AN MILP APPROACH FOR ALLOCATING AND SEQUENCING BATCHES IN A SINGLE PIPELINE WITH MULTIPLE BLEED-OFFS

William Hitoshi Tsunoda Meira, Marcos Henrique da Silva,
Leandro Magatão, Flávio Neves Junior, Lúcia Valéria Ramos de Arruda
Federal University of Technology – Paraná (UTFPR) /
Graduate Program in Electrical and Computer Engineering (CPGEI)
Av. Sete de Setembro, 3165, 80230-901, Curitiba, PR
williammeira@gmail.com, tio.makin@gmail.com
magatao@utfpr.edu.br, neves@utfpr.edu.br, lvrarruda@utfpr.edu.br

Paulo Cesar Ribas
PETROBRAS / CENPES
Avenida Jequitiba, 950, Expansão 3º andar, 21941-915, Ilha do Fundão, Rio de Janeiro, RJ
paulo.ribas@petrobras.com.br

ABSTRACT

The transportation of oil derivates through pipelines is a cost-effective logistic option to distribute the production and supply the network demands. This paper focuses on determining the optimized allocation and sequencing for batches of oil derivatives transported in a specific pipeline network, which involves a single pipeline and multiple bleed-offs. The proposed MILP model determines the sequence and volume of each allocated batch, along a scheduling horizon of 30 days, obeying operational constraints, and trying to minimize inventory violations, maintaining the pump flow rate as constant as possible, and maximizing batch volumes. The model is also capable of dealing with incompatibilities when sequencing two or more products, maintenance of tanks, reduction of pipeline flow rate, or even a complete pipeline stop, and non-linear demands. Results were obtained using data of real scenarios in order of few minutes for a three-product input condition.

KEYWORDS. MILP model. Pipeline scheduling. Multiple bleed-offs.

Main area. P&G (OR in Oil & Gas), L&T (Logistics and Transportation), PM (Mathematical Programming).

1. Introduction

The demand and production of oil are constantly growing, bringing the need to use an efficient logistic modal in order to reduce operating costs. In this scope, pipeline transportation has become a key transportation modal for all the oil derivatives in Brazil. However, the efficiency in pipeline transportation brings the need of considering several constraints and operational procedures, including determining the management of transport operations to attend safety aspects of refineries and customers involved. Currently, the management, or scheduling, of delivering production from a refinery to consumer terminals is a decision of a group of experts. These experts generate a management that is based on, mostly, past experiences and manual calculations. Thus, there is a need for better programming decision practices in order to maintain inventories at more controlled levels, which improves the operational and financial gains.

The literature contains articles that deal with scheduling decisions in multi-product pipelines. These articles, generally, use MILP models in cooperation with decomposition strategies. Some feasible solutions, which contributed to the state of art of the theme, were described by Magatão *et al.* (2004), Relvas *et al.* (2008), Boschetto *et al.* (2010), and Kira *et al.* (2013). The simplifications/decompositions are needed due to the high combinatorial complexity of addressed scheduling problems.

This paper is based on the computational approach proposed by Kira *et al.* (2010) and Kira *et al.* (2012). The paper presents improvements for the MILP model of Kira *et al.* (2013). For instance, in this paper the former model is modified in order to attend initial inventory within the pipeline and accept more than two products. The model describes an operational scheduling that optimizes the sequencing and the volume allocation of batches of products to be pumped from a refinery to terminals, at a certain flow rate, for a given scheduling horizon.

The paper is organized as follows. Section 2 describes the addressed problem. Section 3 details the problem approach. Section 4 details the allocation/sequencing MILP model. Section 5 presents some operational results, based on a real-world scenario. Section 6 presents the concluding remarks.

2. Problem Description

The problem is structured as shown in Figure 1. It consists of a single pipeline network and six operational areas, which includes one main refinery and five distribution terminals (also called as consumers – T1 to T5). Each terminal has several tanks to store the inventory of determined products. The terminals and the refinery are linked by segments of the pipeline (PL1 to PL5), each one with specific volume and pump-flow constraints. In a simplified explanation, the refinery pumps oil derivatives in a unidirectional sequence along the terminal consumers, according to their demands. The sequencing of pumped products by the refinery is made in batches of determined volume in the first pipeline (PL1), e.g.: diesel, plug product, gasoline, plug product, diesel, plug product, LPG, and so on. The plug product is a particular type of product that needs to be inserted between two different products due to their incompatibility (Magatão *et al.*, 2004).

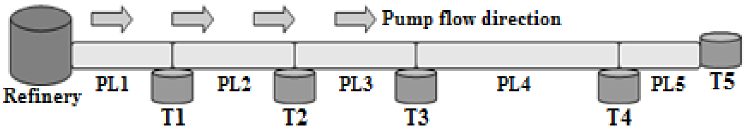

Figure 1 - Single pipeline with multiple bleed-off example.

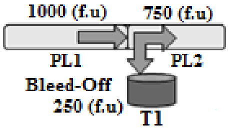

Figure 2 - Bleed-off example.

One important goal of the system is to maintain the refinery pump flow rate as constant as possible, since the extension of the pipeline is more than 900 km. Thus, pipeline stoppages are just accepted for maintenance purposes and do not configure a typical operational characteristic. In addition, terminals can receive products using bleed-off operations, while products pumped by the refinery are still crossing the pipeline to other terminals. One example of a bleed-off operation can be seen in Figure 2, demonstrating a process that causes a batch to "shrink", losing volume along the subsequent pipeline. Still in Figure 2, PL1 pumps a batch at flow rate of 1000 f.u. (flow units) and terminal T1 bleeds-off at 250 f.u. in order to receive part of the passing batch. Thus, the PL2 will notice the batch at a reduced flow rate, 750 f.u. This flow difference is given because the bleed-off operation at terminal T1. Besides that reduced flow rate, the pump-flow balance is still maintained, because pipeline PL1 flow rate is equivalent to the flow rate at pipeline PL2 added to the bleed-off at terminal T1.

3. Problem Approach

A full scope of the proposed problem approach can be seen in Figure 3. The solution approach is based on heuristic processes in cooperation with MILP models. The heuristics are developed to simplify the variables scope used in MILP models. The entire approach uses three MILP models: allocation and sequencing; bleed-offs allocation; and repumping model. The main focus of this work is to explain how the allocation and sequencing MILP model works and how it interferes in the final scheduling solution.

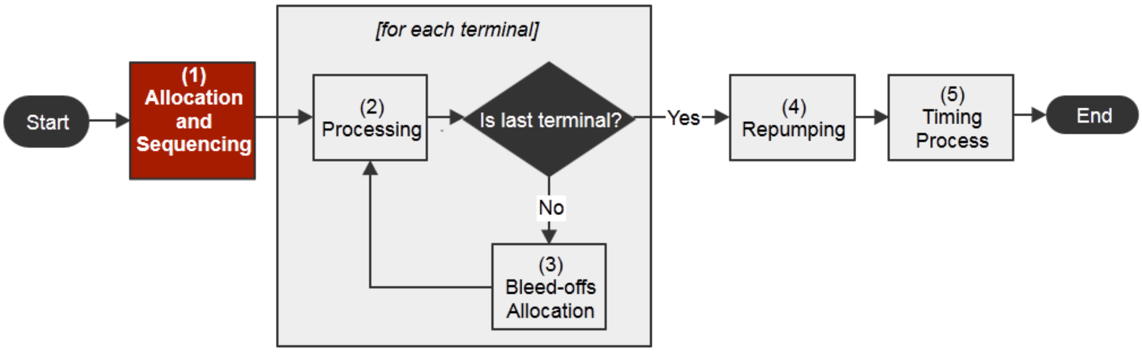

Figure 3 – Problem approach workflow.

The steps to create a scheduling consists in: input data (named start), executing allocation and sequencing (1), executing the processing heuristic (2), determining bleed-offs (3), obtaining the repumping (4), executing timing process (5), and providing the outputs.

The first step is to input the needed data to solve the problem in a given programming horizon, typically 30 days or more. It includes: the expected demands of all terminals; the expected production of the refinery; the actual inventory in the pipeline and the current flow-rate of each initial batch; the initial inventory in each tank; maintenances of tanks or the pipeline scheduled during the given programming horizon; and, operational constraints.

The second step is to allocate everything that the sequencing model needs to solve the problem, determining how the refinery will pump the products from the first pipeline to the others. The full description of the allocation and sequencing model is described in section 4 and is the focus of this work. The sequencing model provides all parameters needed to create the next steps.

The third step is processing, bleed-offs allocation. It starts by inserting a small volume of plug product between two sequenced batches. Then, considering operational constraints, it propagates the flow through the pipeline, one terminal per iteration. Each processed terminal generates data to the next and so on, until T4. The last case occurs in segment PL5 and terminal T5, when a repumping model is needed to the last iteration.

After all terminals have been scheduled, the last step is to generate the timing process, correctly adjusting the temporal issues in order to generate the complete output.

4. Allocation and Sequencing MILP Model

Allocation and sequencing of batches are obtained by a single MILP optimization model. The allocation determines the volume of each batch; the sequencing is responsible for the ordering of the batches to be pumped during the scheduling horizon. A batch is associated with a product, a volume, and a flow rate. The total volume allocated to a particular product should supply the demand of the consumer terminals considering, also, the initial inventory of the scenario and the already pumped batches through the pipeline.

The model assumes that the refinery works in an ideal way. In other words, the production and inventory of the refinery is sufficient to supply the total demand from all the consumer terminals.

Other model assumption is the aggregation of tanks that store the same product in a unique tank, instead of working with them separately. If the terminal has 3 tanks of 1.000 v.u. (volumetric units) available, then the model will assume the existence of just one tank, whose capacity is equal to the aggregated capacity, 3.000 v.u.

The model is capable to support the following cases:

- Non-linear (step-wise) demands are allowed for products of each consumer terminal, as illustrated in Figure 4;
- Pipeline reduction of flow rate, due to pipeline or pump maintenance. It supports also full maintenance, which stops the refinery from pumping during a period;
- Tank maintenance in terminals. Thus, the respective assumed aggregated tank reduces immediately its capacity during the maintenance period;
- Incompatibility between two products. As a physical or operational incompatibility some products cannot be pumped in a determined order;
- Two or more products can be transported in the pipeline.

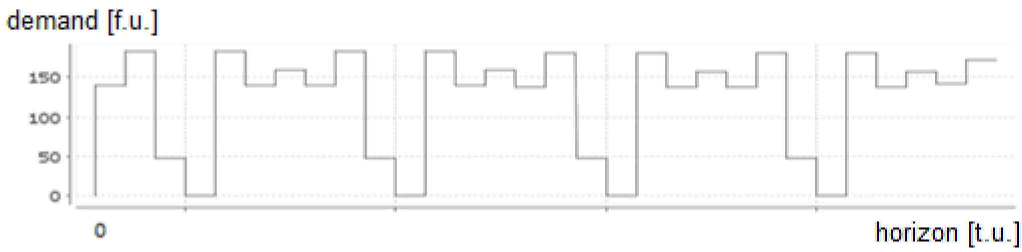

Figure 4 : Example of demand in a terminal

Basically, the model is an evolution of the study described in Kira *et al.* (2013). The nomenclature used is maintained as originally presented by Kira *et al.* (2013). The sets and elements are shown in Table 1, with products, events, intervals, and batches. An event is a combination of a product batch in a given time. The interval is a combination of two events (initial and final events that define a time slot). Parameters are described in Table 2 and Table 3. Decision variables are in Table 4. In order to easy the identification, all variables are named with a starting upper case letter and parameters with a lower case.

Table 1 – Sets and elements

Set/Element	Description
$p \in P$	p is a product (element) of products P (set)
$p_2 \in IP_{p_1}$	p_2 is a product (element) that is incompatible with p_1; the set IP_{p_1}, indicates

	incompatible products with p_1
$e \in E$	*e* is an event (element) of events *E* (set); initial event starts with $e = 0$
$i \in I$	*i* is an interval (element) of intervals *I* (set); initial interval starts with $i = 1$
$t \in T$	*t* is a terminal (element) of terminals *T* (set)
$b \in BP$	*b* is a batch (element) of the batches *BP* (set) that initially, event *e* = 0, are in the pipeline

Table 2 - Parameters

Parameters	Description
$minFlow_{p,i}$ / $maxFlow_{p,i}$	Minimum and maximum refinery's pumping flow rate, respectively in f.u. (flow units)
$intervalLength_i$	Elapsed time between the interval's beginning and ending time in t.u. (time units)
uV	Volume (in v.u. – volumetric units) upper bound
$minBatchVolume_p$/ $maxBatchVolume_p$	Minimum and maximum volume for batches pumped from the refinery (v.u.)
$initialInventory_{t,p}$	Terminals initial inventory state at the initial event $e = 0$ (v.u.)
$pipelineInventory_b$	Pipeline initial inventory state at the initial event *e* = 0 (v.u)
$initialProduct_p$	Binary parameter that indicates if the product *p* is being pumped from the refinery at the initial event *e* = 0
$initialCumulatedVolume$	Cumulated volume pumped at the initial event *e* = 0 (v.u.)
$demand_{t,p,i}$	Local market demand for each terminal and products, during each interval (v.u.)
$maxBleedOffFlow_{t,p,i}$	Maximum bleed-off flow rate for each terminal and products, in each interval (f.u.)
$capacity_{t,p,e}$	Storage capacity for each terminal and product, during each interval (v.u.)
$minInventory_{t,p,e}$/ $maxInventory_{t,p,e}$	Minimum and maximum inventory for each terminal and product, during each interval, respectively (v.u.)
$minGoal_{t,p,e}$ / $maxGoal_{t,p,e}$	Minimum and maximum inventory goal (desired level) for each terminal and product, during each interval, respectively (v.u.)

Table 3 - Weighting parameters for the objective function.

Parameter	Description
kShort / *kOver*	Inventory shortage and inventory overflow cost function weighting, respectively
kMinI / *kMaxI*	Minimum and maximum inventory violation cost function weighting, respectively
kMinG / *kMaxG*	Minimum and maximum goal inventory violation cost function weighting, respectively
kHMeanF / *kLMeanF*	Higher and lower than mean flow rate violation cost function weighting, respectively
kSwap	Product swap cost function weighting

Table 4 – Decision variables.

Decision Variable	Type	Description
$AllocatedProduct_{p,i}$	{0;1}	Variable that assigns 1 if a product *p* was allocated in the interval *i*, and 0 otherwise
$ProductSwapped_{p,i}$	{0;1}	Boolean variable denoting if a product *p*, allocated in interval *i-1*, was swapped by other product in interval *i*
$ProductFlow_{p,i}$	$\Re+$	Product *p* flow rate assigned in the interval *i* (f.u.)
$AllocatedVolume_{p,i}$	$\Re+$	Product *p* volume allocated in the interval *i* (v.u.)
$VolumeAllocatedPipelineInventory_{b,i,t}$	$\Re+$	Volume of the pipeline inventory allocated for a terminal *t* in the interval *i* (v.u.)
$CumulatedVolume_{p,i}$	$\Re+$	Accumulated volume obtained due to the same consecutive product *p* allocated between interval *i* and *i-1* (v.u.)
$MeanFlow$	$\Re+$	Mean pumping flow rate at the refinery (f.u.)
$OverMeanFlowDiff_i$ / $UnderMeanFlowDiff_i$	$\Re+$	Over and under mean flow difference between flow rate allocated in interval *i* and mean flow, respectively (f.u.)
$Inventory_{t,p,e}$	$\Re+$	Inventory of product *p* in terminal *t*, at the event *e* (v.u.)

$FractionedVolume_{t,p,e}$	$\Re+$	Fraction of allocated volume of product p in interval i delivered in terminal t (v.u.)
$InventoryShortage_{t,p,e}$	$\Re+$	Lack of product p in terminal t at event e that was undelivered to market demand (v.u.)
$InventoryOverflow_{t,p,e}$	$\Re+$	Overflow capacity of product p on terminal t at event e (v.u.)
$MinInventoryViolation_{t,p,e}$	$\Re+$	Minimum inventory violation of product p on terminal t at event e (v.u.)
$MaxInventoryViolation_{t,p,e}$	$\Re+$	Maximum inventory violation of product p on terminal t at event e (v.u.)
$MinGoalViolation_{t,p,e}$	$\Re+$	Minimum goal inventory of product p on terminal t at event e (v.u.)
$MaxGoalViolation_{t,p,e}$	$\Re+$	Maximum goal inventory of product p on terminal t at event e (v.u.)

The objective function of the model, shown in equation (1), minimizes three groups of variables. The first group (*Group* 1) is responsible for maintaining the inventory level, during the horizon, near the desired by the terminals, so it minimizes the inventory violation variables: $InventoryOverflow_{t,p,e}$, $MaxInventoryViolation_{t,p,e}$, $MaxGoalViolation_{t,p,e}$ $MinGoalViolation_{t,p,e}$ $MinInventoryViolation_{t,p,e}$, and $InventoryShortage_{t,p,e}$. The second part (*Group* 2) maintains the pumping flow rate at the refinery as equal as possible the mean flow rate, which means the model minimizes the $OverMeanFlowDiff_i$ and $UnderMeanFlowDiff_i$. The third part (*Group* 3) minimizes products swaps ($ProductSwapped_{p,i}$), in order to maximize the volume of each batch. All the terms of the equation are weighted, respectively, by the parameters presented in Table 3. The parameters where established in order to consider violations of *Group* 1 with more impact than the ones of *Group* 2 and *Group* 3, respectively. Thus, in practice, *kOver* > *kShort* > *kMinI* > *kMaxI* >> *kMinG* > *kMaxG* >> *kHMeanF* > *kLMeanF* >> *kSwap*.

$$
\begin{aligned}
&\textit{Minimize}\\
&z = \underbrace{\begin{aligned}&\sum_{t\in T, p\in P, e\in E} InventoryOverflow_{t,p,e}\cdot kOver + \sum_{t\in T, p\in P, e\in E} MaxInventoryViolation_{t,p,e}\cdot kMaxI +\\
&+\sum_{t\in T, p\in P, e\in E} MaxGoalViolation_{t,p,e}\cdot kMaxG + \sum_{t\in T, p\in P, e\in E} MinGoalViolation_{t,p,e}\cdot kMinG +\\
&+\sum_{t\in T, p\in P, e\in E} MinInventoryViolation_{t,p,e}\cdot kMinI + \sum_{t\in T, p\in P, e\in E} InventoryShortage_{t,p,e}\cdot kShort +\end{aligned}}_{Group\,1}\\
&\underbrace{+\sum_{i\in I} OverMeanFlowDiff_i\cdot kHMeanF + \sum_{i\in I} UnderMeanFlowDiff_i\cdot kLMeanF +}_{Group\,2}\\
&\underbrace{+\sum_{p\in P, i\in I} productSwapped_{p,i}\cdot kSwap}_{Group\,3}
\end{aligned} \quad (1)
$$

The expressions from (2) to (34) define the constraints of the model; each of them will be explained for a more comprehensive understanding.

The equation (2) defines that the refinery can pump just one product in each interval i.

$$\sum_{p\in P} AllocatedProduct_{p,i} = 1 \qquad \forall\, i \in I \quad (2)$$

The allocated product flow rate, in each interval i, must be between the minimum and maximum limits, as established by the inequalities (3) and (4).

$$ProductFlow_{p,i} \geq minFlow_{p,i}\cdot AllocatedProduct_{p,i} \qquad \forall\, p\in P, i\in I \quad (3)$$

$$ProductFlow_{p,i} \leq maxFlow_{p,i} \cdot AllocatedProduct_{p,i} \qquad \forall\ p \in P, i \in I \tag{4}$$

The allocated volume of product *p* must be the product flow rate multiplied by the interval length, as shown in equation (5).

$$AllocatedVolume_{p,i} = ProductFlow_{p,i} \cdot intervalLength_i \qquad \forall\ p \in P, i \in I \tag{5}$$

The mean flow rate, equation (6), is obtained weighting the product flow by the respective interval length and dividing by the horizon size, which is obtained by the sum of all interval lengths.

$$MeanFlow = \frac{\sum_{i \in I} \sum_{p \in P} ProductFlow_{p,i} \cdot intervalLength_i}{\sum_{i \in I} intervalLength_i} \tag{6}$$

In order to maintain the pumping flow as constant as possible at the refinery, the inequalities (7) and (8) expresses that the difference between the product flow and the mean flow should be close to zero in each interval *i*, since the objective function minimizes the variables $OverMeanFlowDiff_i$ and $UnderMeanFlowDiff_i$.

$$\sum_{p \in P} ProductFlow_{p,i} - MeanFlow - OverMeanFlowDiff_i \leq 0 \qquad \forall\ i \in I \tag{7}$$

$$\sum_{p \in P} ProductFlow_{p,i} - MeanFlow + UnderMeanFlowDiff_i \geq 0 \qquad \forall\ i \in I \tag{8}$$

The decision variable $ProductSwapped_{p,i}$ is related with the $AllocatedProduct_{p,i}$ by the inequalities (9), (10), (11) and (12), which determinates that if an allocated product *p* in interval *i -1* was not allocated in interval *i*, then the product being pumped was swapped.

$$AllocatedProduct_{p,i} + ProductSwapped_{p,i} - initialProduct_p \geq 0 \qquad \forall\ p \in P, i \in I \mid i = 1 \tag{9}$$

$$AllocatedProduct_{p,i} + ProductSwapped_{p,i} - ProductSwapped_{p,i-1} \geq 0 \qquad \forall\ p \in P, i \in I \mid i > 1 \tag{10}$$

$$AllocatedProduct_{p,i-1} - initialProduct_p \geq 0 \qquad \forall\ p \in P, i \in I \mid i > 1 \tag{11}$$

$$AllocatedProduct_{p,i-1} - ProductSwapped_{p,i} \geq 0 \qquad \forall\ p \in P, i \in I \mid i > 1 \tag{12}$$

$$AllocatedProduct_{p,i} + ProductSwapped_{p,i} \leq 1 \qquad \forall\ p \in P, i \in I \tag{13}$$

Due to chemical or physical incompatibility some products cannot be pumped in sequence. Hence, the sequencing must verify the incompatibility rules determined by the set IP_{p_1}, which are covered by the inequalities (14) and (15).

$$initialProduct_{p1} + AllocatedProduct_{p2,i} \leq 1 \qquad \forall\ p1, p2 \in P, i \in I \mid i = 1, p2 \in IP_{p1} \tag{14}$$

$$AllocatedProduct_{p1,i-1} + AllocatedProduct_{p2,i} \leq 1 \qquad \forall\ p1, p2 \in P, i \in I \mid i > 1, p2 \in IP_{p1} \tag{15}$$

The cumulated volume is obtained by the sum of the allocated volume during consecutive intervals of the same allocated product. It defines the volume of a batch along the intervals. Inequalities (16) to (20) define the $CumulatedVolume_{p,i}$ variable, as explained.

$$\begin{aligned} &CumulatedVolume_{p,i} - (AllocatedVolume_{p,i} + initialCumulatedVolume) \leq \\ &uV \cdot (1 - AllocatedProduct_{p,i}) \qquad \forall\ p \in P, i \in I \mid i = 1 \end{aligned} \tag{16}$$

$$CumulatedVolume_{p,i} - (AllocatedVolume_{p,i} + CumulatedVolume_{p,i-1}) \leq uV \cdot (1 - AllocatedProduct_{p,i}) \quad \forall\ p \in P, i \in I \mid i > 1 \tag{17}$$

$$CumulatedVolume_{p,i} - (AllocatedVolume_{p,i} + initialCumulatedVolume) \geq -uV \cdot (1 - AllocatedProduct_{p,i}) \quad \forall\ p \in P, i \in I \mid i = 1 \tag{18}$$

$$CumulatedVolume_{p,i} - (AllocatedVolume_{p,i} + CumulatedVolume_{p,i-1}) \geq -uV \cdot (1 - AllocatedProduct_{p,i}) \quad \forall\ p \in P, i \in I \mid i > 1 \tag{19}$$

$$CumulatedVolume_{p,i} \leq uV \cdot (AllocatedProduct_{p,i}) \quad \forall\ p \in P, i \in I \tag{20}$$

The batch volume is bounded by a minimum and a maximum amount due to the system operation constraints. The inequality (21) indicates that the cumulate volume should be higher than the minimum batch volume if the product is swapped. However, this constraint is not applicable to the last interval, because the volume could be supplemented after the programming horizon. For the maximum batch volume, the inequalities (22) and (23) determine the upper bound of the decision variable $CumulateVolume_{p,i}$.

$$CumulatedVolume_{p,i-1} \geq minBatchVolume_{p} \cdot ProductSwapped_{p,i} \quad \forall\ p \in P, i \in I \mid i > 1 \tag{21}$$

$$CumulatedVolume_{p,i} \leq maxBatchVolume_{p} + uV \cdot (1 - ProductSwapped_{p,i+1}) \quad \forall\ p \in P, i \in I \mid i < N \tag{22}$$

$$CumulatedVolume_{p,i} \leq maxBatchVolume_{p} \quad \forall\ p \in P, i \in I \mid i = N \tag{23}$$

The $FractionedVolume_{t,p,i}$ represents the fraction of the allocated volume of product *p* in the interval *i* delivered to the terminal *t* by the operation known as bleed-off. According to the inequalities (24) and (25), the fractioned volume is delimited by a maximum flow rate and cannot be higher than the allocated volume in the interval *i*.

$$FractionedVolume_{t,p,i} \leq AllocatedVolume_{p,i} \quad \forall\ t \in T, p \in P, i \in I \tag{24}$$

$$FractionedVolume_{t,p,i} \leq maxBleedOffFlow_{t,p,i} \cdot intervalLength_{i} \quad \forall\ t \in T, p \in P, i \in I \tag{25}$$

The allocated volume of product *p* during the interval *i* is equal to the sum of all fractioned volumes delivered during this interval to all terminals, as stated in equation (26).

$$\sum_{t \in T} FractionedVolume_{t,p,i} = AllocatedVolume_{p,i} \quad \forall\ p \in P, i \in I \tag{26}$$

The inventory of the product *p* at the interval *i* is defined by the equations (27) and (28). The former sets the initial inventory of the scenario (event *e* = 0), which is equals to the parameter $initialInventory_{t,p,e}$. The later determines the inventory for the remaining events.

$$Inventory_{t,p,e} = initialInventory_{t,p,e} \quad \forall\ t \in T, p \in P, e \in E \mid e = 0 \tag{27}$$

$$Inventory_{t,p,e} = Inventory_{t,p,e-1} - demand_{t,p,i} + FractionedVolume_{t,p,i} \quad \forall\ t \in T, p \in P, e \in E \mid e > 0 \tag{28}$$

The variables $InventoryOverflow_{t,p,e}$, $MaxInventoryViolation_{t,p,e}$, $MaxGoalViolation_{t,p,e}$, $MinGoalViolation_{t,p,e}$, $MinInventoryViolation_{t,p,e}$, $InventoryShortage_{t,p,e}$ are based on the respective inventory level as the violation of the desired value of a product *p* at the terminal *t* in the event *e*. These variables are defined by the inequalities (29), (30), (31), (32), (33), and (34). The first three are upper violations and the others are under violations.

$$Inventory_{t,p,e} - InventoryOverflow_{t,p,e} \leq capacity_{t,p,e} \quad \forall\, t \in T, p \in P, e \in E \qquad (29)$$

$$Inventory_{t,p,e} - MaxInventoryViolation_{t,p,e} \leq maxInventory_{t,p,e} \quad \forall\, t \in T, p \in P, e \in E \qquad (30)$$

$$Inventory_{t,p,e} - MaxGoalViolation_{t,p,e} \leq maxGoal_{t,p,e} \quad \forall\, t \in T, p \in P, e \in E \qquad (31)$$

$$Inventory_{t,p,e} + MinGoalViolation_{t,p,e} \geq minGoal_{t,p,e} \quad \forall\, t \in T, p \in P, e \in E \qquad (32)$$

$$Inventory_{t,p,e} + MinInventoryViolation_{t,p,e} \geq minInventory_{t,p,e} \quad \forall\, t \in T, p \in P, e \in E \qquad (33)$$

$$Inventory_{t,p,e} + InventoryShortage_{t,p,e} \geq 0 \quad \forall\, t \in T, p \in P, e \in E \qquad (34)$$

5. Results

The MILP model explained in section 4 was applied into a real scenario data, with thirty days of programming horizon time, as described in section 2. It contains three products (P1, P2, and P3) with linear demands showed in Table 5. Further, the incompatibility constraints to the model are given in Table 6, indicating whether a particular product batch can stand next to a different product batch by the model.

Table 5 - Linear demand for each product during the programming horizon.

	Demand (in v.u.)
P1	554,990.00
P2	148,000.00
P3	89,000.00

Table 6 - Incompatibility between two given products (0 compatible, 1 incompatible).

	P1	P2	P3
P1	0	0	0
P2	0	0	1
P3	0	1	0

Furthermore, as input data, the model receives the current inventory of all pipelines, as illustrated in Figure 5. Therein, R indicates the refinery, T1-T5 indicate terminals, P1-P3 indicate products, and between batches of subsequent products there exists, in fact, a small batch with a plug product. For programming purposes, a batch route is established, as detailed in Table 7.

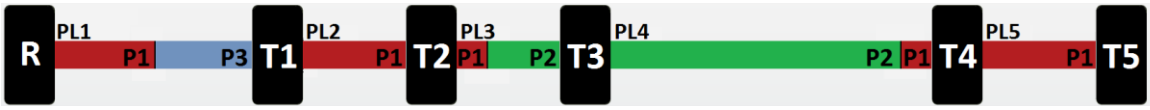

Figure 5 - Inventory overview of all segments of the pipeline.

Also, a full description of the initial inventory of all pipelines is showed in Table 7. The batch route indicates where the batch of a product started the pumping to its final destination, a terminal. For example, RT4009 indicates that a product is going from refinery (R) to terminal 4 (T4), being the ninth batch (009) beginning from refinery to terminal 4.

Table 7 - Initial inventory overview for the entire pipeline.

Pipeline	Batch Route	Volume Units (v.u.)
PL1	RT4009	20036
PL1	RT4008	500
PL1	RT4007	19223
PL2	RT4006	500
PL2	RT4005	25379
PL3	RT4005	7547
PL3	RT4004	500
PL3	RT4003	17274
PL4	RT4003	53530
PL4	RT4002	500
PL4	RT4001	5646
PL5	T4T5001	13739

The capacities of each pipeline are shown in Table 8, with the overall pipeline network capacity of 164374 v.u. In this scenario, neither terminals have maintenance in their tanks nor the pipelines have maintenance during the programming horizon. The overall aggregate capacity for each product in each terminal can be seen in Table 9.

Table 8: Inventory for each pipeline.

	Inventory capacity (v.u.)
PL1	39759
PL2	25879
PL3	25321
PL4	59676
PL5	13739
Total inventory	164374

Table 9: Overall tank capacity in each terminal.

	P1 (v.u.)	P2 (v.u.)	P3 (v.u.)
T1	8000	22800	20000
T2	8000	16800	8000
T3	7600	22800	16000
T4	25800	60400	23800
T5	22800	15800	15800

The MILP model was implemented and solved in IBM ILOG CPLEX Optimization IDE Studio v12.5, running on a platform composed of an Intel Core i7 (2.93 GHz) with 4.00 GB of RAM, using Windows 7 64-bit operating system. The approximate runtime of the allocation and sequencing MILP model was 288 seconds, reaching the integer optimal solution. The final runtime of all processing steps presented in Figure 3 was, approximately, 10 minutes.

Table 10 lists the batches generated by the MILP model for the sequencing step, already including plug products, which are inserted by the “processing step” shown at Figure 3. It lists: batch number; product type; beginning and ending time of the pumping interval (day/month hour:minute); pumping volume given the current interval; flow rate at the given batch. Also, there is a plug product between each batch to maintain compatibility between products. Table 10 also indicates that the flow rate obtained by the MILP model was maintained constant, keeping all operational constraints in a feasible way.

Table 10 - Batch analysis of the first pipeline.

Batch Number	*Product*	*Begin time* (t.u.)	*End time* (t.u.)	*Interval Volume* (v.u.)	*Flow rate* (f.u.)
1	*P1*	01/11 00:00	01/11 16:52	19,604.986	1,062.06
2	*P2*	01/11 17:18	03/11 15:41	53,916.726	1,062.06
3	*P1*	03/11 16:07	04/11 18:52	31,075.566	1,062.06
4	*P3*	04/11 19:18	05/11 21:41	30,675.566	1,062.06
5	*P1*	05/11 22:07	06/11 19:52	25,265.276	1,062.06
6	*P2*	06/11 20:18	08/11 23:41	59,727.016	1,062.06
7	*P1*	09/11 00:07	10/11 02:52	31,075.566	1,062.06
8	*P2*	10/11 03:18	12/11 15:41	70,185.538	1,062.06
9	*P1*	12/11 16:07	13/11 14:52	26.427,334	1,062.06
10	*P3*	13/11 15:18	15/11 03:41	42,296.146	1,062.06
11	*P1*	15/11 04:07	16/11 06:52	31,075.566	1,062.06
12	*P2*	16/11 07:18	18/11 22:41	73,671.712	1,062.06
13	*P1*	18/11 23:07	20/11 01:52	31,075.566	1,062.06
14	*P3*	20/11 02:18	20/11 23:41	24,865.276	1,062.06
15	*P1*	21/11 00:07	22/11 00:52	28,751.45	1,062.06
16	*P2*	22/11 01:18	25/11 03:41	86,454.35	1,062.06
17	*P1*	25/11 04:07	26/11 05:52	29,913.508	1,062.06
18	*P3*	26/11 06:18	26/11 19:41	15,568.812	1,062.06
19	*P1*	26/11 20:07	27/11 18:52	26,427.334	1,062.06
20	*P2*	27/11 19:18	30/11 00:41	62,051.132	1,062.06
21	*P1*	30/11 01:07	01/12 00:00	26,577.334	1,062.06

The results obtained by the allocation and sequencing MILP model can be propagated through the pipeline segments and terminals by the next steps illustrated in Figure 3. At the end of timing process step, a Gantt chart for all the pipelines is generated with the output data, as showed in Figure 6. For example, the Gantt line for pipeline PL1 represents in a graphical form pumping batches detailed in Table 10. In PL1 it is possible to verify that the MILP model respected compatibility constraints for three products, as highlighted in Figure 7. As indicated in Table 6, products P2 and P3 are incompatible. Thus, the model indicates a batch of product P1 in between them. Another important feature that can be observed in Figure 6 is that pipeline PL5 has no direct pumping dependency with pipeline PL1. This fact occurs because PL5 works with the repumping process between terminal T4 and terminal T5, being able to have a different schedule in comparison to the other pipelines.

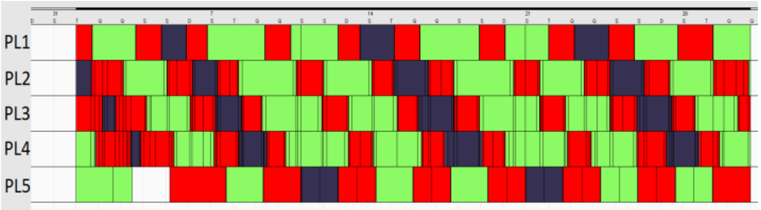

Figure 6 - Gantt chart for the final schedule of pipelines

Figure 7 : Batch order for incompatibility reasons

Another study was done by increasing the programming horizon of the given scenario at intervals of five days to obtain the allocation and sequencing model computational behavior. Within the considered real problem, the number of pipelines and terminals is constant. The number of involved products can often change from two to three. Therefore, study cases with up to three products represent a real operation. The considered scheduling horizon, however, can be increased for more than 30 days, since a product can take many days until to reach its final destination. Thus, investigations with longer scheduling horizons can be relevant. The new scenarios (35, 40, and 45 days) were replicated from the original one, only increasing their linear demands in proportion of the respective programming horizon.

The growing of constraints, total number of variables, and binary variables can be seen in Table 11. These increase tended to be linear. In contrast, the MILP model runtime behaved exponentially as shown in Figure 8. Thus, as future developments, improvements in the MILP model can be made in order to reduce the processing time.

Table 11 – Increase of constraints, variables, and binary variables as the programming horizon is increased.

Horizon Time (in days)	Constraints	Variables	Binary Variables	MILP runtime
30	11384	8684	330	00:04:40
35	13229	10070	384	00:14:55
40	15279	11610	444	01:28:14
45	16921	12844	492	06:43:00

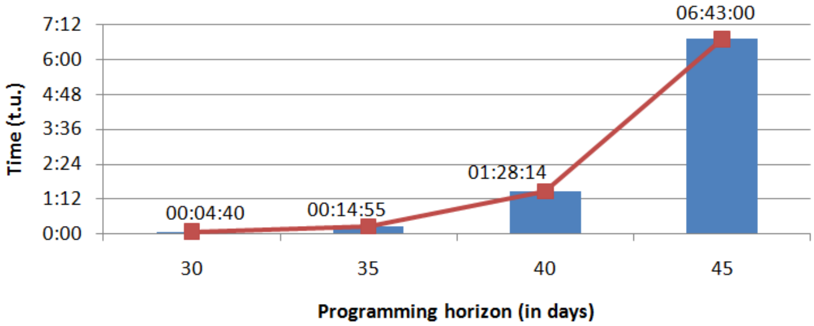

Figure 8 – MILP runtime behavior within different programming horizons.

6. Conclusions

This paper presented an allocation and sequencing MILP model for a single pipeline with multiple bleed-offs that can transport two or more products. Due to the pipeline length (≈940 km) and volume (164374 v.u.), a single batch can take many days to reach its final destination. Thus, a bad combination of volume and product sequencing can result in inventory shortages or overflows. Due to these issues, the proposed MILP model is an important feature of the proposed decomposition approach, highlighted in Figure 3. The model can create sequences of batches, considering four main objectives: maximizing batch volume, minimizing variations on refinery's flow rate, keeping inventory control, and maintaining a valid product order of batches (compatibility issues).

The allocation and sequencing MILP model has obtained quite satisfactory results for operational purposes, as shown in Section 5. The model can suggest solutions that maintain a constant flow rate in pipeline 1, while still kept the ordering constraints with three products. Besides, the model still took in consideration the inventory of tanks at their respective terminals in order to create the final scheduling. However, the duration time to create the full scheduling for three products was slightly longer, when compared to a scenario of two products, as shown by Kira *et al.* (2012), which took 90 seconds for the entire scheduling procedure. Furthermore, the MILP model can solve scenarios with longer programming horizon than 30 days but it needs a much higher execution runtime, as it was shown in Figure 8.

For further studies and researches, improvements in the MILP model can be made, in a way to reduce the processing time, also looking forward to generate scenarios with longer scheduling horizons, supporting a feasible response with the existence of two or more products.

Acknowledgements

To the financial support from ANP/FINEP (PRH-ANP/FINEP, PRH10/UTFPR), PFRH-PETROBRAS (Agreement 6000.0067933.11.4), CENPES-PETROBRAS (Cooperation Term 0050.0066666.11.9) and CNPq (Grants 304037/2010-9, 311877/2009-5, and 305405/2012-8).

References

Boschetto, S.N.; Magatão, L.; Brondani, W.M.; Neves-Jr., F.; Arruda, L.V.R.; Barbosa-Póvoa; A.P.F.D.; Relvas, S. *Ind. Eng.Chem. Res*. 2010, 49, 5661-5682.

Ibm Ilog Cplex. *Language User's Manual*. IBM Corporation, 2012.

Kira, G.; Silva, M.H.; Magatão, L.; Tsuda, H. T.; Nascimento, G. L.; Arruda, L.V.R.; Ribas, P.C. Oil derivatives allocation and sequencing in a single pipeline with multiple bleed-offs. In: Rio Pipeline, 5, 2013, Rio de Janeiro.

Kira, G.; Lara, L.G.; Magatão, L.; Arruda, L.V.R.; Silva, M.H.; Ribas, P.C. Otimização das operações de transporte de derivados em um poliduto com múltiplas sangrias. In: Rio Oil & Gas Expo and Conference, 16, 2012, Rio de Janeiro.

Kira, G.; Lara, L.G.; Felizari, L.C.; Stebel, S.L.; Boschetto, S.N., Magatão, L.; Arruda, L.V.R.; Fabro, J.A.; Bettoni, L.M.M.; Ribas, P.C. Scheduling das operações de transporte de derivados de petróleo em um poliduto com múltiplas sangrias. *In: Rio Oil & Gas Expo and Conference*, 15, 2010, Rio de Janeiro.

Magatão, L., Arruda, L.V.R., Neves-Jr., F. A mixed integer programming approach for scheduling commodities in a pipeline. *Comp. and Chemical Eng.*, 28(1-2), 171-185, 2004.

Relvas, S.; Matos, H.A.; Barbosa-Póvoa, A.P.F.D.; Fialho, J. Oil products pipeline scheduling with tank farm inventory management, *Proceedings of 18th European Symposium on Computer Aided Process Engineering*, Vol. 25, p. 277-282, 2008.