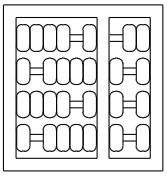

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Planning The Operation of a Large Real-World Oil Pipeline

Tony Minoru Tamura Lopes
Arnaldo Vieira Moura *Cid Carvalho de Souza*

Technical Report - IC-10-14 - Relatório Técnico

May - 2010 - Maio

Planning The Operation of a Large Real-World Oil Pipeline

Tony Minoru Tamura Lopes * Arnaldo Vieira Moura †
Cid Carvalho de Souza ‡

Abstract

A set of oil derivative distribution depots, including refineries and terminals, have local demands and productions for different products in a given time horizon. However, in a certain period there may be not enough local stock of some product to satisfy the corresponding demand. This brings the need for transportation of oil derivatives through a network of pipelines. To accomplish that, a tactical pumping plan is composed monthly, and a more detailed operational schedule, spanning a few days, must be updated daily. In real-world applications, both the planning and the scheduling must satisfy a large set of operation constraints. This work defines the tactical planning problem and proposes a novel network flow model to solve it. Also, a procedure is given to decompose the solution into a specific input format, as needed by another solver that computes the final, detailed, daily scheduling solution. Our model treats the oil pipeline network that is operated by the Brazilian oil company Petrobras. This is one of the most complex and large topologies when compared to other networks treated in the open literature. The model was tested with real-world instances and showed significant improvements over human planning.

1 Introduction

The Oil Industry faces many difficult logistic problems that must deal with unstable markets and large amount of resources. In this context, planning problems are amongst the most important and have received intense attention [8, 15, 11, 10, 16].

This paper focus the inland oil derivatives distribution problem stemming from of the Brazilian oil company Petrobras. Being the 15th largest oil company in the world[1], it faces a very difficult transportation problem in which ethanol and several petroleum derivatives, like gasoline, diesel, and naphtha, must be transported from refineries to depots, where consumer markets are located. Pipeline networks offer the most economical way available to transport oil derivative products inland. The scenario studied here has an extension of 7,000 kilometers, comprising 29 individual interconnecting pipelines. There are 14 distribution

*Institute of Computing - University of Campinas. This research was supported by grant 05/57343-0 from FAPESP.

†Institute of Computing - University of Campinas
‡Institute of Computing - University of Campinas

[1]See `www.energyintel.com`.

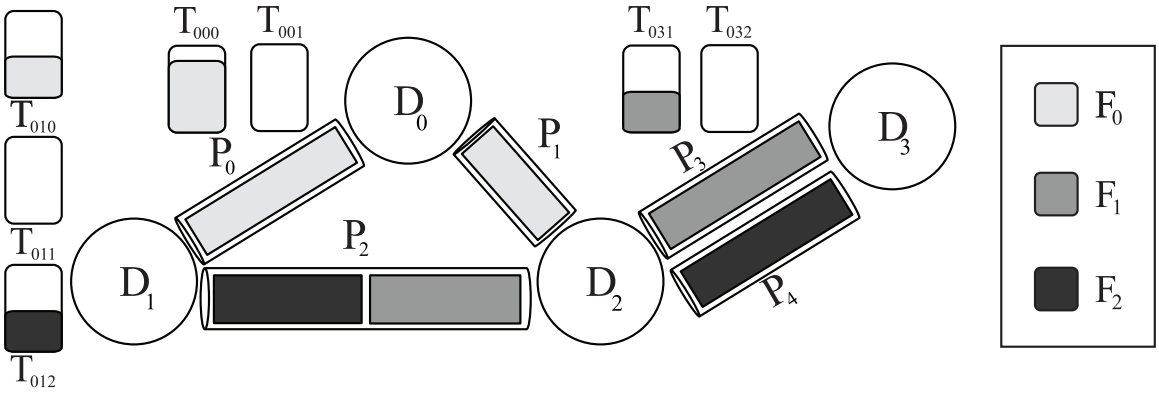

Figure 1: A Sample Pipeline Network.

depots that can store up to 10 millions cubic meters of products, stocked in more than 200 tanks located at various depots. Figure 1 depicts the network topology.

The use of such a complex network must be approached at the strategic, tactical, and operational levels [14]. While the first level deals mostly with planning adjustments and extensions to the current network, the last two manage the network operation. The difference between the latter is in the amount of details each one treats, the time horizon considered, and in their operational objectives. At the tactical level, one usually aims at checking if the production plans in the refineries are enough to satisfy the forecasted demand. It must take into account as many operational decisions as possible, while still keeping the problem solvable in a feasible amount of time. It usually spans large horizons comprising from months to years. Most of the previous approaches to this problem focus the operational level [1, 3, 17, 9, 4, 5, 6, 7, 12, 13, 11] where detailed daily pumping schedules must be planned. A few other works could be considered tactical approaches [18, 20, 19]. In any case, they could not be directly applied to our problem, since they handle much simpler networks, usually with only one pipeline.

Solving the tactical problem manually can be ineffective. It is necessary to obey many refinery production schedules and the pipeline network is too complex for in depth manual analysis of all possible operational constraint violations. In fact, the only constraints usually verified in manually constructed solutions are mass conservations at pipelines and tanks. On the other hand, the resulting tactical plan has a huge impact when scheduling daily plans. A bad tactical plan can result in overloading certain pipelines while underusing others. This, in turn, can lead to overdue demands. As another consequence, refinery productions of some derivatives might have to be reduced because local stocks are too high and products are not properly scheduled to be extracted from the corresponding tanks at refineries. To avoid these situations, engineers have to produce very conservative plans with many decisions depending solely on their past experiences. Furthermore, such manual procedures do not admit precise cost optimization considerations.

In this paper, we present a formal description for the planning problem, as we are going to call the tactical planning problem henceforth. This formalization takes into account the operation constraints that will have the greatest influence at the subsequent daily scheduling

problem. A network flow model is proposed to solve the planning problem. The main difficulty was to model specific pipeline characteristics, *e.g.* that they must be always completely full, while moving products through predefined pipeline routes. Transportation costs will be the objective to minimize. A decomposition algorithm is also given to extract individual pumping operations from the network flow solution. These operations could, then, be used as input to detailed daily scheduling algorithms, such as the one proposed in [11]. Computational results showed that the proposed model has an adequate execution time and produces solutions with up to 25% cost reductions when compared to manual solutions.

Section 2 presents the problem description. Section 3 briefly introduces network flow models and section 4 applies them to the tactical planning problem. The computational results can be seen in section 5. Section 6 summarizes our contributions and suggests further advances.

2 Problem Definition

The topology of a pipeline network system is given by three sets: *tanks*, *depots*, and *pipelines*. Tanks are used for product storage. During the whole planning period, a tank can store only a single type of product. Depots are geographically dispersed units where local demands for oil-derivatives occur. Each depot has its own subset of tanks and some depots may also hold refinery facilities. Pipelines interconnect the depots and are used for product transportation. Each individual pipeline connects only two depots.

An illustration of a pipeline system is presented in figure 1, in which products (or *fluids*) F_0, F_1, and F_2 can circulate. In the figure, the four depots D_0 ,D_1, D_2 and D_3 are connected by the pipelines P_0, P_1, P_2, P_3, and P_4. A label T_{ijk} refers to the i-th tank in depot D_j being designated to stock product F_k. Note that two pipelines connect depots D_2 and D_3, which is a common situation in practice.

Volumes must be extracted from tanks before pumped into a pipeline. After pumped out of a pipeline, volumes can either enter a tank or move directly into another pipeline. The sequence of pipelines traversed by a volume when moving from its origin to its destination comprises its route. More precisely, we define a *route* as an alternating sequence of depots and connecting pipelines. For example, the sequence $D_0P_1D_2P_3D_3$ represents a valid route in figure 1. A single pipeline in a route is called a *segment*. All volumes in circulation must have a predefined route assigned to them. Also, a volume cannot be stocked at intermediate depots while moving along its route; it can only be deposited in an available tank at its destination depot.

The problem consists of planning *oil derivative transfers* between depots through valid routes in order to satisfy all depots inventory constraints, meet local demands, and accommodate all the refineries production schedules. The planning must also obey a complex set of operational restrictions over a given planning time period, or *horizon*, usually a month.

2.1 Tactical Viewpoint

The constraints described in the next sections were selected among many real operational constraints [11]. They represents the minimum set of constraints that are necessary to observe in order to produce a feasible tactical planning. All of these constraints must be verified by the end of each day, except for the stock levels, which are verified weekly. When producing a detailed operational pumping schedule, there are additional aspects that must be taken into consideration, and which are not considered by the tactical planner. These restrictions will not be treated here (see [11] for a discussion on this topic). Restrictions regarding tanks, pipelines and stock levels are described in the following three sections, respectively.

2.2 Tank Restrictions

All tanks must satisfy the following restrictions:

(1) Tanks have a fixed and limited maximum capacity which must always be respected. Minimum capacities are all set to zero.

(2) A tank can only store one type of product during the whole planning horizon. This constraint, required by field operators, ensures adequate product quality by avoiding possible mixtures. A depot can contain more than one tank for a given product, but it does not necessarily contain tanks for all products. It is possible that a depot contains no tanks at all, being used solely as an intermediate transmission node between two pipelines. In figure 1, depot D_2 depicts such a situation.

(3) There is a limit to the amount of product that can be extracted and/or injected within a time period. This is due to many operational constraints that arise when managing tanks.

(4) The initial product stock level at each tank is given and must be respected.

Problem specifications in which constraint 3 is satisfied are said to have the *individual tank* property. An alternative view used in some scheduling models [4] relax this restriction by considering virtual tanks for each product at each depot. Such virtual tanks aggregate the capacities of all real tanks of each product in a given depot.

2.3 Pipeline Restrictions

Pipeline restrictions are as follows:

(5) Since they are pressurized, pipelines must be completely filled with products at all times. Hence, in order to pump a certain product out of a pipeline, it is necessary to inject an equal volume at the other pipeline extremity.

(6) At the beginning of the planning horizon, every pipeline is filled with products, separated in batches. All such batches have already being assigned a route that must be preserved.

(7) As there is a limited number of pumps per depot, and given that products have different densities, a maximum flow rate must be observed per pipeline, per product and per flow direction.

(8) In order to simplify the modeling, a set of assignments of products to corresponding acceptable routes is given as input.

2.4 Inventory Constraints

At each depot, stock levels must obey the following restrictions:

(9) The stock of a product at a certain depot at time t is obtained by summing the volumes of that product at time t in all tanks that can stock the product at the given depot. The desired maximum and minimum level of each product stock per depot must be satisfied at the end of specific time periods, usually a week. Some depots might have undefined stock levels for some products. In these cases, the stock of any volume is accepted as valid.

(10) The stock levels at a depot vary mainly due to *production* and *demand* operations. Productions represent volumes created at a local refinery, while demands represent volume consumptions by the local market. Both are given in advance, using market estimates and other data such as raw products availabilities and refinery capabilities. Usually, a refinery depot produces more than the local tanks can actually store within the given planning horizon. Excess must be pumped out so as to accommodate the local productions and satisfy market demands at other locations.

The consumption rate of each product may vary greatly according to monthly seasonal markets. Moreover, it is difficult to foretell exactly the local market needs for a long time period. As a result, pipeline operators are required to constantly update the network schedule to accommodate new demands, guaranteeing they will be satisfied on time. This can be done by solving the problem, taking an updated input instance, every time there is a need for it.

2.5 Input Instances and Solutions

The problem data is composed by: (1) the network, described by the sets of tanks, pipelines, depots and their initial states; (2) operational parameters, such as tank capacities, pipeline flow rates and stock levels; (3) a time horizon and a set of production and demand schedules (see section 2.4) that must be accommodated and satisfied, respectively, at each depot.

Given the problem data, a feasible *solution* is obtained by defining the amount of products that must be pumped into pipelines each day, while satisfying all restrictions. Sometimes, for a given set of production and demands, there may not be a feasible solution. In these cases, it is interesting to identify which productions or demands cannot be satisfied so that the input data can be adjusted.

In this work, the objective function considers the minimization of the total transfer cost in the whole network. In reality, different pipelines might have distinct operational costs.

Here, we are considering homogeneous costs as these data were not fully available. So, minimizing costs is the same as minimizing the total volume transfered over the same time horizon.

3 Network Flow Models

Network Flow Models are a very important subclass of linear programming models. In such models, one has a network of facilities through which commodities can flow in order to meet local demands and to accommodate local production plans. There are directed and constrained connections between some of these facilities, each incurring in different transportation costs. The main objective is to minimize the overall transportation cost while meeting all demands and accommodating all productions around the network. Many well-known problems can be modeled as network flows, such as the shortest path, the maximum flow, the circulation, and the assignment problems.

In our context, we will be focusing on the more general minimum cost flow problem. The following definitions are pertinent:

- $G = (N, A)$ is a directed network, where $N = \{1, \ldots, n\}$ is the set of nodes and A is a set of directed arcs. An arc going from node i to node j, with $i \neq j$, is represented by a pair (i, j), and is also written $i \to j$.
- c_{ij} is the cost to transfer one flow unit across arc (i, j).
- l_{ij} and u_{ij} are lower and upper bounds, respectively, for flows that traverse arc (i, j).
- b_i is the demand at a node i when $b_i < 0$, or is the production at node i when $b_i > 0$.
- x_{ij} is a variable that will hold the actual flow along arc (i, j).

Then, the minimum cost flow problem can be formulated as follows:

$$\begin{aligned} \text{minimize} = & \sum_{(i,j)\in A} c_{ij}x_{ij} \\ \text{subject to:} & \\ & \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i, \quad \forall i \in N \\ & l_{ij} \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A \end{aligned}$$

This formulations is powerful, but it considers only one commodity. We will need a more general formulation where there are different commodities being transported across the network. Clearly their flows must be kept separated. Moreover, the network will be divided in time periods, so that the flow evolution through time is also taken into account.

For the multi-commodity network flow problem, we need the following definitions:

- $G = (N, A)$ is a directed network as defined before.
- $1, 2, \ldots, K$ label the commodities.
- c_{ij}^k is the cost to transfer one flow unit of commodity k along arc (i, j).
- x_{ij}^k is the actual flow of commodity k along arc (i, j).
- $\mathbf{c}^k$ and $\mathbf{x}^k$ are the cost and flow vectors, respectively, for commodity k along all arcs.
- l_{ij}^k and u_{ij}^k are lower and upper bounds, respectively, for the flow of commodity k that traverse arc (i, j).
- l_{ij} and u_{ij} are lower and upper bound, respectively, for the flow of all commodities along arc (i, j).
- $\mathbf{M}$ is a $N \times A$ node-arc incidence matrix, where m_{ia} is 1 if arc a is incident at node i, and and zero otherwise.
- b_i^k is an integer vector that gives the demand or production of commodity k at node i.

Using these definitions, we can formulate the multi-commodities network flow as follows:

$$\begin{aligned}
\text{minimize z} = \quad & \sum_{1 \le k \le K} \mathbf{c}^k \mathbf{x}^k \\
\text{subject to:} \quad & \\
& l_{ij} \le \sum_{1 \le k \le K} x_{ij}^k \le u_{ij}, \quad \forall (i,j) \in A \\
& \mathbf{M}\mathbf{x}^k = \mathbf{b}^k, \quad \forall k \in K \\
& \\
& l_{ij}^k \le x_{ij}^k \le u_{ij}^k, \quad \forall (i,j) \in A \text{ and } \forall k \in K
\end{aligned}$$

Another important property is the possibility to decompose the flow in paths and cycles. We will describe the single commodity case as it is easier to understand, the multi-commodity case being a direct generalization.

Firstly, define $\mathcal{P}$ and $\mathcal{W}$ as the sets of simple paths and cycles, respectively, that are induced by the network G. We can associate flow volumes with these paths and cycles using a function $f : \mathcal{P} \cup \mathcal{W} \mapsto \mathbb{R}$ in such a way that:

$$x_{ij} = \sum_{P \in \mathcal{P}} \delta_{ij}(P) f(P) + \sum_{W \in \mathcal{W}} \delta_{ij}(W) f(W)$$

Algorithm 1 Flow Decomposition Algorithm

Input: $G = (N, A)$ with arc flow x

Output: $\mathcal{P}$ with $f(P), \forall P \in \mathcal{P}$, $\mathcal{W}$ with $f(W), \forall W \in \mathcal{W}$

NOTATION:
y - flow working copy
$A(y) = \{(i,j) \in A | y_{ij} > 0\}$ (Arcs with positive flow in y)
$N(y) = \{i | (i,j) \in A(y) \text{ or } (j,i) \in A(y)\}$ (Nodes incident to arcs in $A(y)$)
$G(y) = (N(y), A(y))$
$\mathcal{S} = \{i \in N(y) | b_i > 0\}$ (supply nodes)
$\mathcal{D} = \{i \in N(y) | b_i < 0\}$ (demand nodes)
s and t are the start and end nodes of path P.
$\Delta(P) = min\{b(s), -b(t), min\{y_{ij} | (i,j) \in P\}\}$ (Capacity of path P)
$\Delta(W) = min\{y_{ij} | (i,j) \in W\}$ (Capacity of cycle W)

1: **procedure** FLOWDECOMPOSITION
2: $y = x$, $\mathcal{P} = \emptyset$, $\mathcal{W} = \emptyset$
3: **while** $A(y) \neq \emptyset$ **do**
4: $s =$ SELECT(y)
5: SEARCH(s,y)
6: **if** Cycle W found **then**
7: $\mathcal{W} = \mathcal{W} \bigcup \{W\}$
8: $f(W) = f(W) + \Delta(W)$
9: $y_{ij} = y_{ij} - \Delta(W), \ \forall (i,j) \in W$
10: **end if**
11: **if** Path P found **then**
12: $\mathcal{P} = \mathcal{P} \bigcup \{P\}$
13: $f(P) = f(P) + \Delta(P)$
14: $y_{ij} = y_{ij} - \Delta(P)$, for all $(i,j) \in P$
15: $b(s) = b(s) - \Delta(P)$
16: $b(t) = b(t) - \Delta(P)$
17: **end if**
18: Update $A(y)$, $N(y)$, $\mathcal{S}$, $\mathcal{D}$
19: **end while**
20: **end procedure**
21: **function** SELECT(y)
22: **if** $\mathcal{S} \neq \emptyset$ **then**
23: **return** $s \in \mathcal{S}$
24: **else**
25: **return** $s \in N(y)$
26: **end if**
27: **end function**
28: **function** SEARCH(s,y)
29: Do a DFS starting with node s until a cycle W in $G(y)$
or a path P in $G(y)$ ending at a node $t \in \mathcal{D}$ is found
30: **return** W or P, accordingly
31: **end function**

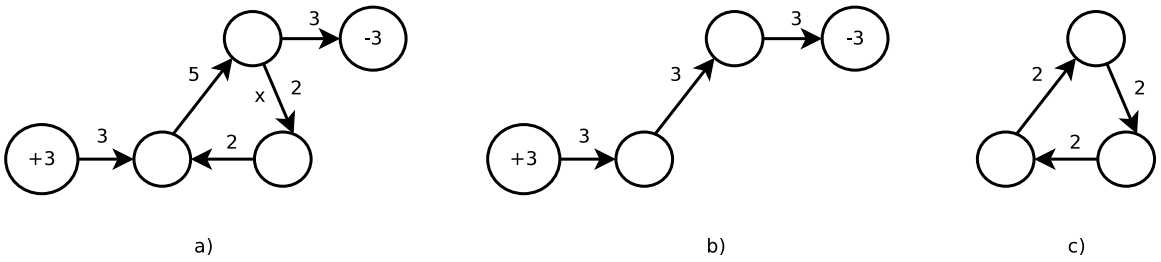

Figure 2: (a) The initial flow. After decomposition, we have a path in (b) with 3 units of volume used to supply the demand, and a cycle in (c) with 2 units of volume that satisfy a lower bound constraint at arc `x`.

where, $\delta_{ij}(P)$ and $\delta_{ij}(W)$ tell if arc (i, j) is ($\delta_{i,j} = 1$) or is not ($\delta_{i,j} = 0$) in the corresponding path or cycle.

This equation gives rise to the decomposition algorithm described as algorithm 1, at page 8. At every iteration, one path or cycle starting at the node provided by function SELECT is found by function SEARCH. The flow that pass through the path or cycle is the maximum that can be extracted without leaving any of its arcs with a negative flow. After assigning this maximum flow value to the path or cycle, it can be canceled out from the remaining flow. The procedure ends when every arc has no flow left. Figure 2 shows a simple example of this procedure.

This decomposition algorithm will take the solution produced by running the network flow model as the basic information from where to extract input instances for the daily scheduling algorithm.

4 A Network Flow Model for the Planning Phase

In a previous work [4], a classical network flow model for our problem has been studied. This integer programming model, however, proved inefficient in practice. Another work [2] used a linear relaxation from this model in order to get a solution for the tactical planning. This relaxed solution proposed flows along each pipeline, at every time instant. With this information, a heuristic was designed to construct operational schedules with feasible pumping movements. But the detailed operational schedules so produced spanned only a few days or a few products. The model we propose overcomes these difficulties and its output can be directly used by the daily scheduler proposed in [11].

But, first, note that the classical network flow model for the problem is not yet suitable because one cannot constrain the flow to follow predefined routes. This could be troublesome, as we could obtain inadequate paths for some movements. An alternative approach is to build the network flow using the predefined routes for each product. This approach is described in the next section.

4.1 Network Model Definition

The network model will be built by considering pairs (r, p), where r is a route and p is a product. Each of these pairs represents a commodity. There will be T time periods and the last one will be a special one. It will signal the end of the horizon. Each period spans one day, as demands and productions are given in a daily basis. This will also help to keep the model small, with less variables and with less constraints applying to each time period. Along with the given sets of depots, pipelines, and products the input also includes the constants described in table 1.

Table 1: Input constants in an instance

Constant	**What it represents**
$vol(l)$	total volume of pipeline l.
$PilelineStart_l$	A set of triples (p, qty, r) each representing a quantity qty of product p that must follows route r that is initially inside pipeline l. The sum of the qty values over all triples is equal to $vol(l)$.
$stock_{d,tk}$	the initial volume inside tank tk at depot d.
$production_{p,d,t}$	the volume of product p produced at depot d in period t.
$demand_{p,d,t}$	the volume of product p demanded at depot d in period t.
$stockLevelMin_{d,p}$, $stockLevelMax_{d,p}$	the minimal and maximum volumes of product p allowed at each depot.
$maxQtyInject_{e,p,l}$, $maxQtyInject_{p,l}$, $maxQtyInject_{e,l}$, $maxQtyInject_l$	the maximum quantity that can be injected in one time period in pipeline l. Here, e represents a flow direction for product p in pipeline l: P for the normal direction and R for the reverse direction. When any of the three indices is missing, the intended meaning is to take the sum over the missing indices.
$maxTransfer_{r,p,n,u}$	the maximum volume of product p that can be fully injected inside the n-th pipeline of route r after u time periods. If $n-1$ the last pipeline in route r, this constant is also defined for n meaning the amount that can be delivered at the last depot in route r.
$maxDepotQtyInject_d$	the maximum quantity that can be injected into pipelines leaving depot d in one time period. It is usually calculated by considering the number of pumps available at depot d.

Nodes will be created in accordance with the following rules (see also Table 1 for a

summary of term definitions):

- **Tank Nodes**: For each tank tk, each depot d and each time period t, with $1 \leq t \leq T$, there will be two nodes: (1) node i (or $NodeStartTk_{d,tk,t}$) with $b_i = stock_{d,tk}$ if $t = 0$, or else $b_i = 0$ if $t > 0$, and (2) node j (or $NodeEndTk_{d,tk,t}$) with $b_j = 0$. The first node represents the tank state at the beginning of period t and the other represents it at the end of that period. The tank stock, as well as its production and demand flows will go through these nodes.

- **In and out nodes**: For each product p with tanks at depot d and each time period t, with $0 \leq t < T$, there will be a node i (or $NodePIn_{p,d,t}$) and a node j (or $NodePOut_{p,d,t}$) with $b_i = b_j = 0$. Every incoming and outgoing flow at a depot will pass through these nodes. These are intermediate nodes between tanks and the pipeline network.

- **Demand and Production Nodes**: For each product p, each depot d and each period t, with $0 \leq t < T$, there will be: (1) a node i (or $NodeDem_{p,d,t}$) with $b_i = -demand_{p,d,t}$, and (2) a node j (or $NodePro_{p,d,t}$) with $b_j = production_{p,d,t}$.

- **Pipeline Initial Stock Nodes**: For each pipeline l and each triple (p, qty, r) in $PilelineStart_l$, there will be a node i (or $NodePipeStart_{p,r}$) with $b_i = qty$.

- **Pipeline Terminal Stock Nodes**: All pipelines must remain completely filled at the end of the planning horizon. So, for each pipeline l there will be a node i (or $NodePipeEnd_l$) with $b_i = vol(l)$. In this way, the flow at the end of the horizon will sum up to the pipeline volume.

- **Pipeline Route Nodes**: Some nodes will be used to model movements of products along the pipelines. An index ti names the period where the movement starts, and an index t will represent the current period. For each pair (r, p) of a route r and a product p, and each pipeline l that is part of route r, there will be a node i (or $NodePipe_{r,l,p,ti,t}$) with $b_i = 0$, for each time period $0 \leq ti \leq t < T$. In order to handle situations when the volume was already inside the pipeline, at the first period we have nodes with $ti = 0$.

Table 2 summarizes each node function. The next step is to connect these nodes with arcs. In the following description, the traffic costs are zero and flows must be greater than zero unless stated otherwise.

- **Tank Stock Arcs**: For each node $NodeEndTk_{d,tk,t}$ there will be two arcs. The first arc, $NodeStartTk_{d,tk,t} \rightarrow NodeEndTk_{d,tk,t}$, aggregates production and previous stocks. The second one, $NodeEndTk_{d,tk,t} \rightarrow NodeStartTk_{d,tk,t+1}$, represents the transition of a stock between time periods. Figure 3 show these arcs as the down arrows that come out of tank nodes.

Table 2: Node description for the Pipeline Network flow Model

Node	What it agregates
$NodeStartTk_{d,tk,t}$	production and previous stocks at the start of period t in depot d and tank tk.
$NodeEndTk_{d,tk,t}$	total of volumes received from the network and volumes that were not distributed at period t in depot d and at tank tk.
$NodePIn_{p,d,t}$	volumes received from the network at the start of period t of a particular product p in depot d.
$NodePOut_{p,d,t}$	volumes that will be distributed to the network at period t of a particular product p in depot d.
$NodePro_{p,d,t}$	the production volume $production_{p,d,t}$ of a product p in depot d at period t.
$NodeDem_{p,d,t}$	the demmanded volume $demmand_{p,d,t}$ of a product p in depot d at period t.
$NodePipeStart_{p,r}$	the volume of product p inside a pipeline l at the beginning of the horizon and that has to follow route r.
$NodePipeEnd_{l}$	volumes that will be stocked inside pipeline l at the end of the horizon.
$NodePipe_{r,l,p,ti,t}$	volumes of product p that traveled along route r and are currently at pipeline l at period t having departed from route's r origin at period ti.

- **Tank In and Out Arcs**: For each pair of nodes $NodeStartTk_{d,tk,t}$ and $NodeEndTk_{d,tk,t}$ there will be two arcs, namely, $NodePIn_{p,d,t} \rightarrow NodeEndTk_{d,tk,t}$ and $NodeStartTk_{d,tk,t} \rightarrow NodePOut_{p,d,t}$. Here, tank tk stores product p. Volumes pass through these arcs before coming from or going to some pipeline of the network. Figure 3 shows them as the crossing arcs.
- **Production and Demand Arcs**: For each pair of nodes $NodeDem_{p,d,t}$ and $NodePro_{p,d,t}$, there will be arcs $NodeEndTk_{o,tk,t} \rightarrow NodeDem_{p,d,t}$ and $NodePro_{p,d,t} \rightarrow NodeStartTk_{o,tk,t}$, for all tanks tk that can store product p at depot d. The maximum capacity of these arcs will be the same as the demand or production volumes of the respective products. Figure 3 shows these arcs as the two left horizontal arrows at the left.
- **Initial Stock Arcs**: For each node $NodePipeStart_r$ and its associated content (p, qty, r), there will be an arc $NodePipeStart_r \rightarrow NodePipe_{r,l,p,0,0}$ with minimum and maximum capacities given by qty. Figure 4 illustrates the subnetwork generated

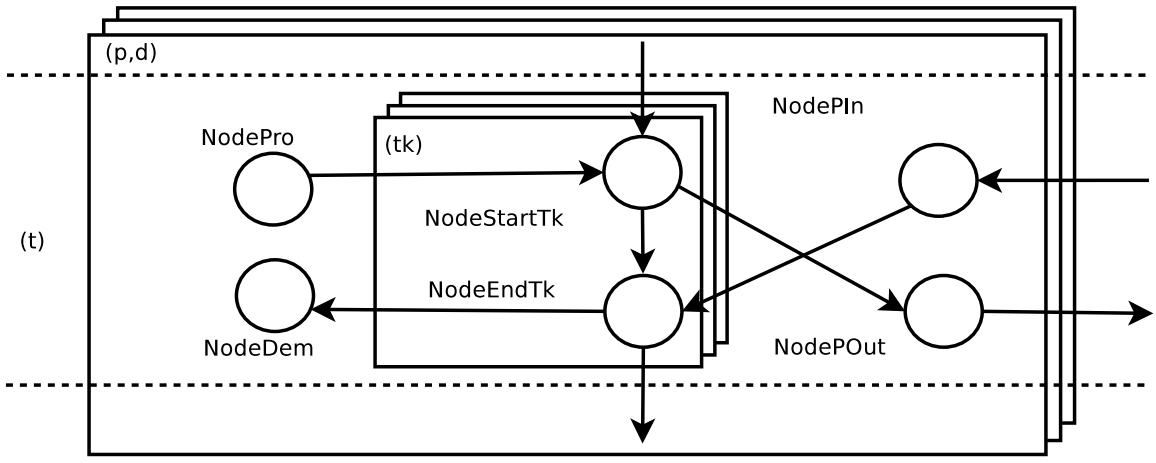

Figure 3: Nodes and arcs representing pairs of depots d and products p. Indexes in parenthesis are used to index the nodes enclosed inside the rectangle. The occluded rectangles represent the multiplicity of that network with regard to the inner index.

to represent one pipeline stock.

- **Outgoing Terminal Arcs**: For each node $NodePipe_{r,l,p,t,t}$ where l is the first pipeline and d is the starting terminal of route r, there will be an arc $NodePOut_{p,d,t} \rightarrow NodePipe_{r,l,p,t,t}$. The maximum capacity of this arc is $maxTransfer_{r,p,0,1}$ which measures the maximum quantity of product p that can be injected in the pipeline l over a whole day. These arcs are the leftmost ones shown in figure 5.

- **Intermediary Stock Arcs**: For each node $NodePipe_{r,l,p,ti,t}$ there will be an arc named $NodePipe_{r,l,p,ti,t} \rightarrow NodePipe_{r,l,p,ti,t+1}$, and with capacity given by the pipeline volume $vol(l)$. These arcs are the vertical ones in figures 4 and 5.

- **Route Arcs**: For each pair of nodes $NodePipe_{r,l_1,p,ti,t}$ and $NodePipe_{r,l_2,p,ti,t}$ where l_1 and l_2 are sequential pipelines in route r, there will be an arc $NodePipe_{r,l_1,p,ti,t} \rightarrow NodePipe_{r,l_2,p,ti,t}$ between them. If x is the index of pipeline l_2 in route r, the maximum capacity of this arc is calculated as $maxTransfer_{r,p,x,t-ti+1}$. This is the same as the maximum quantity of product p that can be transfered out of pipeline l_1 starting at period ti and ending at period t from the beginning of route r. Figures 4 and 5 show these arcs connecting adjacent nodes that are inside pipeline rectangles.

- **Incoming Terminal Arcs**: For each node $NodePipe_{r,l,p,t,t}$ where l is the last pipeline and d is the last terminal of route r, there will be an arc $NodePipe_{r,l,p,ti,t} \rightarrow NodePIn_{p,d,t}$. If $n-1$ is the index of pipeline l in route r, the maximum capacity of this arc is calculated as $maxTransfer_{r,p,n,t-ti+1}$. This is the same as the maximum quantity that can be transfered to the ending depot d starting at period ti and ending at period t along route r. These arcs are the rightmost ones in figure 5.

- **Pipeline Terminal Stock Arcs**: For each node $NodePipeEnd_l$, with T the last

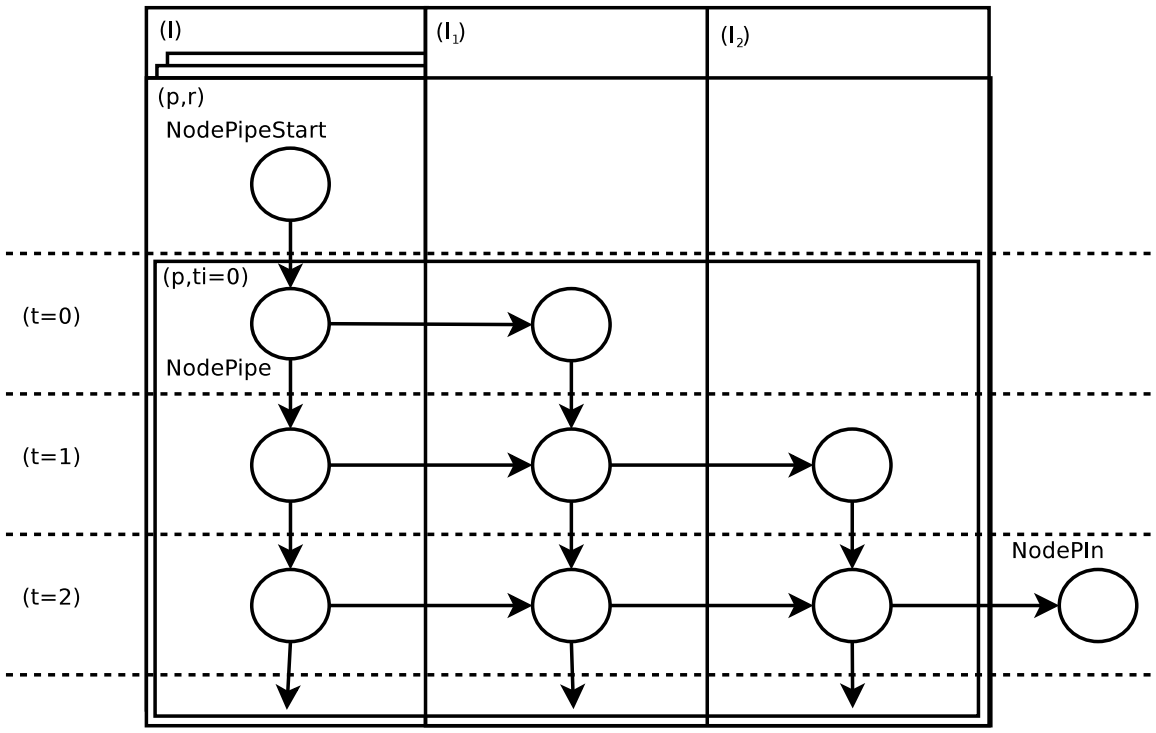

Figure 4: Nodes and arcs for the stock of a pipeline with contents (p, qty, r). Items within parenthesis index the nodes enclosed inside the rectangle. The occluded rectangles represent the multiplicity of that network with regard to the inner index.

period, there will be an arc $NodePipe_{r,l,p,ti,T-1} \rightarrow NodePipeEnd_l$ with maximum capacity equal to $vol(l)$.

4.2 A Linear Programming Model

In order to define the constraints and the objective function of the model we need the set of nodes at period t, $Nodes_t$, the set of nodes along pipeline l at period t, $Nodes_{l,t}$, and the set of all nodes for product p at pipeline l and at period t, $Nodes_{p,l,t}$. Also, PL will indicate the set of all pipelines.

The following model is not strictly a network flow model. After removing some constraints and adding others, we will be working with a linear programming model. But the problem structure itself retains many features of a Multi-commodity Multi-period Network Flow Model.

Nodes Mass Balance

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i + Slack, \quad \forall i \in Nodes_t, \quad 0 \leq t < T \tag{1}$$

We consider that the pipeline initial and terminal stock nodes are all in the $Nodes_0$ set.

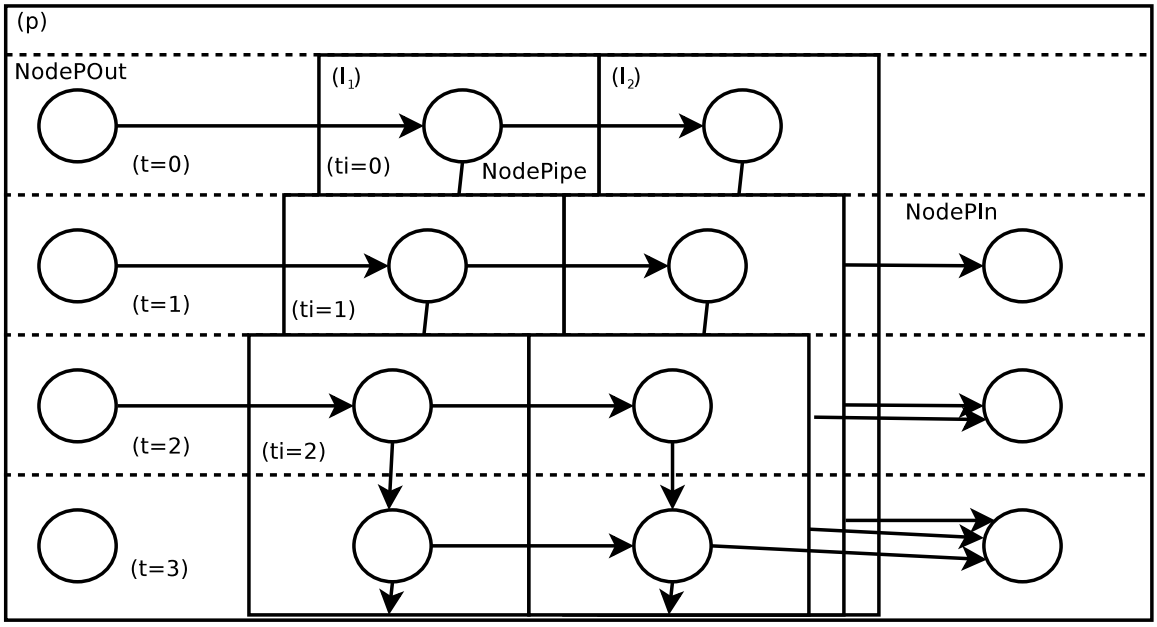

Figure 5: Nodes and arcs representing volumes traveling along routes. For a fixed route r and product p, at every period ti a new subnetwork is created so that the quantity will only be delivered after enough periods have passed. The subnetwork for period $t = 3$ was omitted to show more of the subnetwork for period $t = 2$. Items within parenthesis index the nodes enclosed inside the rectangle. The occluded rectangles represent the multiplicity of that network with regard to the inner index.

Also, this constraint is not applied to tank nodes in the last period. This is necessary in order to model what will be in stock inside tanks after the end of the horizon. The *Slack* variable will only be present when one wants to know which stocks cannot be satisfied. In this case, if node i is a production node we define $Slack = -InsatisfiedProduction_i$, and for a demand node i we let $Slack = InsatisfiedDemand_i$.

Arc Bounds

$$l_{ij} \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A \tag{2}$$

Pipelines must always remain full

$$\sum_{j:(i,j)\in A} x_{ij} = vol(l), \quad \forall i \in Nodes_{l,t}, \quad l \in PL, \quad 0 \leq t < T \tag{3}$$

An amount injected at one end equals the amount delivered at the other end

$$\sum_{\substack{j:(i,j)\in A \\ Deliver(R,i,j)}} x_{ij} = \sum_{\substack{j:(j,i)\in A \\ Inject(P,j,i)}} x_{ji}, \quad \forall i \in Nodes_{l,t}, \quad l \in PL, \quad 0 \leq t < T \tag{4}$$

$$\sum_{\substack{j:(i,j)\in A \\ Deliver(P,i,j)}} x_{ij} = \sum_{\substack{j:(j,i)\in A \\ Inject(R,j,i)}} x_{ji}, \quad \forall i \in Nodes_{l,t}, \quad l \in PL, \quad 0 \leq t < T \tag{5}$$

The functions $Inject(e, j, i)$ and $Deliver(e, i, j)$ tell if arcs (j, i) and (i, j) inject or deliver volumes, respectively, through extremity e in the pipeline represented by node i.

Pipelines Flow Rates

$$\sum_{\substack{j:(i,j)\in A \\ Inject(R,i,j)}} x_{ij} \leq maxQtyInject_{R,p,l}, \quad \forall i \in Nodes_{p,l,t}, \quad p \in P, \quad l \in PL, \quad 0 \leq t < T \tag{6}$$

$$\sum_{\substack{j:(i,j)\in A \\ Inject(P,i,j)}} x_{ij} \leq maxQtyInject_{P,p,l}, \quad \forall i \in Nodes_{p,l,t}, \quad p \in P, \quad l \in PL, \quad 0 \leq t < T \tag{7}$$

$$\sum_{\substack{j:(i,j)\in A \\ Inject(R,i,j)}} x_{ij} + \sum_{\substack{j:(i,j)\in A \\ Inject(P,i,j)}} x_{ij} \leq maxQtyInject_{p,l}, \forall i \in Nodes_{p,l,t}, \, p \in P, \, l \in PL, \, 0 \leq t < T \tag{8}$$

$$\sum_{\substack{j:(i,j)\in A \\ Inject(R,i,j)}} x_{ij} \leq maxQtyInject_{R,l}, \quad \forall i \in Nodes_{l,t}, \quad l \in PL, \quad 0 \leq t < T \tag{9}$$

$$\sum_{\substack{j:(i,j)\in A \\ Inject(P,i,j)}} x_{ij} \leq maxQtyInject_{P,l}, \quad \forall i \in Nodes_{l,t}, \quad l \in PL, \quad 0 \leq t < T \tag{10}$$

$$\sum_{\substack{j:(i,j)\in A \\ Inject(R,i,j)}} x_{ij} + \sum_{\substack{j:(i,j)\in A \\ Inject(P,i,j)}} x_{ij} \leq maxQtyInject_{l}, \forall i \in Nodes_{l,t}, \, l \in PL, \, 0 \leq t < T \tag{11}$$

Depot Injection Limit

$$\sum_{(i,j)\in ArcsOutD_{d,t}} x_{ij} \leq maxDepotQtyInject_{d}, \quad \forall d \in DP, \quad 0 \leq t < T \tag{12}$$

The set $ArcsOutD_{d,t}$ contains all the arcs that start in $NodePOut_{p,d,t}$, for all products in depot d and at period t.

Stock Levels

$$stockLevelMin_{d,p} \leq \sum_{(i,j)\in ArcsOutTankEnd_{d,p,t}} x_{ij} \leq stockLevelMax_{d,p} : \forall d \in DP, \forall p \in P, \quad \forall t \in WE \tag{13}$$

The set $ArcsOutTankEnd_{d,p,t}$ contains all arcs connecting node $NodeEndTk_{d,tk,t}$ to node $NodeStartTk_{d,tk,t+1}$, for every tank tk at depot d with product p and at period t. These are the arcs that transmit stocks between periods. The set WE contains the periods where stock levels should be verified. It should be of the form $\{6, 13, 20, 27...\}$, representing a weekly check.

Objective Function When engineers want to check if a proposed production plan is feasible against the forecasted demands they use the following objective function:

$$\text{minimize}\Big(\sum_{i\in NodesPro} InsatisfiedProduction_i + \sum_{i\in NodesDem} InsatisfiedDemmand_i\Big), \tag{14}$$

where the sets $NodesPro$ and $NodesDem$ contain the all production and demand nodes, respectively. The solution obtained with this objective function can also be used to correct production or demand values. This correction deals with many economical and operational issues that are not discussed here.

After finding a feasible set, the next objective function to be used is the classic one:

$$\text{minimize} \sum_{(i,j)\in A} x_{ij} \tag{15}$$

Together with the constraints, we have a complete model for the planning phase. The next step is to extract the orders using a modified decomposition algorithm.

4.3 Network Flow Decomposition

Figures 3, 4 and 5 show no bidirectional arcs and no paths going back to a previously visited node. So, no cycle can be generated within the network. This observation allows us to use a specialized version of the flow decomposition algorithm. We also introduced some modifications that are specific to our problem, as described in Algorithm 2, at page 18.

The main purpose of the preferences mentioned in SELECT and SEARCH is to make it easier to supply a demand by considering the longest time between the moment it is needed and the moment it is available. These heuristics do not affect the result stating that the flow can be fully decomposed into paths.

Let $\mathcal{P}'$ the set of paths and let $f(P)$ be the flow associated to a path $P \in \mathcal{P}'$. We create the necessary orders to be input to an operational scheduling solver [11] by analyzing the nodes traversed by P. An order O will be represented by the tuple

$$O = (OriginTank, DestinationTank, Product, Volume, Route, Deadline).$$

Algorithm 2 Pipeline Network Flow Decomposition Algorithm

Input: $G = (N, A)$ with arc flow x

Output: $\mathcal{P}$ with $f(P), \forall P \in \mathcal{P}$

NOTATION:
y - flow working copy
$A(y) = \{(i,j) \in A | y_{ij} > 0\}$ (Arcs with positive flow in y)
$N(y) = \{i | (i,j) \in A(y)$ or $(j,i) \in A(y)\}$ (Nodes incident to arcs in $A(y)$
$G(y) = (N(y), A(y))$
$\mathcal{S} = \{i \in N(y) | b_i > 0\}$ (supply nodes)
$\mathcal{D} = \{i \in N(y) | b_i < 0\}$ (demand nodes)
s and t are the start and end nodes of path P.
$\Delta(P) = min\{b(s), -b(t), min\{y_{ij} | (i,j) \in P\}\}$ (Capacity of path P)

procedure PIPELINEFLOWDECOMPOSITION
  $y = x$, $\mathcal{P} = \emptyset$, $\mathcal{W} = \emptyset$
  while $A(y) \neq \emptyset$ **do**
    $s =$ SELECT(y)
    SEARCH(s,y)
    if Path P found **then**
      $\mathcal{P} = \mathcal{P} \bigcup \{P\}$
      $f(P) = f(P) + \Delta(P)$
      $y_{ij} = y_{ij} - \Delta(P) \forall (i,j) \in P$
      $b(s) = b(s) - \Delta(P)$
      $b(t) = b(t) - \Delta(P)$
    end if
    Update $A(y)$,$N(y)$,$\mathcal{S}$,$\mathcal{D}$
  end while
end procedure
function SELECT(y)
  Order nodes in $\mathcal{S}$ first by the most pressing demands;
  secondly, by those tank nodes at the last period, and
  lastly, by pipeline terminal stock nodes.
  return the first node in $\mathcal{S}$ by the given ordering.
end function
function SEARCH(s,y)
  Do a DFS starting with node s until a path P is found in $G(y)$ ending at node $t \in \mathcal{D}$.
  Inside the DFS, before choosing the next node,
  order the candidate nodes firstly by being of the same tank, if applicable,
  then by those nodes being in the same time period.
  return The path P found.
end function

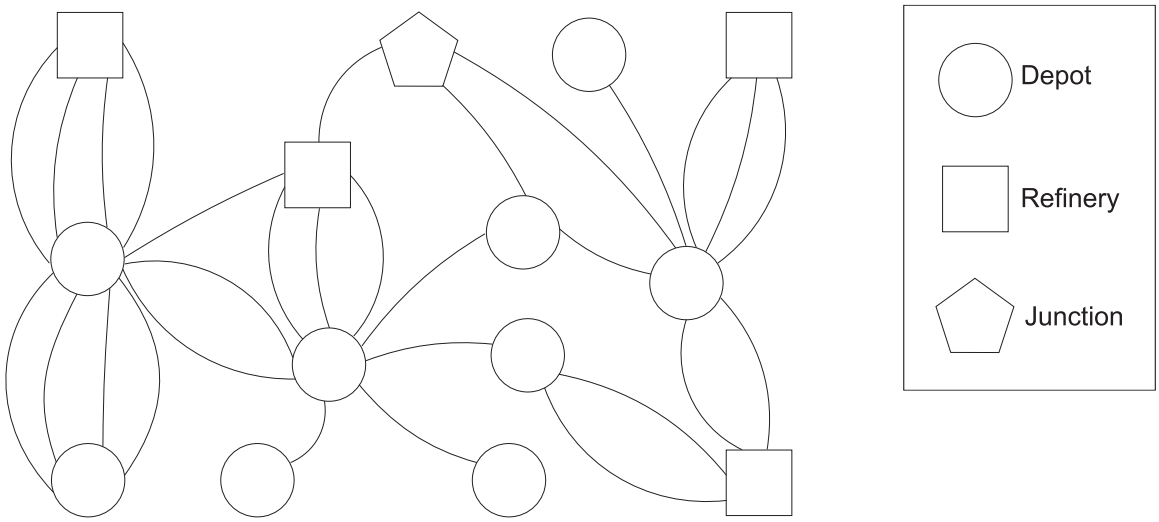

Figure 6: The inland pipeline network.

A path P is built following the inverse order with respect to the network timeline. This way it will always start at a demand node, a tank node or a pipeline stock node at the last period. The deadline is either the demand node period or the last period when the node corresponds to tank or pipeline stocks. A path P should always end at a production node, tank initial node or pipeline initial stock node. Sometimes the path will not contain any pipeline nodes. Such is the case when a demand is satisfied by a production inside the depot. In these cases, the route will be null. Conversely, the path P can also contain more than one kind of pipeline node. In this case, one must create as many orders as there are pipeline nodes in P, because each order has exactly one route associated to it. The initial and terminal tanks should be the ones connecting to pipeline nodes. If there is no route, the two tank variables will have be the same value. Finally, the volume is given by the flow $f(P)$ associated with the path.

A decomposition of the network can generate a huge number of orders with small volumes. It is interesting, then, to aggregate similar orders. If two orders have the same origin tank, destination tank, product, route, and close deadlines, they can be merged into one order with their volumes summed up. Two deadlines are close if they happen within a range of days. The horizon can be divided in as many ranges as necessary. In our case, it was divided in ten ranges of three days each, generating a set of orders with adequate size.

5 Computational Results

Solutions were obtained on a *Intel Pentium Core 2 Duo* 2.1 GHz CPU platform, with 3MB L2 Cache and 4GB of RAM. The program was coded in C++ and compiled using *GCC-4.2* without optimization. The network flow model used the CPLEX 10 solver with presolving columns and row elimination. Between the linear programming solving techniques available, the barrier method was selected as it gave the best execution times.

All instances refer to the same network topology, with 19 pipelines, as exposed in figure 6. The number of tanks vary from 0 to 20 at each depot, with a total of 192 tanks. Also, there are 11 products that circulate through the network.

The model was tested against 12 real instances, spanning a whole year of planning, one instance per month. Some characteristics of these instances cannot be disclosed, as they could expose classified data.

From table 3 it can be seen that the whole execution takes around 20 seconds. This makes the procedure suitable for testing "what-if" scenarios. The number of orders per month never exceed 900, and so, roughly, it is necessary to schedule 30 orders per day in order to reach the planning goal.

Instance	Rows	Columns	Non-zeros	Model Time (s)	Decomp. Time (s)	Orders
1	23801	51948	266778	22.1	3.3	700
2	22935	49788	256470	19.5	3.2	685
3	22346	48607	248310	18.4	3.5	700
4	22976	50494	257631	17.9	4.1	731
5	23237	50711	260774	20.3	4.5	759
6	24532	54414	281496	23.0	4.0	841
7	23079	50448	258469	20.3	3.4	826
8	24355	53797	278574	28.0	4.2	897
9	24462	53152	273019	22.9	3.7	776
10	25948	57326	295073	26.8	3.9	796
11	25536	56357	290854	26.5	4.0	818
12	21554	46487	236411	17.6	3.9	727
Mean	**23730**	**51960**	**266988**	**21.9**	**3.8**	**771**

Table 3: Execution Results

Petrobras also provided data on manually generated plans for each of the instances. In table 4, the total volume scheduled to be transfered by a real planning is compared against the one proposed by the model. The mean proportion is 0.75, meaning that the model transfers 25% less units of volume each month on the average. This represents a quite significant economy when multiplied by the very large volumes that are moved each month. It must be noted that the model does transfer what is necessary in order to satisfy every demand and keep the stock levels in safe conditions. Thus, these savings in product shipments were not obtained at the expense relaxing some problem constraints.

Instance	1	2	3	4	5	6	7	8	9	10	11	12	**Mean**
Proportion	0.72	0.69	0.80	0.71	0.81	0.82	0.83	0.76	0.72	0.67	0.69	0.73	0.75 ± 0.05

Table 4: Proposed planning vs Real planning - Total Network Flow Proportion

One reason for the 25% reduction could be related to the choice of products used to push others along pipelines, when really necessary. Sometimes, the products used to push others out of a pipeline are not directly useful for the planning goals. This incurs in wasted pumping moves. Examining table 5 one might see that the differences in product choices are high. As an example, product P_{10} has a proportion of only 0.15. Another important aspect of this issue is the choice of routes. A refinery has many outgoing routes and, so,

poor choices of routes in which products will be set to flow along may also lead to wastes. Table 6 shows that pipelines D_1, D_9 and D_{18} are rarely used. Hence, there will be less need to push products out of them.

Product	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	**Mean**
Proportion	0.91	0.65	1.05	0.73	1.05	0.79	0.92	0.83	0.58	0.15	0.56	0.75 ± 0.26

Table 5: Proposed planning vs Real planning - Total Network Flow Proportion by Product

The objective was to minimize the total volume transported across the network. This means that a product will be moved only to satisfy a constraint. Engineers have the same objective but, by depending solely on their past experiences, and having very limited time to workout a solution, they will accept any solution that keeps wastes below an " acceptable " level. The threshold where the solution is acceptable might be related to the 25% reduction in transported volumes.

Pipeline	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	
Proportion	0.27	0.85	0.65	0.95	0.91	0.85	1.05	1.00	0.34	0.82	
Pipeline	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	D_{16}	D_{17}	D_{18}	D_{19}		**Mean**
Proportion	0.58	0.64	0.70	0.66	1.04	1.14	0.78	0.18	0.91		0.75 ± 0.27

Table 6: Proposed planning vs Real planning - Total Network Flow Proportion by Pipeline

6 Conclusions and Future Work

This paper described the oil pipeline planning problem faced by Petrobras, a Brazilian oil company. The operation of its pipeline network involves many constraints related to tanks, volumes, pipelines utilization, stock levels as well as demand and production schedules. There is also a large set of complex operational constraints that must be satisfied by feasible schedules [11]. The objective was to obtain a minimum cost transportation plan for all oil derivatives, given that the specified demands and productions are adequate to meet the constraints. Although the real network operation requires a daily pumping schedule, here the problem is solved at the tactical level, where the horizon spans one or more months. We focused the tactical plan only. See [11] for a scheduler that takes as input the tactical plan and outputs a detailed daily schedule.

In its fullest, the problem can only be solved for a span of a few days. In order to make it more tractable, only the most important characteristics were included in our formal definition of the problem. Engineers validated these characteristics as the ones having the greatest impact over the daily operation. An important characteristic required that all pipelines must always remain completely full, and tanks should keep the local stock within their capacities. Transfers had to observe permitted pipeline flow rates, which can vary by pipeline, by product and by flow direction. Productions and demands were satisfied on a daily basis. Stock levels were verified weekly. Another important constraint was that volumes should follow along predefined routes that depend on product type.

A model to solve this problem was proposed using network flow ideas. Nodes were defined so as to model the state of each depot. Pipelines were modeled by classifying the nodes that represent each route that a product can travel. The model could also be used to test if the proposed demand and production schedules were feasible or, else, to give a minimum cost transportation plan for feasible movements.

Test results showed that the model can adequately deal with the Petrobras network topology and with monthly scenarios, within small computational times. Also, the solutions generated were compared to the ones proposed by the company expert engineers. It was found that the model almost always gave a 25% cost reduction. Tentative interpretations for this gain were listed.

Although the model already included most of the constraints the engineers consider when managing the tactical planning problem, more constraints could be considered. One of them could be to model what happens when two products that cannot make contact are to be injected in a pipeline. In this case, a third product must be used to separate them. Also, since sometimes pipeline and tank maintenance periods are known beforehand, such constraints could also be incorporated into the model.

The algorithms presented here were developed together with the work described in [11]. Finally, the tactical plan it outputs could be used as input to the daily scheduler modeled in [11].

References

[1] Vanessa Alves and Virgílio J.M.F. Filho. Pipeline scheduling of petroleum derivatives using genetic algorithm. In *IV Congresso Brasileiro de Pesquisa e Desenvolvimento em Petróleo e Gás*, Campinas, Brazil, 2007.

[2] Viviane Monteiro Braconi. Heurísticas multifluxo para roteamento de produtos em redes dutoviárias. Master's thesis, Departamento de Informática, PUC-Rio, Rio de Janeiro, RJ, Brasil, 2002.

[3] Diego C. Cafaro and Jaime Cerdá. Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation. *Computers & Chemical Engineering*, 28(10):2053–2058, 2004.

[4] E. Camponogara and P. S. Souza. A-Teams for oil transportation problem through pipelines. In *Information Systems Analysis and Synthesis*, Orlando, United States, 1996.

[5] Eduardo Camponogara. A-Teams para um problema de transporte de derivados de petróleo. Master's thesis, Instituto de Matemática, Estatística e Ciência da Computação, Universidade Estadual de Campinas, Campinas, Brazil. In Portuguese., 1995.

[6] J.M. de la Cruz, B. Andrés-Toro, A. Herrán-González, E. Besada Porta, and P. Fernandez Blanco. Multiobjective optimization of the transport in oil pipelines networks. In *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation*, volume 1, pages 566–573, 2003.

[7] J.M. de la Cruz, A. Herrán-González, J.L. Risco-Martín, and B. Andrés-Toro. Hybrid heuristic and mathematical programming in oil pipelines networks: Use of immigrants. *Journal of Zhejiang University SCIENCE*, 6A(1):9–19, 2005.

[8] M. A. H. Dempster, N. Hicks Pedron, E. A. Medova, J. E. Scott, and A. Sembos. Planning logistics operations in the oil industry. *The Journal of the Operational Research Society*, 51(11):1271 – 1288, 2000.

[9] Erito M. Souza Filho, Virgílio J.M.F. Filho, and Leonardo S. de Lima. Variable neighborhood search (VNS) applied to pipeline distribution problem with capacity constraints. In *IV Congresso Brasileiro de Pesquisa e Desenvolvimento em Petróleo e Gás*, Campinas, Brazil, 2007.

[10] A. Herran, J.M. de la Cruz, and B. de Andres. A mathematical model for planning transportation of multiple petroleum products in a multi-pipeline system. *Computers & Chemical Engineering*, 34(3):401 – 413, 2010.

[11] Tony M. Lopes, Andre A. Ciré, Cid C. Souza, and Arnaldo V. Moura. Special issue on the 14th international conference on principles and practice of constraint programming. *Constraints*, 15(2), 2010.

[12] L. Magatao, L.V.R. Arruda, and F. Neves. A mixed integer programming approach for scheduling commodities in a pipeline. *Computers & Chemical Engineering*, 28(1):171–185, 2004.

[13] L. Magatao, L.V.R. Arruda, and F. Neves. Using CLP and MILP for scheduling commodities in a pipeline. *Computer-Aided Chemical Engineering*, 20B:1027–1032, 2005.

[14] Christos T. Maravelias and Charles Sung. Integration of production planning and scheduling: Overview, challenges and opportunities. *Computers & Chemical Engineering*, 33(12):1919 – 1930, 2009. FOCAPO 2008 - Selected Papers from the Fifth International Conference on Foundations of Computer-Aided Process Operations.

[15] R.L. Milidíu, Frederico dos Santos Liporace, and Carlos José P. de Lucena. Pipesworld: Planning pipeline transportation of petroleum derivatives. In *Proceedings of ICAPS'03 - Workshop on the Competition: Impact, Organization, Evaluation, Benchmarks*, Trento, Italy, 2003.

[16] Arnaldo V. Moura, Romulo A. Pereira, and Cid C. de Souza. Scheduling activities at oil wells with resource displacement. *International Transactions in Operational Research*, 15(6):659–683, 2008.

[17] R. Rejowski and José M. Pinto. A novel continuous time representation for the scheduling of pipeline systems with pumping yield rate constraints. *Computers & Chemical Engineering*, 32:1042–1066, 2008.

[18] Susana Relvas, Ana Paula F. D. Barbosa-Póvoa, Henrique A. Matos, João Fialho, and António S. Pinheiro. Pipeline scheduling and distribution centre management - a real-world scenario at CLC. In *Proceedings of the 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, pages 2135–2140, Garmisch-Partenkirchen, Germany, 2006.

[19] Susana Relvas, Henrique A. Matos, Ana Paula F. D. Barbosa-Póvoa, and João Fialho. Reactive scheduling framework for a multiproduct pipeline with inventory management. *Industrial & Engineering Chemistry Research*, 46(17):5659–5672, 2007.

[20] Susana Relvas, Henrique A. Matos, Ana Paula F. D. Barbosa-Póvoa, João Fialho, and António S. Pinheiro. Pipeline scheduling and inventory management of a multiproduct distribution oil system. *Industrial & Engineering Chemistry Research*, 45(23):7841–7855, 2006.