

PETROLEUM ALLOCATION AT PETROBRAS: MATHEMATICAL MODEL AND A SOLUTION ALGORITHM

Roger Rocha^{*1}, Ignacio E. Grossmann² and Marcus V. S. Poggi de Aragão³

Logistic Division - Research and Development Center, PETROBRAS ¹
Av. Horácio de Macedo 950, Ilha do Fundão, Rio de Janeiro, Brazil

Dept. of Chemical Engineering - Carnegie Mellon University²
5000 Forbes Avenue, Pittsburgh, PA USA

Dept. of Informatics – Pontific Catholic University – PUC-Rio³
R. Marques de São Vicente 225, Rio de Janeiro, Brazil

Abstract

Petroleum Allocation is an important link for the integration of Petroleum Supply Chain at PETROBRAS as it is responsible for refining the strategic supply planning information to be used at the operation levels. In this work we describe how mathematical programming is being used to solve the Petroleum Allocation Problem and we show the effectiveness of a local search method by optimization to solve real industrial problems. We propose a Mixed-Integer Linear Programming formulation of the problem that relies on a time/space discretization network. As the model cannot be solved for the industrial size instances of the problem, and not even a feasible solution can be found after 15 days of computation, we implement an algorithm based on a heuristic to find a feasible solution and on a local search procedure based on optimization to improve it. Using this algorithm, solutions are found for all the case studies within 10% of optimality in less than 5 hours.

Keywords

Petroleum Supply Chain; Mixed-Integer Linear Programming; Local Branching Procedure.

* Corresponding author. Tel.: +55 21 3865 4505; fax: +55 21 3865 4565

Email address: rogerocha@petrobras.com.br (R. Rocha).

1. Introduction

Petroleum allocation involves programming crude oil from platforms to refineries on a daily basis. This must be done by taking into account strategic planning and operational constraints in the petroleum supply chain as follows. Crude oil can either be locally produced or imported from abroad. Local crude oil comes from production sites, mostly offshore, and is transported either by tankers or pipelines. Imported oil is only transported by tankers. After reaching maritime terminals, domestic crude oils are either exported, or shipped to PETROBRAS refineries. At the refineries, petroleum is processed in crude distillation units (CDUs) on daily scheduled production campaigns. These campaigns are defined by consumption rates of different petroleum categories, and start and finish dates to completing them.

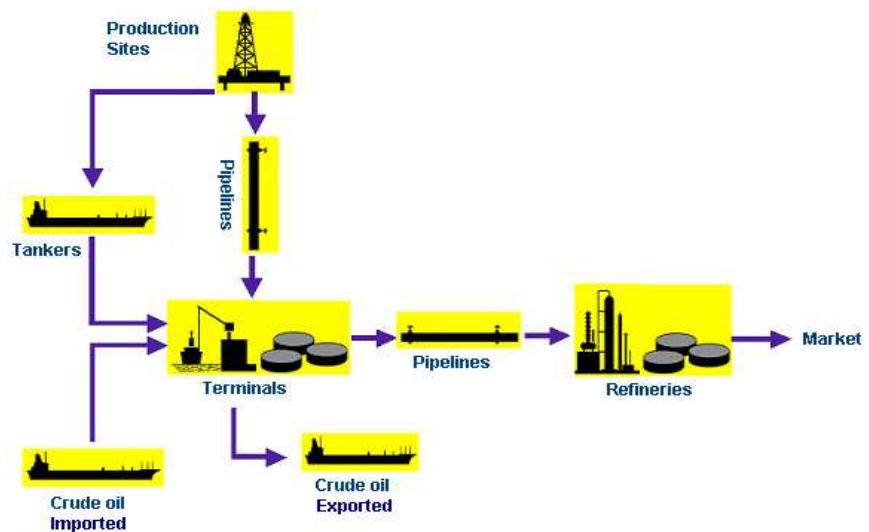


Figure 1. Infrastructure of PETROBRAS' Petroleum Supply Chain

The petroleum allocation activity at PETROBRAS represents a big challenge. Firstly, the size of the network, as PETROBRAS has assets spreading all over the territory of Brazil. Secondly, these operations have to be planned for an average horizon of 72 days. At present, these operations are

planned manually. Consequently, since a large number of simplifications is needed, this leads to suboptimal operations. Moreover, as some constraints need to be disregarded, the flow of information in the supply chain is not properly accounted for at the low levels.

Applications of mathematical programming in the petroleum industry date back from the 1950's with the work of Charnes et al (1952) and Sysmond (1955). Since then, we have seen an enormous advancement of algorithms and modeling techniques to solve problems related to the petroleum industry. Today we can say that the relevance of mathematical programming tools are common ground among all oil companies (Forrest and Oettly, 2003). However, most of the tools are focused on specific parts of the petroleum supply chain, often leading to a lack of integration. Lasschuit and Thijssen (2004) stress the importance of achieving a full integration in the oil and chemical supply chain and describe a tool developed by Shell Global solutions with this objective. Pinto and Neuro (2004) point out the significance for the oil industry to have a broader view of the supply chain and propose a general framework for modeling operations in the supply chain. Nonetheless, the offshore portion of the problem is not considered. The model presented in this paper is built with the requirement of integration in mind, and intends to close the gap between the strategic and operational levels at PETROBRAS.

We should mention that the problem addressed in this work is particular to the PETROBRAS logistic process, and to the best of our knowledge, no reference can be found in the literature that treats a similar problem in its whole extension. Typically, this problem is divided into two sub problems: inventory and ship scheduling (Miller, 1987; Brown, 1987), and planning operations at refineries (Lee et al, 1996; Pinto et al, 2000; Wenkay, 2003). In this study we have chosen to model the entire problem. Needless to say, if some simplifications are not performed, any real instance of this problem would remain out of reach.

We propose a Mixed-Integer Linear Programming model to solve this important problem for the integration of petroleum supply chain at PETROBRAS. This is a large scale problem, and despite all the efforts that were spent to tighten the formulation, not even a feasible solution was achieved after 15 days of computation. To solve this problem, an algorithm is proposed in this paper based on a heuristic to find a feasible solution, and on a local search procedure by optimization known as Local Branching. Using this algorithm, we can find for all the case studies considered a solution guaranteed to be within 10% of optimality in less than 5 hours. We compare our results against a standard solver (XPRESS-MP, 2007) supplying the heuristically generated upper bound. Our approach outperformed the standard solver in all instances.

This paper is organized as follows. In section 2, we present the problem and describe each element that is considered in the mathematical formulation. The mathematical model is introduced in section 3, along with an example solution, an extension of the formulation, and a family of cuts. The proposed solution algorithm is explained in section 4, while computational results supporting the solution algorithm are shown in section 5. Finally, in section 6 we draw some conclusions and discuss future work.

2. Problem Description

To describe the problem, first we need to understand the logistic process of petroleum supply to the refineries at PETROBRAS. This process follows the general hierarchical structure of a typical supply chain and is divided into three levels: Strategic, Tactical and Operational as shown in Figure 2.



Figure 2. Simplified Hierarchical structure of the Petroleum Supply Chain at PETROBRAS

At the strategic level, a Linear Programming tool, known as PLANAB, is used to determine on a monthly basis the amount and type of petroleum that will be processed in each refinery, as well as the amount of petroleum exported and imported for a horizon of three months. The Petroleum Allocation, which is a tactical activity, refines this information on a daily basis for a horizon of two and a half months considering in greater detail the constraints on the petroleum supply chain. This information will be further used for refinery planning and for the operational level activities.

In the following subsections we describe each element considered in our mathematical model, as well as some simplifications that are necessary.

2.1. Production Sites

A production site is one or more platforms that usually produce oil from the same petroleum field. They can be offshore or terrestrial. However, in the PETROBRAS case, almost 95% of them are offshore. Moreover, depending on the infrastructure installed, the petroleum can be shipped to

terminal via pipeline or tankers. Nonetheless, in Brazil, only three production sites are linked to terminals by pipelines. The petroleum that comes out from a given production site is commonly named after the offloading platform, as depicted in Figure 3. This allows us to use interchangeably the same index for the production site and for the crude oil produced by it.

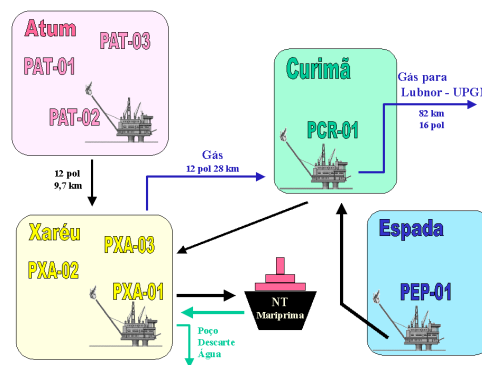


Figure 3. Schematic flow of the Xareu production site – Source: PETROBRAS

The data related to the production sites are those of the offloading platforms and can be summarized as follows:

- Production rate per day
- Shipment transportation mode – Tankers or pipelines
- Anchorage restrictions that are translated into the class of tankers that can be used for offloading them
- Storage capacity - It can be either the storage capacity of the offloading platform, or the auxiliary ship that is being used as a tank

2.2. *Tanker Fleet*

Regarding crude oil tankers, PETROBRAS owns or rents in time charter contracts most of its fleet. Crude oil tankers are used to transport petroleum from the production sites to the terminals, and occasionally if needed from terminals to terminals. They are usually classified according to their size and a common measurement unit to this purpose is the Dead Weight Tonnage (DWT). In this work, we consider the number of tankers in each class owned or rented by PETROBRAS, and we associate to each class an average transportation volume capacity and an average cost per day. Following are the classes and corresponding average capacities used in this study:

- Handy – C (19,000 m³)
- Handy – R (30,000 m³)
- Handy – L (40,000 m³)
- Panamax (65,000 m³)
- Aframax (100,000 m³)
- Suezmax. (140,000 m³)
- New-Suezmax (160,000 m³)
- VLCC (350,000 m³)

2.3. *Petroleum Category*

Petroleum category means a subset of the petroleum crudes with the same properties and equivalent product yields. It is introduced here for simplification as we manage inventory by category instead of by individual crude oil. This concept is used in inventory management in terminals as well as in refineries. In this work six petroleum categories are used, namely:

- Lubricant
- Light

- Cracked Atmospheric Residues, named here Rat-Craq
- Low Sulfur Content - LSC
- Marlim – a subset of petroleum produced in the Marlim field
- Asphaltic

2.4. *Maritime Terminals*

These are intermediate elements in the petroleum supply chain, where all petroleum from offloading platforms is sent to before reaching the refineries. They can be linked to one or more refineries by pipelines. This imposes some difficulties in the modeling, since we are not managing the inventory tank by tank. Therefore, it is necessary to pre-assign tanks to refineries, and to leave some volume corresponding to tanks that can be used either by one refinery or another for the sake of flexibility, as explained in Figure 4. In terms of modeling, the following data are considered:

- Number of berths
- Class of ships that can dock at the terminal – These data depend on the Dead Weight Tonnage (DWT) and the draft of ships, as well as the depth of sea and the structure of piers at the terminals.
- Storage capacity: by refinery connected to the terminal; by petroleum category and refinery; and total

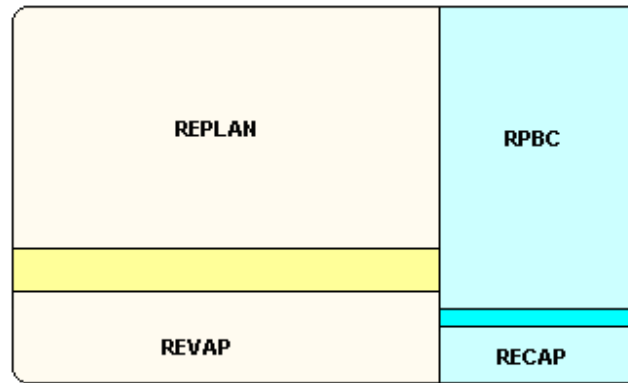


Figure 4 Schematic of the storage capacity assigned to the refineries connected to the São Sebastião terminal in São Paulo – The rectangles between refineries represent the tanks shared by them.

2.5. Pipelines

In this problem, pipelines are used to transport crude oil from platforms to terminals, and from terminals to refineries. We should mention that pipeline scheduling issues are out of the scope of this model, and basically we are only concerned with the estimated transfer time and daily maximum pumping rate. Regarding the pipelines that link platforms to terminals, transfer times are not considered, since they always handle the same crude oil.

2.6. Refineries

The refineries are the demand points of the problem considered, and need to be fed by the correct amount and type of petroleum in order to supply their market with the right products. We do not go into the detail of refinery modeling, furthermore we assume that the daily consumption rates by category of its crude distillation units are given. This information is usually obtained by analysis of the strategic planning directives. In summary, the following data are taken into account in our study for each refinery:

- Number of CDU
- Storage capacity by petroleum category and total
- Terminals pumping to it. Figure 5 shows the terminals that can pump petroleum to each refinery in the actual problem configuration.
- CDU's campaigns. These data are given by the starting time, the ending time and the daily consumption rate by category.

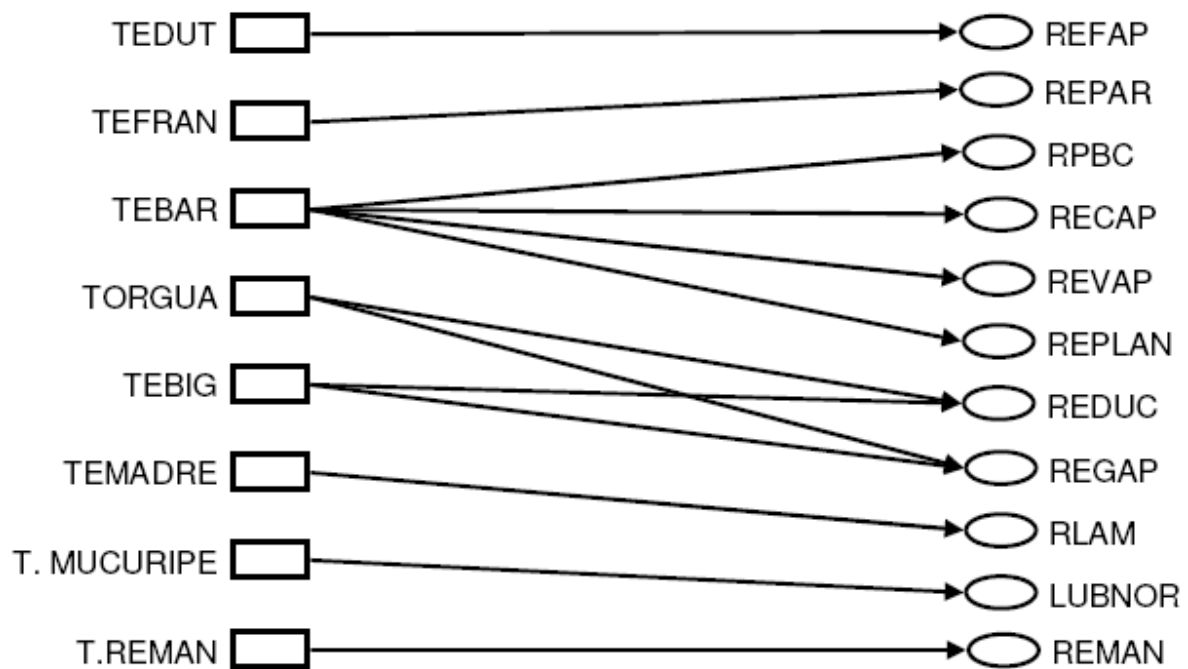


Figure 5 Connection between terminals and refineries

2.7. Problem Description

Given:

- A set of production platforms P
- A set of terminals T
- A set of Refineries R
- A set of tanker classes Cl

- (e) The daily crude oil production of each platform
- (f) The consumption campaign of each crude distillation unit in refineries
- (g) The logistic infrastructure restrictions, namely, storage capacity of production sites, terminals, and refineries; maximum pipeline flow rate between terminals and refineries; number and transportation capacity of tankers available in each tanker class; number of tankers that can operate simultaneously in each terminal.

Determine a minimum cost offloading scheduling for each platform in order to supply each refinery with the correct type and amount of oil indicated by the strategic planning.

The costs considered in the objective function are as follows:

- Tanker operational cost
- Deviation penalty from strategic planning;
- Penalty of petroleum shortage in refineries;
- Freight cost of additional tanker

3. Mathematical Model

This model is based on a fixed charge network flow structure over a discretized time representation (Wolsey, 1998). Time intervals of equal duration are considered and activities allocated to a given interval must be capable of being performed within it.

The nomenclature used in our model is as follows:

Indices

b: berth

c: crude oil category

cl: class of ship

$cp(u)$: campaigns of the crude distillation unit u

p : is used interchangeably to refer to crude oil or platform

r : refinery

u : crude distillation unit

t : time period

z : terminal

Parameters

$CAMP_{c,cp(u)}$: Consumption rate of category c in campaign $cp(u)$ at CDU u

$CAPL_{c,r}$, $CAPH_{c,r}$: Lower and higher ideal storage levels of category c at refinery r . Inventory below or above this ideal range is penalized.

$CAPT_{cl}$: Average transportation capacity of tanker class cl .

CF_{cl} : Freight cost of an additional tanker cl

$CP_{p,r}$: Penalty for deviation from strategic planning of crude oil p in refinery r .

$CRH_{c,r}$, $CRLl_{c,r}$, $CRI_{c,r}$: Penalty for having stock of category c at refinery r over interval t , high, low and infeasible, respectively.

CT_{cl} : Transportation cost of tanker cl per period t .

FU_{cl} : Fraction of the total number of tankers in class cl available to be used in a given time.

MSP_p : Maximum storage capacity at platform p .

MSZ_z : Maximum storage capacity at terminal z .

$MSZR_{z,r}$: Maximum storage capacity at terminal z by refinery connected to it.

NT_{cl} : Number of tankers in class cl .

$P_{p,t}$: Production of crude oil p over interval t .

$PLAN1_{p,r}, PLAN2_{p,r}$: Amount of crude oil p planned for refinery r for the first and second months, respectively.

$VT_{p,z}$: Voyage time between platform p and terminal z .

Binary variables

$bp_{p,b,cl,t}$: 1 if crude oil from platform p is sent to berth b by tanker cl over interval t ; 0 otherwise.

Continuous variables

df_{cl} : Number of tankers of class cl having to be freighted during the study horizon.

$dplan1_{p,r}, dplan2_{p,r}$: Deviation from strategic planning for crude oil p in refinery r , for the first and second months, respectively.

$pz_{p,c,z,r,t}$: Amount of crude oil p of category c that arrives at terminal z to supply refinery r over interval t .

$sp_{p,t}$: Amount of crude oil stored in platform p over interval t .

$strn_{c,r,t}, strh_{c,r,t}, strl_{c,r,t}, stri_{c,r,t}$: Stock of category c at refinery r over interval t , in the normal, high, low, and infeasible levels, respectively.

$stz_{c,z,t}$: Amount of category c stored at terminal z over interval t .

$zr_{c,z,r,t}$: Amount of category c pumped from terminal z to refinery r over interval t .

Constraints:

Inventory balance at the platforms for each time period t is given by,

$$sp_{p,t} - sp_{p,t-1} - P_{p,t} + \sum_{b,cl} CAPT_{cl} \cdot bp_{p,b,cl,t} = 0 \quad \forall p, t \quad (1)$$

The inventory at each platform p and time period t must be less than or equal to its maximum storage capacity,

$$sp_{p,t} \leq MSP_p \quad \forall p, t \quad (2)$$

At most one tanker should visit a given platform p at each time period t ,

$$\sum_{b,cl} bp_{p,b,cl,t} \leq 1 \quad \forall p, t \quad (3)$$

At most one tanker should arrive at a berth b at each time period t ,

$$\sum_{p,cl} bp_{p,b,cl,t} \leq 1 \quad \forall b, t \quad (4)$$

The amount of petroleum p that arrives at each terminal z in each time period t is expressed by,

$$\sum_{cl} CAPT_{cl} \cdot bp_{p,b,cl,t} - \sum_{r,c} pz_{p,c,z,r,t+VT_{p,z}} = 0 \quad \forall p, z, t \in [0, T - VT_{p,z}] \quad (5)$$

Notice that these constraints implicitly define the refineries to be supplied.

Inventory balance at the terminals for each time period t is given by,

$$stz_{c,z,r,t} - stz_{c,z,r,t-1} + zr_{c,z,r,t-1} - \sum_p pz_{p,c,z,r,t-1} = 0 \quad \forall c, z, r, t \quad (6)$$

Observe that the equation (6) is for each refinery r and petroleum category c . Thus, we need additional inequalities to account for the storage limit by refinery and for the storage capacity in the terminal. These inequalities are written as follows,

$$\sum_c stz_{c,z,r,t} \leq MSZR_{z,r} \quad \forall z, r, t \quad (7)$$

and,

$$\sum_{c,r} stz_{c,z,r,t} \leq MSZ_z \quad \forall z, t \quad (8)$$

Figure 6 illustrates the range of each variable as well as how the inventory is modeled. The middle portion of the figure represents the ideal inventory of a given category in the refinery. The upper portion corresponds to high inventory and, therefore, is penalized, since it can give rise to logistical problems in the refinery. The low inventory is depicted by the lower portion and, in the same way, is penalized because the refinery may need to shut down some units. The variables $stri_{c,r,t}$ are associated with the highest penalty, and have an interesting interpretation as they indicate to the specialist the need to import more oil to supply this particular refinery. The equations describing the inventory balance are then as follows,

$$\begin{aligned} & strn_{c,r,t} + strh_{c,r,t} - strl_{c,r,t} - stri_{c,r,t} - strn_{c,r,t-1} - strh_{c,r,t-1} + strl_{c,r,t-1} + \\ & + stri_{c,r,t-1} + zr_{c,z,r,t-1} - CAMP_{cp(u),c,t} = 0 \quad \forall c, z, r, t \end{aligned} \quad (9)$$

In addition, there are bounds on the stock variables,

$$CAPL_{c,r} \leq strn_{c,r,t} \leq CAPH_{c,r} \quad \forall c, r, t \quad (10)$$

$$strl_{c,r,t} \leq CAPL_{c,r} \quad \forall c, r, t \quad (11)$$

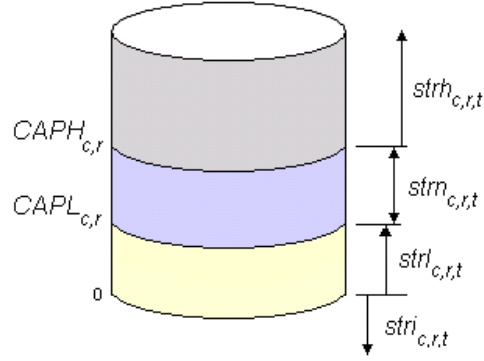


Figure 6. Schematic representation of the inventory in a given refinery

Solution deviation from the strategic planning is written for each month as follows,

First month,

$$dplan1_{p,r} + \sum_{c,z,t} pz_{p,c,z,r,t} \geq PLAN1_{p,r} \quad \forall p, r \quad (12a)$$

$$dplan1_{p,r} - \sum_{c,z,t} pz_{p,c,z,r,t} \geq -PLAN1_{p,r} \quad \forall p, r \quad (12b)$$

Second month,

$$dplan2_{p,r} + \sum_{c,z,t} pz_{p,c,z,r,t} \geq PLAN2_{p,r} \quad \forall p, r \quad (13a)$$

$$dplan2_{p,r} - \sum_{c,z,t} pz_{p,c,z,r,t} \geq -PLAN2_{p,r} \quad \forall p, r \quad (13b)$$

The maximum number of additional tankers required during the time horizon for each tanker class t is estimated by,

$$df_{cl} \geq [FU_{cl} \cdot NCL_{cl}] - \sum_{p,b} bp_{p,b,cl,t} \quad \forall cl, t \quad (14)$$

It is important to point out that the ship routing is not being considered in this study. However, we use the parameter FU_{cl} to have an estimate on the number of tankers available at each time period. In

this study, we set $FU_{cl} = 0.50$, representing that we have only half of the tankers available in each tanker class as we are considering that the other half is already busy transporting crude oil to a terminal.

The model attempts to minimize the total cost, which involves shipping costs, inventory costs, penalty for deviation from the strategic planning, and freight cost for additional tanker,

$$\min \left\{ \sum_{p,b,cl,t} CT_{cl} \cdot bp_{p,b,cl,t} + \sum_{c,r,t} CRH_{c,r} \cdot strh_{c,r,t} + \sum_{c,r,t} CRL_{c,r} \cdot strl_{c,r,t} + \right. \\ \left. + \sum_{c,r,t} CRI_{c,r} \cdot stri_{c,r,t} + \sum_{cl} CF_{cl} \cdot df_{cl} \right\} \quad (15)$$

3.1. Solution Example

We present in this subsection part of the solution of a test instance to give some insights about the decisions that the model handles. Figure 7 summarizes the offload scheduling at platform P1. As we can see, the important questions confronted at the platform offload scheduling are to avoid that the inventory reaches its maximum capacity with subsequent platform shutdown, and prevent sending a ship before having sufficient inventory to fill it up completely. Additionally, each ship assigned to offload each platform has its terminal destination determined. The picture in the terminals is slightly complicated, since for each discharge of a ship the model has to determine how its volume will be split between possible refinery and category. In the case shown in Figure 8, the terminal T1 is only linked to one refinery and therefore the representation becomes easier as the model only needs to manage the classification into category. When it comes to the refineries, the most relevant issue is to

keep the inventory inside the safe region delimited by the parallel lines in Figure 9. As pointed out in the modeling subsection every time the inventory goes beyond these limits the solution is penalized.

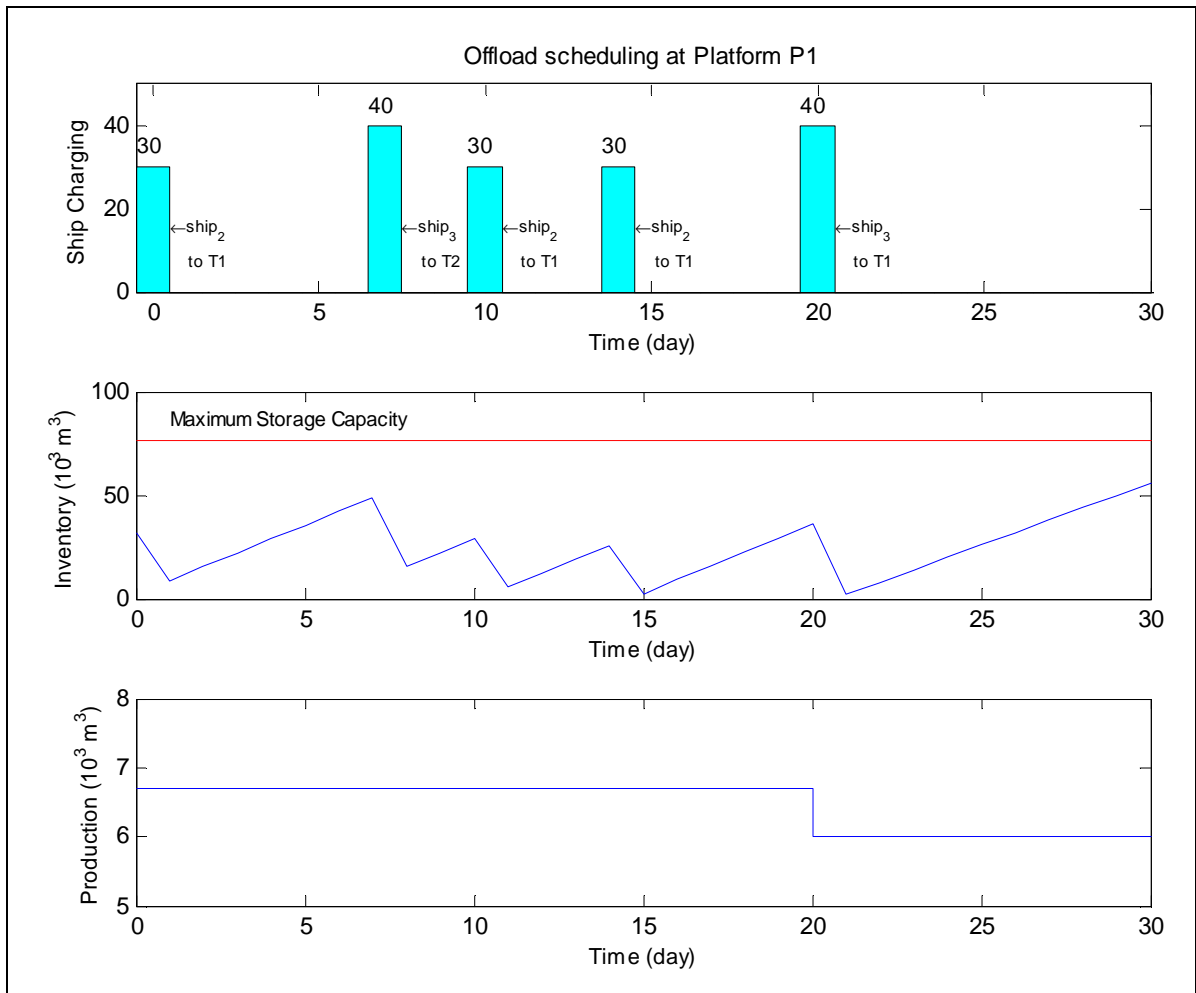


Figure 7. Summary of offload scheduling at Platform P1

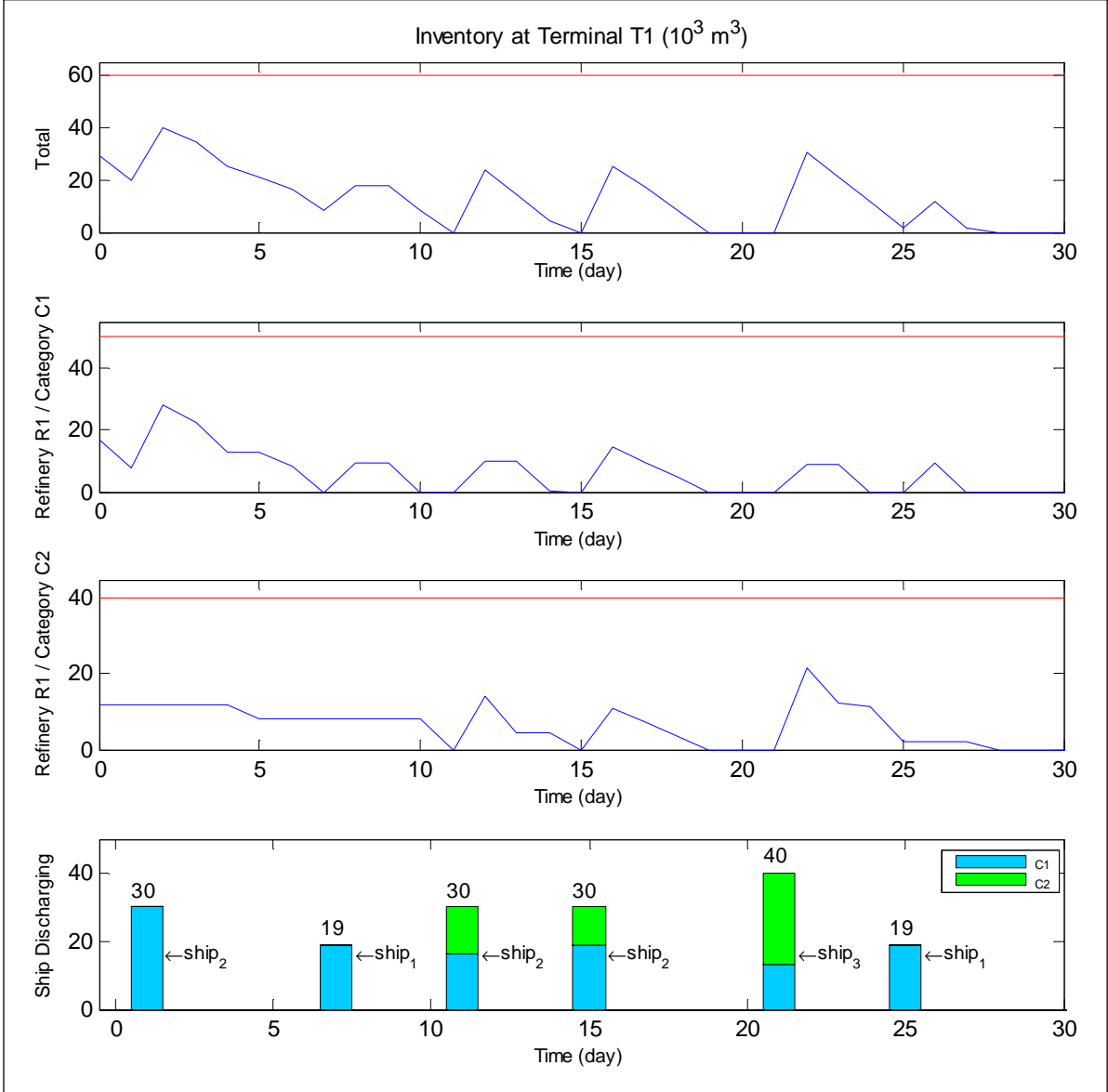


Figure 8. Summary of the inventory and discharging activity at Terminal T1

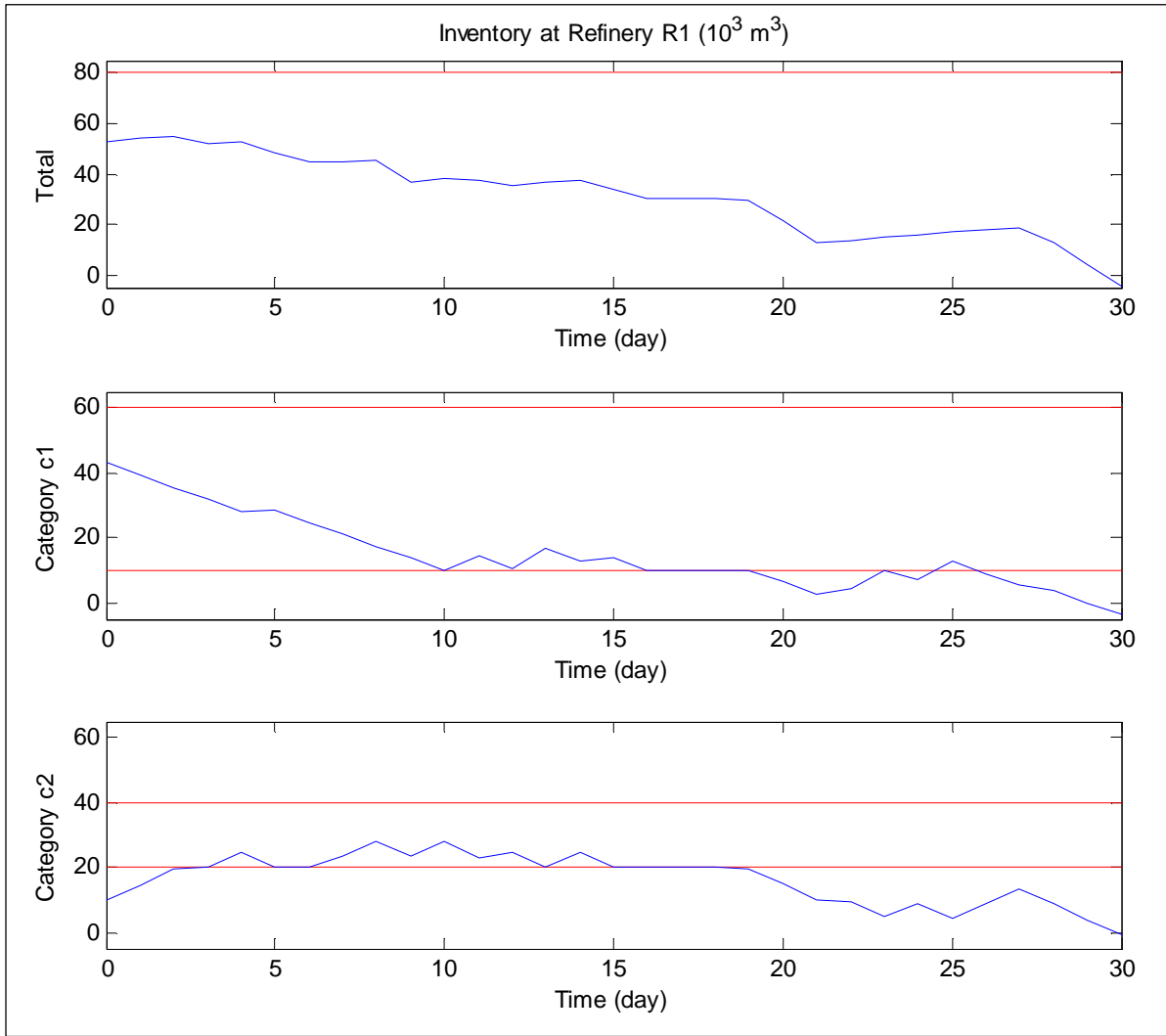


Figure 9. Summary of the inventory levels at Refinery R1

3.2. Flexibility of campaigns

In this model we study the effects of considering flexible dates for the initial campaigns defined by the specialist. In addition to the initial campaign given, we ask the user to inform the earliest, the latest, and the duration of each campaign. If it is worth to change the campaign dates, the model will perform it and can even break up the original campaign into the earliest and latest dates, on condition

that the crude oil estimate profiles match better with the refineries consumption. It should be noticed that every time the model divides a campaign, a changeover cost has to be paid. Hence, this is carried out only if it is absolutely necessary. Figure 10 illustrates how the assignment of campaigns works. As we can notice, the campaign 1 is initially set up to start on day 1 and finish on day 6. Nevertheless, the model could for instance move it to start on day 4 and finish on day 9. In the same way, the campaign 2 is originally programmed to start on day 7 and finish on day 15. However, the model could split it into two campaigns, one starting on day 1 and finishing on day 3, and another taking place from day 10 to 15.

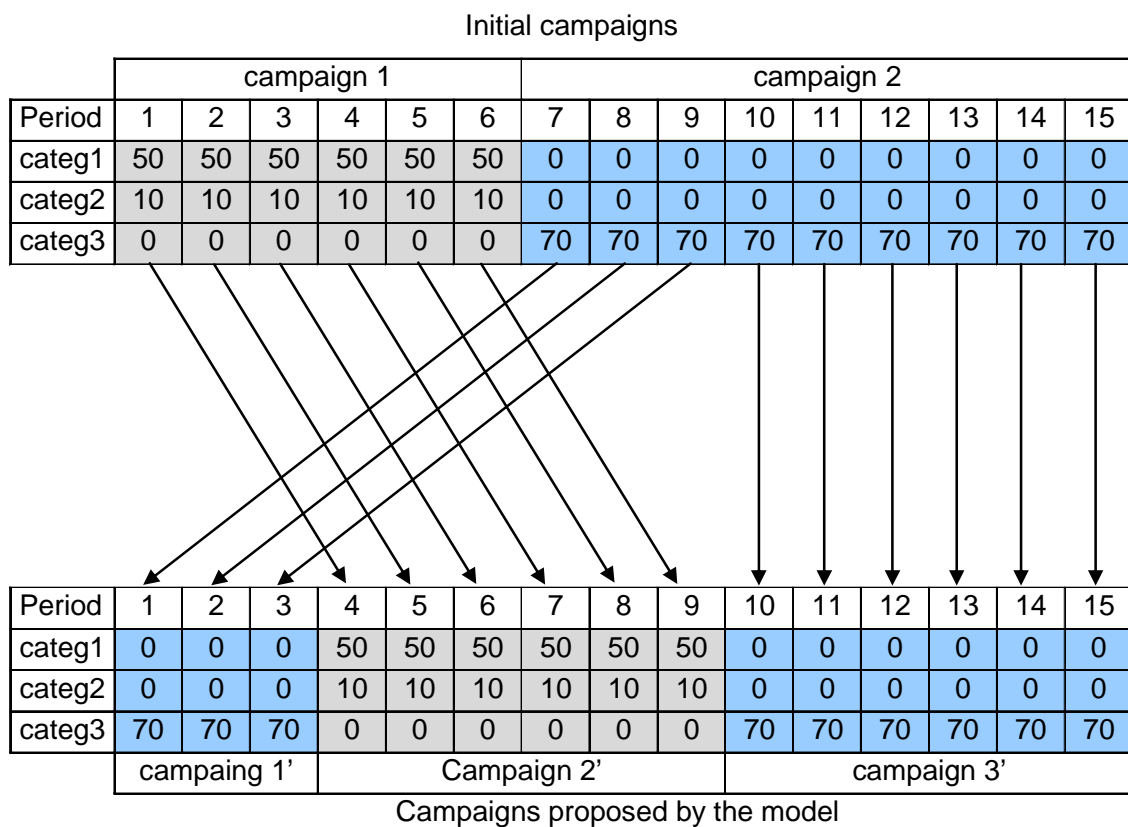


Figure 10. Flexibility of campaigns at refineries

In order to model the flexibility of campaigns, we need the following additional definitions:

Parameters

CS : Set up cost for CDU campaign changes

$DC_{cp(u)}$: Duration of campaign $cp(u)$ at CDU u .

$TD_{cp(u)}$: Deadline for completing campaign $cp(u)$ at CDU u

$TS_{cp(u)}$: Release date for campaign $cp(u)$ at CDU u .

Binary variables

$bcp_{u,cp(u),t}$: 1 if CDU u processes campaign $cp(u)$ at CDU u over interval t ; 0 otherwise.

$bsu_{u,t}$: 1 if a set-up is necessary in CDU u at time t .

The inequalities describing the flexibility of campaigns are as follows:

Assignment of production campaign to time slots within valid time windows,

$$\sum_{cp(u)} bcp_{u,cp(u),t} = 1 \quad \forall u, TS_{cp(u)} \leq t \leq TD_{cp(u)} \quad (16a)$$

The duration of each campaign must hold,

$$\sum_{t=TS_{cp(u)}}^{TD_{cp(u)}} bcp_{u,cp(u),t} = DC_{cp(u)} \quad \forall u, cp(u) \quad (16b)$$

If a change of campaign takes place from time period t to $t+1$, then a crude distillation set-up is necessary,

$$bsu_{u,t} + bcp_{u,cp(u),t+1} - bcp_{u,cp(u),t} \geq 0 \quad \forall u, t \quad (17)$$

Moreover, we incorporate the changeover costs, $\sum_u CS \cdot bcp_{u,cp(u)}$, to the objective function given in

(15).

Figures 11-12 show the advantage of flexible campaigns. We observe that when the flexibility of campaigns is considered we avoid in some time periods stockout of some petroleum categories in the refineries.

This concept of flexible campaigns is not considered yet in practice in the Petroleum Allocation activity. However, we introduce it in this model to draw attention to the importance of considering this aspect in the problem, since as verified by Figures 11-12, it allows us to manage our resources in a more economic way.

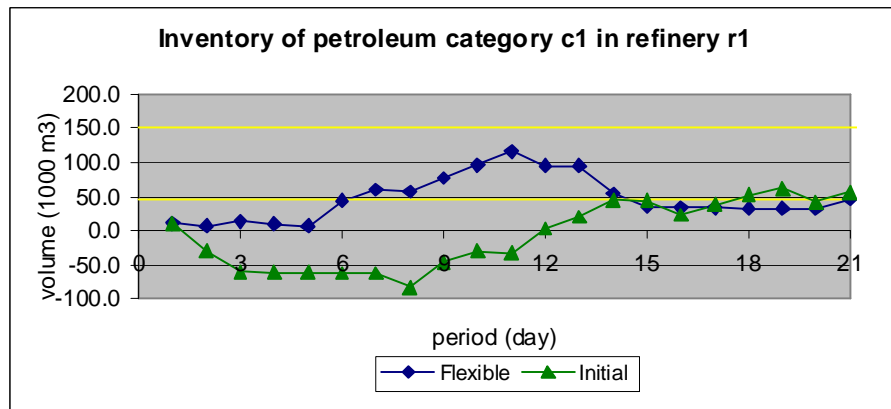


Figure 11. Comparison between inventory evolution of category c1 at refinery r1 using initial and flexible campaigns

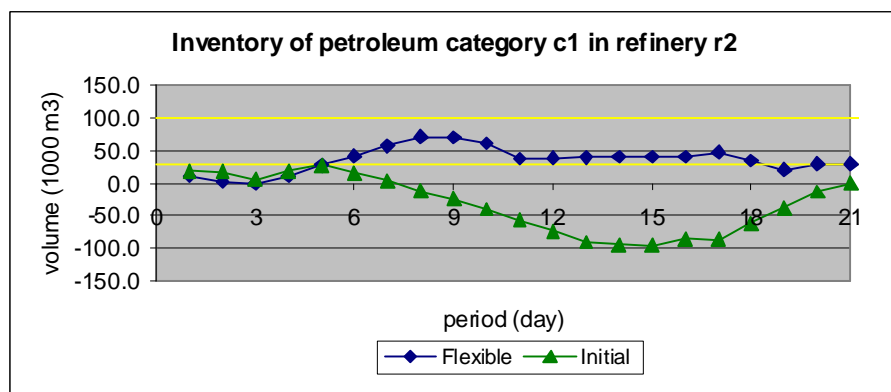


Figure 12. Comparison between inventory evolution of category c1 at refinery r2 using initial and flexible campaigns

3.2. Changeover cut

As shown by Yee and Shah (1998), the presence of changeovers in an MILP scheduling model may lead to a large relaxation gap. To overcome this difficulty, usually some cut constraints are added to enforce that a minimum number of changeover tasks must be performed. It is easy to show that scheduling problems with changeover costs present a large integrality gap that increases the computation burden in branch and bound algorithms. The example in Figure 13 motivated this discussion and shed some light on the possible outcome when this aspect is disregarded.

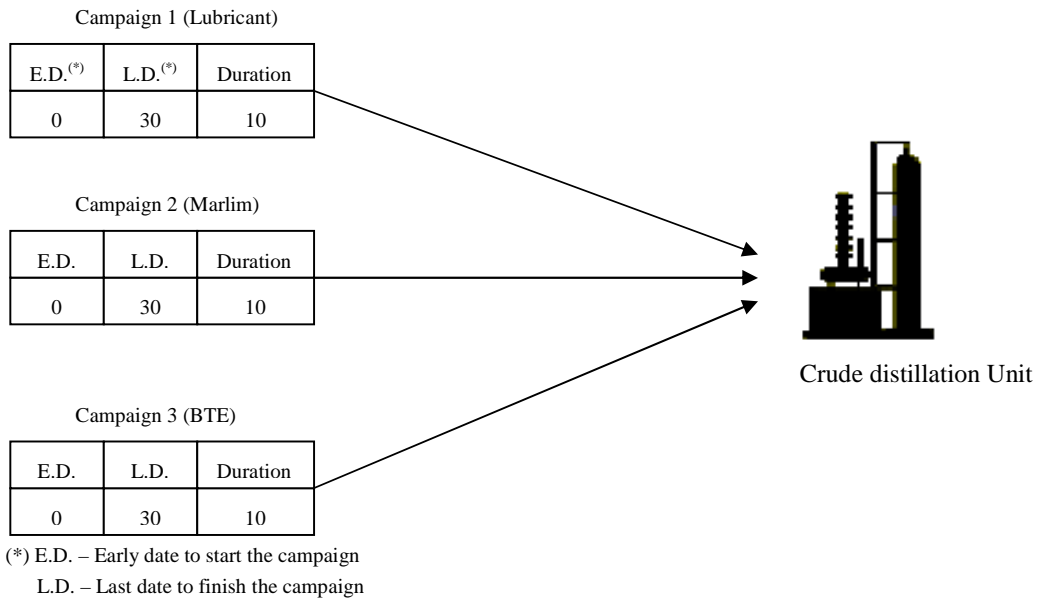


Figure 13. Motivated example to justify the changeover cuts

It is likely that the LP relaxation solution to this example happens to be,

$$bcp_{u, cp(u), t} = \frac{1}{3} \quad \forall cp(u), t$$

Thus from (16),

$$bsu_{u,t} = 0 \quad \forall t$$

Therefore, the changeover costs will be zero in the objective function. However, we can verify by inspection that at least two changeovers will be necessary, as three campaigns were initially assigned to this crude distillation unit. We can generalize this idea and write the following cuts for each crude distillation unit,

$$\sum_t bsu_{u,t} \geq |NCP_u| - 1 \quad \forall u \quad (18)$$

where, $|NCP_u|$, is the number of campaigns in CDU u .

To demonstrate the importance of adding changeover cuts to our formulation, in the sequel we present some computational results on three instances whose sizes are detailed in Table 1. All the results were obtained by setting the integrality tolerance gap to 10%.

Table 1 – Dimension of test instances

Elements	Instances		
	#1	#2	#3
Production sites	6	6	11
Terminals	4	4	5
Berths	6	6	7
Refineries	5	5	6
CDUs	6	6	7
Crude oil categories	3	3	3
Ship classes	3	3	6
Horizon (days)	10	10	60

Table 2 – Computational results for solving instance 1 using changeover cuts.

Instance 1	Without changeover cuts	With changeover cuts
no. of constraints	1550	1553
no. of variables	2740	2740
no. of binary variables	940	940
no. of visited nodes	>> 1000000	202
CPU(s)	>> 100000	3
LP relaxation solution (Z_{pl})	321,40	464,73
Best solution (Z_o)	-	605,50
Initial integrality gap *	47 %	23 %

$$\left(\frac{Z_* - Z_{pl}}{Z_*} \right) \times 100\%, Z_* = \min\{ Z_o \}$$

Table 3 – Computational results for solving instance 2 using changeover cuts.

Instance 2	Without changeover cuts	With changeover cuts
no. of constraints	1550	1553
no. of variables	2740	2740
no. of binary variables	940	940
no. of visited nodes	551	313
CPU(s)	4	3
LP relaxation solution (Z_{pl})	539,00	832,00
Best solution (Z_o)	1111,00	1104,50
Initial integrality gap	51 %	25 %

Table 4 – Computational results for solving instance 3 using changeover cuts.

Instance 3	Without changeover cuts	With changeover cuts
no. of constraints	13847	13855
no. of variables	46992	46992
no. of binary variables	22123	22123
no. of visited nodes	>> 1000000	631243
CPU(s)	>> 100000	32760
LP relaxation solution (Z_{pl})	16484,03	17471,83
Best solution (Z_o)	-	25743,72
Initial integrality gap	35 %	32 %

Although these test instances do not represent the complexity of the real problem, the results in Tables 2-4 make clear the strength of the cuts in (18). After adding the changeover cuts, the integrality gap was less than half its value compared to the original formulation for instance tests 1 and 2. For instance 3, although the decrease in the integrality gap was smaller, the changeover cuts made it possible to solve this very large problem. Moreover, the number of changeover cuts introduced is negligible compared to the size of the original formulation, and thus the time to solve the Linear Relaxation at each node of the Branch and Bound is not affected.

4. Solution Algorithm

As one can see from Tables 2-4, the changeover cuts substantially improve the formulation of the problem and allow solving test instances that could not be solved by the initial model. However, as shown by the instance 3, which required about 9 hours of CPU-time, as the problem becomes larger the computational time grows rapidly, making prohibitive the use of this formulation for solving real instances in reasonable time. To circumvent this problem, we propose an algorithm based on a

heuristic to find a feasible solution, and on an MIP heuristic known in the literature as Local Branching (Fischetti and Lodi, 2003). The basic steps of the algorithm are as follows.

1. *Heuristic. Determine the tank classes and time periods to offload each platform.*
2. *Solve the entire problem with variables fixed according to the heuristic.*
3. *Apply Local Branching to the original problem with the initial solution found by step 2. The output is the best solution after 5 hours.*

In the following subsection, we describe in detail the heuristic and the Local Branching procedure.

4.1. *Heuristic*

We verified that the part of the model that gives rise to a weak linear relaxation is the platform offloading sub-problem, since the transportation cost per volume decreases as the ship capacities increase. The linear relaxation tends to use a fraction of a bigger available ship instead of trying a smaller ship or an entirely bigger ship. Therefore, we propose an algorithm that fixes the time and the ship class that will be used to offload a given platform. A description of the algorithm is as follows:

- 1- *Define OL_p as the set containing all pairs of ship and time to offload platform p ;*
- 2- *Construct a list L of platforms by sorting them in decreasing order of their total production;*
- 3- *for each p in L do*
 - $OL_p \leftarrow \text{empty};$*
 - $stock = \text{initial_stock}$*
 - $t=0$*

while $t \leq T$ do

pick out a ship class cl to offload p ; (*)

calculate the next time, t_{next} , when there is enough stock to entirely load ship cl ;

$OL_p \leftarrow (cl, t_{next})$;

set $t = t_{next}$;

end-while

end-for

4- fix all variables such that the indices cl and t are not in OL_p to zero;

5- solve_problem(P);

Note that in (*), the choice is made by spinning a roulette where the probability of each ship class is calculated by the number of route from p with ship cl over the total number of route from platform p . Figure 14 illustrates how the heuristic works.

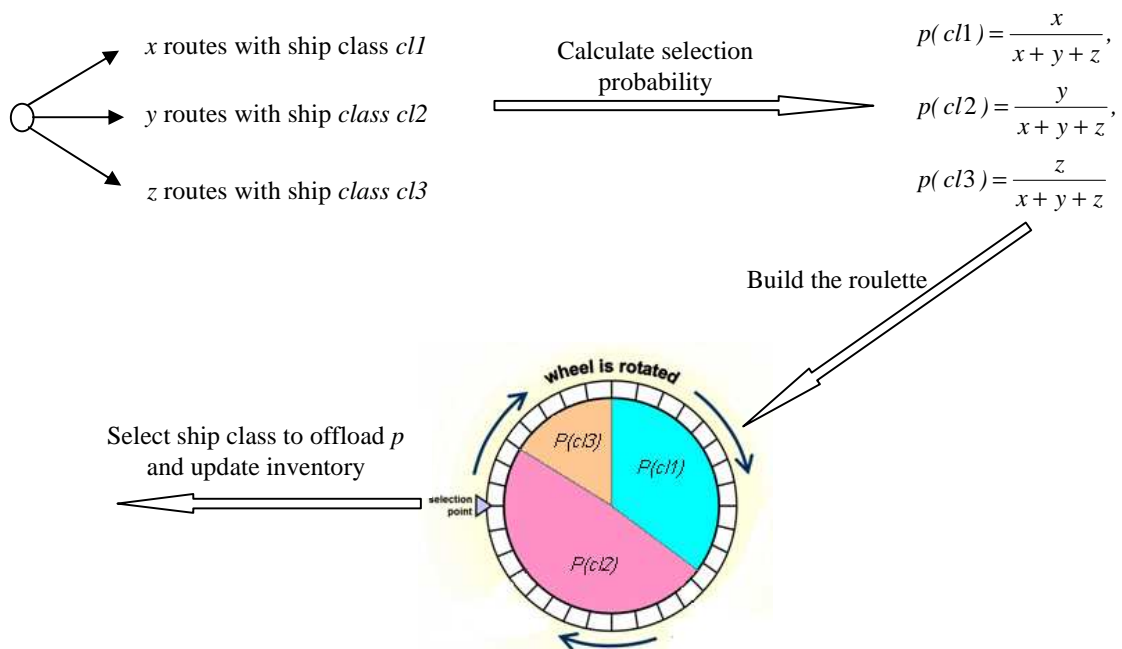


Figure 14. Representation of the heuristic algorithm

4.2. Local Branching

The Local Branching procedure was first proposed by Fischetti and Lodi (2003) with the aim of improving the efficiency of an exact algorithm like Branch and Bound. It consists of reducing the solution search space by introducing to the model some inequalities called “Local Branching cuts”, i.e., given a new incumbent solution \bar{x} and a nonnegative integer parameter k (the neighborhood size), the space solution is reduced by means of the following inequalities:

$$\Delta(x, \bar{x}) = \sum_{j \in S} (1 - \bar{x}_j) + \sum_{j \in B, \bar{S}} \bar{x}_j \leq k \quad (19)$$

where, B is the set of binary variables and \bar{S} is the index set of binary variables that take value of one in the solution \bar{x} .

The basic idea of Local Branching is to optimize over a small neighborhood, given by inequality (19), in an attempt for finding better solutions faster. It should be noticed that the Local Branching cuts embody the idea of a neighborhood search, similar to the idea present in most of metaheuristics, as they express that the solution at hand can be changed by at most k elements. One parameter of this procedure that needs to be determined is the neighborhood size k . There is a trade-off in choosing the value for k , as we want it to be as small as possible to make the problem easy to solve, but large enough to contain solutions better than the given one. Fischetti and Lodi suggested that a value of k in the interval $[10, 20]$ works best in the majority of cases. Besides, in the implementation of the procedure this parameter can be dynamically modified as the algorithm proceeds.

In this work, we implement a version of the Local Branching, called asymmetric, that counts only the number of variables changing from 1 to 0, as given by the expression (20). Furthermore, only the platform offloading variables are considered to generate them, leaving more room to the solver for finding a better solution.

$$\sum_{j \in S} (1 - \bar{x}_j) \leq k' \quad (20)$$

where, $k' \equiv \lceil k/2 \rceil$

We can better understand how this procedure works by its basic tree representation as given in Figure 15. The upshot of this method is that we build a search tree where only the nodes appended by triangles with an S (for Solver) are explored by an off-the-shelf MIP solver. We start out with an initial feasible solution \bar{x}_1 , and we add to the model the Local Branching constraint $\Delta(x, \bar{x}_1) \leq k$ that correspond to the node 2 in the tree. We solve the problem represented by node 2, and find the optimum solution \bar{x}_2 which is better than \bar{x}_1 . At this point, we delete from our model the constraint $\Delta(x, \bar{x}_1) \leq k$, adding simultaneously the constraints $\Delta(x, \bar{x}_1) \geq k+1$ and $\Delta(x, \bar{x}_2) \leq k$ (these steps lead us to node 4). We keep on doing these basic steps until we reach a point where the problem is infeasible (node 6). In this case, we simply delete the last constraint $\Delta(x, \bar{x}_6) \leq k$ and add $\Delta(x, \bar{x}_6) \geq k+1$, abandoning the Local Branching procedure and proceed solving the problem represented by node 7 using an MIP solver. This is the most basic description of the implementation of the Local Branching procedure. A more elaborated algorithm can be found in Fischetti and Lodi (2003), where they borrowed some ideas from the metaheuristic community to improve the efficiency of this implementation. We implemented this algorithm in C++ using the modeling library provided by Dash Optimization (XPRESS-BCL, 2006) to add and remove constraints dynamically from the model.

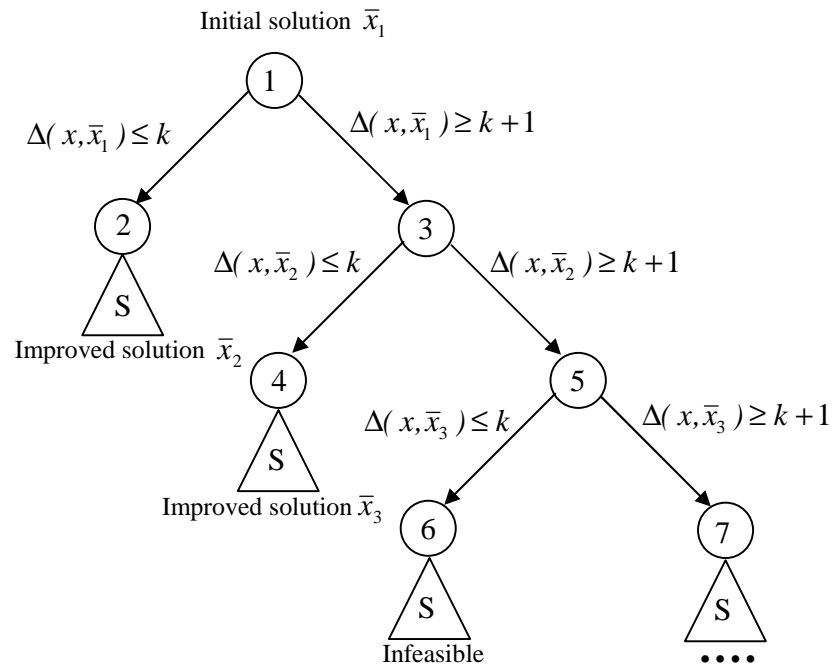


Figure 15 . Basic Tree representation of the Local Branching procedure

In summary, the key points of this method are:

- The search is concentrated on a reduced part of the search space defined by the left nodes in the tree in Figure 15.
- The neighborhood of a given solution is explored with an MIP solver.

5. Results

Experiments were conducted on a Pentium IV 2.0GHz 1.0Gb RAM PC and the code was compiled with Visual C++ 6.0 under a Windows Platform. Our algorithm was tested on real instances of the problem involving 42 platforms/crude oils, 6 tanker classes, 8 maritime terminals involving 13 berths, 11 refineries, and 20 distillation units over a time horizon of 72 discretized intervals. The typical instance gives rise to a model with approximately 40,000 binary variables, 45,000 continuous variables, and 32,000 constraints. Tables 5 and 6 show the results for 4 different real instances of the

problem where the efficiency of Local Branching and the quality of solution for the minimization problem are reported. In all instances the heuristic solution were found in less than 3 minutes. We used a value $k' = 6$ for the local branching cuts, and set a time limit of 5 hours for XPRESS and Local Branching referred to as LBr in Tables 5 and 6.

Table 5. Comparison of best solution

Instance	Heuristic solution	XPRESS	LBr
4	68021.2	68021.2	62939.8
5	13883.6	12561.6	10424.8
6	19231.2	19231.2	16760.4
7	223522.0	223522.0	207519.0

Table 6. Quality of Local Branching solutions

Instance	Best Lower Bound (Z_{LB})	LBr (Z^*)	Gap* (%)
4	57435.4	62939.8	9.6
5	9532.6	10424.8	9.4
6	15240.1	16760.4	10.0
7	193814.3	207519.0	7.1

(*) $Gap = 100.(Z^* - Z_{LB})/Z_{LB}$

The performance of the Local Branching procedure was quite impressive as shown in Tables 5-6. As we can observe in Table 5, XPRESS could not improve the initial solution in 3 out of 4 instances and the only one that was possible, the result was only marginally better. On the other hand, the Local Branching procedure was able to considerably reduce the cost of the initial solution for all instances, with average improvement of 13%. The results in Table 5 clearly show the significance of optimizing a large scale problem over a small region in order to obtain good solutions faster. Furthermore, Table 6 shows that the solutions obtained by the Local Branching procedure are within 10% of the optimum, even though this is not guaranteed by the procedure.

6. Conclusions

In this paper we have presented a Mixed-Integer Linear formulation to a problem faced by an integrated petroleum company. The problem is of considerable importance in the petroleum supply chain as it provides a link between strategic and operational decisions. It is a large scale problem and even finding a feasible solution has not been possible using commercial solvers.

An extension of the initial problem was proposed, namely the flexibility of campaigns, and we showed its benefits in terms of the solution quality. Additionally, we presented some cuts that allow curbing the poor relaxation introduced by the changeover costs. To overcome the computational expense for finding a feasible solution, we designed a heuristic to schedule the offloading of platforms, greatly reducing the search space, and allowing the solver to find a feasible solution rapidly. In addition, we showed that the Local Branching procedure is an important tool to find good solutions for large scale problems.

Future work will try to exploit the fact that the problem becomes easier to solve when the offloading variables are fixed. Another alternative is to use some decomposition technique, as the problem can be naturally split up into a platform offloading problem and a refinery planning problem.

References

- Brown G; Graves W; & Ronen D. (1987). Scheduling ocean transportation of crude oil. *Management Science*, 33, 335–346.
- Charnes, A.; Cooper, W. W., and Mellon, B. (1952). Blending Aviation Gasolines, *Econometrica*, 22, 135-159
- Fischetti, M. & Lodi, A. (2003). Local Branching, *Mathematical Programming*, 98, 439-467
- Forrest, J. & Oettli, M. (2003). Rigorous simulation supports accurate refinery decisions. In I. E. Grossmann and C. M. MacDonald (Eds.). *Proceedings of fourth international conference on foundations of computer-aided process operations* (pp. 273-280), Coral Springs, CACHÉ.
- Lasschuit, W.; Thijssen, N. (2004). Supporting supply chain planning and scheduling decisions in the oil and chemical industry, *Computers and Chemical Engineering*, 28, 863-870.
- Lee, H.; Pinto J. M.; Grossmann, I. E. & Park, S. (1996). Mixed-integer Linear Programming Model for Refinery Short-term Scheduling of Crude Oil Unloading with Inventory Management , *Industrial & Engineering Chemistry Research*, 35, 1630-1641
- Mehring, J. S., & Gutterman, M. M. (1990). Supply and Distribution Planning Support for Amoco(U.K.) Limited , *Interfaces*, 20, 95-104
- Miller, D. M. (1987). An Iterative, Computer-aided Ship Scheduling System, *European Journal of Operational Research*, 32, 363-379
- Pinto, J. M.; Joly, M. & Moro, L. F. L. (2000). Planning and Scheduling models for refinery operations, *Computers and Chemical Engineering*, 24, 2259-2276
- Pinto, J. M.; & Neiro, S. M. S. (2004). A general modeling framework for the operational planning of petroleum supply chains, *Computers and Chemical Engineering*, 28, 871-896

Sysmonds, G. H. (1955). *Linear Programming: The Solution of Refinery Problems*. New York: Esso Standard Oil Company

Wenkai, L.; Hui, Chi-Wai; Hua B. & Tong, Z. (2003). Scheduling Crude Oil Unloading, Storage and Processing, *Industrial and Engineering Chemistry Research*, 41, 6723-6734

Wolsey, L. A. (1998). *Integer Programming*, John Wiley and sons, New York.

XPRESS-BCL (2006) Reference Manual, Dash optimization, release 3.0.

XPRESS-MP (2007) Optimizer Reference Manual, Dash Optimization, release 18.

Yee, K. L. & Shah N. (1998). Improving the efficiency of discrete time scheduling formulation, *Computers and Chemical Engineering*, 22, 403-410.