248	2061	564	5360	36.76	642700	35600	8678300	831900	0.904
NSIM	9	18000	2265	628	5885	31.98	712000	30200	7702200	831900	0.891

Table 6: Amount of products transferred to pipelines and depots of Example 2 (in 10 m^3).

Time Interval [h]	output node		Branchline			Output nodes						
	R1	R2	N1	N2	N3	D1	D2	D3	D4	D5	D6	D7
0-10.5	450_{P1}	-	240_{P1}	-	-	210_{P3}	-	-	-	240_{P2}	-	-
10.5-45.5	1300_{P1}	-	600_{P1}	-	-	700_{P1}	-	-	-	600_{P3}	-	-
45.5-57.5	580_{P3}	-	-	-	-	400_{P1}	-	180_{P2}	-	-	-	-
57.5-71.5	780_{P3}	-	500_{P3}	-	-	-	280_{P3}	-	-	500_{P1}	-	-
71.5-81.5	776.6_{P3}	-	190_{P3}	100_{P3}	-	366.6_{P3}	-	220_{P2}	90_{P3}	-	-	100_{P1}
81.5-92.7	424_{P3}	560_{P1}	200_{P3}	-	-	56_{P3}	168_{P3}	560_{P2}	-	200_{P3}	-	-
92.7-112.6	799.3_{P3}	940_{P1}	400_{P3}	400_{P3}	200_{P1}	97.33_{P3}	302_{P3}	740_{P1}	-	-	200_{P1}	400_{P2}
112.6-128.3	470_{P3}	-	50_{P3}	-	-	-	-	420_{P1}	50_{P3}	-	-	-
128.3-134.3	320_{P4}	-	-	-	180_{P3}	140_{P3}	-	-	-	-	180_{P1}	-
134.3-154.3	693.3_{P4}	-	200_{P4}	200_{P3}	200_{P3}	-	-	29.33_{P1}	-	-	200_{P1}	200_{P3}
154.3-161.0	416.6_{P4}	-	100_{P4}	-	200_{P3}	50_{P4}	-	66.66_{P1}	100_{P3}	-	200_{P3}	-
161.0-187.0	780_{P4}	-	-	-	-	-	-	780_{P3}	-	-		-
187.0-207.0	1590_{P4}	-	440_{P4}	-	-	350_{P4}	-	800_{P4}	440_{P4}	-		-

CONCLUSIONS

In this paper, a mixed integer linear programming model for the detailed scheduling of a tree-like pipeline system with multiple branching lines, refineries and depots has been presented. It used a continuous time representation, in which the pumping run duration and volume are determined by the optimization. The model considered all the operational constraints, such as forbidden sequences, mass balances, injection and distribution constraints and product demands. Compared to previous works on multi-level tree like pipelines, the proposed model has the ability of considering interface volumes between products and simultaneous delivery to distribution depots. The problem aim was to meet product demand during a fixed planning horizon at minimum operational costs including pumping, interface and on/off pump costs. To show the advantages of allowing simultaneous deliveries to distribution depots, two case studies have been solved using the proposed formulation. In the first example, a tree-like pipeline featuring a mainline, two refineries, five depots and two branching lines was considered. The second example dealt with a large scale case study in which a mainline with two refineries and three branching lines should convey four products to seven distribution depots over 10-day scheduling horizon. Results illustrated that the simultaneous delivery mode permits to find the optimal solutions of pipeline system in lower CPU times with regards to non-simultaneous mode.

Future work will involve extending the proposed approach to more complex tree-like pipeline systems in which refineries can be located on branch points and branching lines.

REFERENCES

Boschetto, S. N., Magatão, et al. (2010). An Operational Scheduling Model to Product Distribution through a Pipeline Network. *Industrial & Engineering Chemistry Research, 49, 5661-5682.*

Cafaro, D.C., & Cerdá, J. (2004). Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation. *Computers & Chemical Engineering*, *28*(10), 2053-2068.

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

Cafaro, D. C., Cerdá, J., (2012). A rigorous mathematical formulation for the scheduling of tree-structure pipeline networks. *Industrial & Engineering Chemistry Research, 50: 5064-5085.*

Cafaro, D. C., Cerdá, J., (2012). Rigorous scheduling of mesh-structure refined petroleum pipeline networks. *Computers & Chemical Engineering. 34, 158-203.*

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

Cafaro, D.C., Cerdá, J., (2011). Optimal scheduling of multi-level tree-structure pipeline networks. *Computer Aided Chemical Engineering 29, 980.*

Cafaro, V.G., Cafaro, D.C., Mendéz, C.A. et al. (2012).Detailed scheduling of single-source pipelines with simultaneous deliveries to multiple offtake stations. *Industrial & Engineering Chemistry Research, 51: 6145-6165.*

Cafaro, V.G., Cafaro, D.C., Mendéz, C.A. et al. (2015b).Optimization model for the detailed scheduling of multi-source pipelines. *Computers & Industrial Engineering, 88: 395-409.*

Castro, P. M. (2017). Optimal scheduling of multiproduct pipelines in networks with reversible flow. *Industrial & Engineering Chemistry Research*, *56*(34), 9638-9656.

Castro, P.M. Mostafaei, H. (2019). Batch-centric scheduling formulation for treelike pipeline systems with forbidden product sequences. *Computers & Chemical Engineering, 122, 2-8.*

Castro, P.M., Mostafaei, H., (2017).Product-centric continuous-time formulation for pipeline scheduling. *Computers & Chemical Engineering, 104, 283-295.*

Ghaffari-Hadigheh, A., & Mostafaei, H. (2015). On the scheduling of real world multiproduct pipelines with simultaneous delivery. *Optimization and engineering*, *16*(3), 571-604.

Magatão, L., Arruda, L. V., & Neves Jr, F. (2004). A mixed integer programming approach for scheduling commodities in a pipeline. *Computers & chemical engineering*, *28*(1-2), 171-

185.

Mostafaei, H., & Castro, P. M. (2017). Continuous□time scheduling formulation for straight pipelines. *AIChE Journal*, *63*(6), 1923-1936.

Mostafaei, H., & Castro, P. M. (2018). Hierarchical Decomposition Approach for Detailed Scheduling of Pipeline Systems with Branches. In *International Conference on Operations Research and Enterprise Systems*, 243-266. Springer, Cham.

Mostafaei, H., & Ghaffari Hadigheh, A. (2014). A general modeling framework for the long-term scheduling of multiproduct pipelines with delivery constraints. *Industrial & Engineering Chemistry Research*, *53*(17), 7029-7042.

Mostafaei, H., Castro, P. M., & Ghaffari-Hadigheh, A. (2015C). A novel monolithic MILP framework for lot-sizing and scheduling of multiproduct treelike pipeline networks. *Industrial & Engineering Chemistry Research*, *54*(37), 9202-9221.

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

Rejowski Jr, R., & Pinto, J. M. (2003). Scheduling of a multiproduct pipeline system. *Computers & Chemical Engineering*, *27*(8-9), 1229-1246.

Relvas, S., Matos, H. A., Barbosa-Póvoa, A. P. F., Fialho, J., & Pinheiro, A. S. (2006). Pipeline scheduling and inventory management of a multiproduct distribution oil system. *Industrial & Engineering Chemistry Research*, *45*(23), 7841-7855.

Taherkhani, M. (2018). An MILP approach for scheduling of tree-like pipelines with dual purpose terminals. *Operational Research*, 1-29.

Taherkhani, M., Seifbarghy, M., Tavakkoli-Moghaddam, R., & Fattahi, P. (2017). Mixed-integer linear programming model for tree-like pipeline scheduling problem with intermediate due dates on demands. *Operational Research*, 1-27.