EXACT AND HEURISTIC APPROACHES TO THE MULTIPRODUCT PIPELINE SCHEDULING PROBLEM

Erito Marques Souza Filho
Programa de Engenharia de Produção – COPPE/UFRJ
Caixa Postal 68507, CEP 21945-970, Rio de Janeiro, RJ, Brasil
eritomarques@yahoo.com.br

Laura Bahiense
Programa de Engenharia de Produção – COPPE/UFRJ
Caixa Postal 68507, CEP 21945-970, Rio de Janeiro, RJ, Brasil
laura@pep.ufrj.br

Leonardo Silva de Lima
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca – CEFET/ Produção
Av. Maracanã 229 - Maracanã - 20271-110 - Rio de Janeiro - RJ - Brasil
llima@cefet-rj.br

Virgílio José Martins Ferreira Filho
Programa de Engenharia de Produção – COPPE/UFRJ
Caixa Postal 68507, CEP 21945-970, Rio de Janeiro, RJ, Brasil
virgilio@ufrj.br

RESUMO
Dutos são reconhecidos como o modo mais seguro e econômico de se transportar o petróleo e seus derivados, principalmente quando grandes quantidades de produtos devem ser bombeadas por longas distâncias. Neste trabalho abordamos o escalonamento de curto-prazo de um sistema composto por vários derivados de petróleo que devem ser transportados de uma única refinaria para vários terminais, conectados a mercados consumidores locais, por meio de um único duto multiproduto. Propomos uma formulação por programação inteira (IP) e uma meta-heurística do tipo Variable Neighborhood Search (VNS) para comparar as abordagens exata e heurística para o problema. Testes computacionais foram realizados nas linguagens C e MOSEL® XPRESS-MP® sobre um caso real do sistema de oleodutos brasileiro.

PALAVRAS CHAVE. Escalonamento de duto multiproduto, Programação inteira, Meta-heurística VNS.

Área principal PG – Petróleo e gás

ABSTRACT

Pipeline are known as the most reliable and economical mode of transportation for petroleum and its derivatives, especially when large amounts of products have to be pumped for large distances. In this work we address the short-term schedule of a pipeline system comprising the distribution of several petroleum derivatives from a single oil refinery to several depots, connected to local consumer markets, through a single multi-product pipeline. We propose an integer linear programming (IP) formulation and a variable neighborhood search (VNS) metaheuristic in order to compare the performances of the exact and heuristic approaches to the problem. Computational tests in C language and MOSEL® XPRESS-MP® language are performed over a real Brazilian pipeline system.

KEYWORDS. Multiproduct pipeline scheduling problem, Integer programming, VNS metaheuristic.
1. Introduction

Pipelines are known as the most reliable and economical mode of transportation for petroleum and its derivatives, especially when large amounts of products have to be pumped for large distances. We address the short-term schedule of a pipeline system comprising the distribution of several petroleum derivatives from a single oil refinery to several depots, connected to local consumer markets, through a single multi-product pipeline. The major difficulties faced in these operations are related to the satisfaction of product demands by the various consumer markets, and operational constraints such as the maximum sizes of contiguous pumping packs, and the immiscible products.

Several researchers have developed models and techniques for this short-term pipeline scheduling problem (Souza Filho et al. 2006). Two different methodologies have been proposed in the literature: heuristic search techniques and exact methods. Among the heuristic methods we can cite: knowledge-based search techniques (Sasikumar, Prakash, Patil, & Ramani, 1997), GRASP, (Milidiú & Pessoa, 2001), genetic algorithms (Sangineto, 2006) and VNS - variable neighborhood search (Souza Filho, 2007). The exact methods rely on mixed-integer linear mathematical programming (MILP) formulations. Depending on whether or not the pipeline volume and the time horizon are both discretized, model-based scheduling methods can be grouped into two classes: discrete and continuous MILP approaches. Most of the proposed optimization models not only partitioned the horizon into time intervals of equal or unequal sizes but also the pipeline volume is divided into a significant number of single-product packs (Magatão, Arruda, & Neves, 2004; Neiro & Pinto, 2004; Rejowski & Pinto, 2003, 2004; MirHassani, & Ghorbanalizadeh, 2008). In contrast, Cafaro and Cerdà (2004, 2008) and Rejowski & Pinto (2008) present MILP continuous formulation. In Milidiú & Pessoa (2003), the problem of finding a feasible solution to pipeline petroleum transportation is proved to be NP-hard.

We propose an integer linear programming (IP) formulation, relying on a uniform discrete time representation and a set of logical linear disjunctions. We also use a variable neighborhood search (VNS) metaheuristic to yield valid upper bounds for the problem. Computational tests are performed over a real Brazilian pipeline system having five products and five depots. We consider time horizons of one week or 168 hours. A branch-and-bound algorithm from Xpress-MP® was able to find a near-optimal solution for all the week time horizons.

2. Problem Description

In this work we consider the short-term scheduling problem where a single refinery must distribute $P$ petroleum products among $D$ depots connected to a single multi-product pipeline during a $T$ time horizon. The pipeline system is represented in Figure 1.

![Figure 1: General distribution pipeline system.](image)

In the refinery and in the distribution depots several tanks store the same product, although at most one of these is connected to the pipeline at each time. We assume the refinery does not need to control its inventory levels. The depots have to control their inventory levels and satisfy product demands determined by the local consumer markets. The maximum inventory levels for each depot, concerning each product, are expressed as maximum continuous pumping hours.
The pipeline is discretized into packs that we call dominoes. Each domino \( D_{p,d} \) is an ordered pair \((P_p, D_d)\) establishing that depot \(d\) receives one pumping hour of product \(p\). A set of dominoes in sequence yields a schedule. We define a hyper-domino \( D^\tau_{p,d} \) (or a \(\tau\)-domino) as the set of \(\tau\) contiguous pairs of dominoes \((P_p, D_d)^1, \ldots, (P_p, D_d)^\tau\) establishing that depot \(d\) receives \(\tau\) successive pumping hours of product \(p\).

The problem consists in determining the pair \((P_p, D_d)\) for each pumping hour of the time horizon, satisfying all the system constraints at minimum operation cost. The operation cost includes pumping costs, peak time pumping costs, inventory costs at the depots, and transition costs between miscible products inside the pipeline. Pumping costs are proportional to the amount of each product sent by the refinery and to the distance it covers in the pipeline until reach its destination depot. Inventory costs are related to the stored amounts of products at each depot. When operating multi-product pipelines production contamination is inevitable. This occurs at the interface of two miscible products. There is one cost for each pair of interfaced products that accounts for losses as well interface reprocessing at the refinery. These are the transition costs and they are usually very high.

The operational constraints refer to demand matching at each depot, upper bounds on tank capacities (translated into maximum number of contiguous packs or dominoes) and the control of immiscible products. There are some products that are not miscible. This yields forbidden sequences of products inside the pipeline that must also to be considered in the problem.

3. Integer Programming Formulation

We consider the following main assumptions in the optimization models:

- Demands are known during the time horizon;
- Product rates are compatible to demand rates;
- All tanks are treated as aggregated capacities;
- At most one tank at the refinery and at all the depots is connected to the pipeline at any time;
- The refinery does not need to control its inventory levels;
- The pipeline packs (dominoes) are always completely filled;
- All products have constant densities;
- Setup times for switching between tanks in the refinery are not considered.

The nomenclature for the proposed optimization models is listed below.

Indices:

- Products: \( p \in \{1, 2, \ldots, P\} \);
- Depots: \( d \in \{1, 2, \ldots, D\} \);
- Time intervals: \( t \in \{1, 2, \ldots, T\} \) (each time interval corresponds to one pumping hour);
- Peak time intervals: \( t \in P_{17h}^{20h} T \) (from 17hs to 20hs each day: the pumping costs are 10% higher).

Parameters:

- Pumping unit cost of sending product \(p\) to depot \(d\): \( PC_{p,d} \);
- Inventory unit cost of storing product \(p\) on depot \(d\): \( IC_{p,d} \);
- Transition cost from product \(p\) to product \(p'\): \( TC_{p,p'} \);
- Demand of product \(p\) in depot \(d\): \( Dem_{p,d} \);
- Maximum number of allowed contiguous packs (dominoes) of product \(p\) to depot \(d\): \( MCP_{p,d} \);
- Non matching demand cost matrix of product \(p\) on depot \(d\): \( NM DC_{p,d} \);
- Prohibited interfaces (forbidden sequences of products inside the pipeline): Boolean \( PI_{p,p'} \).
Binary variables:
- $x_{p,d,t} = 1$ if product $p$ is sent to depot $d$ at time $t$; $x_{p,d,t} = 0$ otherwise;
- $y_{p,p',t} = 1$ if there is an interface between products $p$ and $p'$ (product $p$ is sent at time $t$ and product $p'$ is sent at time $t + 1$); $y_{p,p',t} = 0$ otherwise.

Now, we are able to present the objective functions and the constraints that we used to propose the integer linear programming (IP) formulations. Four different scenarios, with different arrangements of objective function and constraints were developed, in order to solve the multiproduct pipeline scheduling problem in the case study that will be presented at section 5.

3.1 Objective Functions

The objective function $Z_1$ considers that all demands are matched in the depots, while in $Z_2$ there is a term related to the penalty of no matching the demand in the depot.

$$Z_1 = \sum_p \sum_d \sum_t (PC_{p,d} + IC_{p,d})x_{p,d,t} + \sum_p \sum_d \sum_t \sum_{t' \in P_{eakT}} (0.1 \times PC_{p,d})x_{p,d,t} + \sum_{p,p',t} TC_{p,p'}y_{p,p',t}$$

$$Z_2 = \sum_p \sum_d \sum_t (PC_{p,d} + IC_{p,d})x_{p,d,t} + \sum_p \sum_d \sum_t \sum_{t' \in P_{eakT}} (0.1 \times PC_{p,d})x_{p,d,t} + \sum_{p,p',t} NMDC(p,d) \times svdem(p,d)$$

3.2 Constraints

Equations (3) state that at each time interval we can have just one product delivered to some depot. Equations (4) and (5) are mutually exclusive and will be used or not, depending on the referred scenario: equalities (4) assert that the demands are matched at all the depots while equalities (5) allow to non match the demand at the depots. Inequalities (6) detect when product interfaces are generated between miscible products. Inequalities (7) assert that immiscible products do not yield prohibited interfaces. Inequalities (8) model the maximum number of allowed contiguous packs (dominoes) of each product $p$ to each depot $d$. Constraints (9) and (10) state that variables $x$ and $y$ are binary. Constraint (11) states that slack variables $svdem$ are positive.

$$\sum_p x_{p,d,t} = 1, \forall t$$

$$\sum_{t=1}^T x_{p,d,t} = Dem_{p,d}, \forall p,d$$

$$\sum_{t=1}^T x_{p,d,t} + svdem_{p,d} = Dem_{p,d}, \forall p,d$$

$$y_{p,p',t} - x_{p,d,t} - x_{p',d,t+1} \geq -1, \forall p,p' > p,d,t$$

$$x_{p,d,t-1} + x_{p',d',t} \leq 1, \forall p,p' > p,d,t > 1, PI(p,p') = true$$

$$\sum_{t=1}^{MCP_{p,d}+1} x_{p,d,t+t-1} \leq MCP_{p,d}, \forall p,d,t = 1, \ldots, T - MCP_{p,d}$$

$$x_{p,d,t} \in \{0,1\}, \forall p,d,t$$

$$y_{p,p',t} \in \{0,1\}, \forall p,p' \neq p,t$$

$$svdem_{p,d} \geq 0, \forall p,d$$

4. VNS Metaheuristic

The metaheuristic Variable Neighborhood Search (VNS) was proposed by Hansen and Mladenovic (1997). This procedure has been widely applied in several problems of combinatorial optimization, graph theory and spectral graph theory. For example: traveling salesman, $p$-median,
weighted maximum satisfability, quadratic assignment, maximal clique, graphs with maximum energy, graphs with maximum algebraic connectivity.

The main idea of the VNS consists in a systematic change of neighborhoods during the search. Contrary to other metaheuristics based on local search methods, VNS does not follow a trajectory but explores increasingly distant neighborhoods of the current solution, and jumps from this solution to a new one if and only if an improvement has been made. Moreover, a local search procedure, possible using several neighborhoods, is also applied in order to search for local optimal solutions.

Let $N_k(x)$ be the set of neighbor solutions from $x$, where $k=1,\ldots,k_{max}$. Below we show the basic VNS scheme.

**Initialization**: Select the set of neighborhood structures $N_k$, $k = 1,\ldots, k_{max}$, that will be used in the search; find an initial solution $x$; choose stopping condition.

**Repeat** the following until the stopping condition is met:

- Set $k = 1$;
- **Repeat** the following steps until $k = k_{max}$:
  - **Shaking**: generate a random solution from the $k$th neighborhood of $x$ ($x' \in N_k(x)$);
  - **Local Search**: Apply some local search method with $x'$ as initial solution; denote with $x''$ the so obtained local optimum;
  - **Move or not**: If this local optimum is better than the current, move there ($x := x''$), and continue the search with $N_1$ ($k := 1$); otherwise, set $k := k + 1$.

In this work we consider $k=3$, with the following nested neighborhoods: 2-opt with first improvement, joint dominoes and 2-opt.

The 2-opt neighborhood consists of all different permutations of two dominoes in the schedule, and yields $n(n-1)/2$ neighbors. The 2-opt with the first improvement neighborhood is quite similar to 2-opt, however, in this case permutations between two dominoes are made just until a better solution is found.

The joint dominoes neighborhood uses problem’s structure to reduce the interface costs. The main purpose is to group dominoes with the same characteristics: given a domino, the idea is to find the next domino of the same type and put them together (given the $i$-th domino, suppose that the next domino of the same type is found in the $k$-th position, then the dominoes in the $(i+1)$-th and in the $k$-th positions are permuted).

5. Case Study

The real Brazilian pipeline system studied in this work is represented in Figure 2 below.

![Figure 2: Case study: real Brazilian distribution pipeline system.](image-url)
We have developed four different scenarios to the problem, in order to analyze different operational situations. We describe each of these scenarios below.

5.1 Scenario 1

In this scenario, in order to send the five products to the five depots we consider the following costs: pumping costs, inventory costs, transition costs, peak time costs and suppose that the demand in each depot must be matched. Further, we do not bound the maximum number of allowed contiguous packs (dominoes) of each product \( p \) to each depot \( d \), i.e., we suppose that it is possible to send a pack of the product \( p \) to the depot \( d \) as big as the demand of that product \( p \) in that depot \( d \). So, the model to be optimized in this case is:

\[
\begin{align*}
\text{Min } & \ Z_1 \\
\text{s.t.} & \ (3),(4),(6),(7),(9),(10)
\end{align*}
\]

5.2 Scenario 2

In this scenario, we consider the same costs involved in the Scenario 1 and we also suppose that the sum of the demands in each depot (total demand) is greater than the planning horizon; it means that it is impossible to match the demands to all depots. Consequently, there is a cost associated to the non matching demand. The model to be optimized in this case is:

\[
\begin{align*}
\text{Min } & \ Z_2 \\
\text{s.t.} & \ (3),(5),(6),(7),(9),(10),(11)
\end{align*}
\]

5.3 Scenario 3

In this scenario, we consider the same costs involved in the Scenario 1 and we suppose that the demand in each depot is matched. Here we consider a constraint related to the maximum contiguous pack size, i.e., the maximum contiguous pack size sent of the product \( p \) to the depot \( d \) is given by \( MCP_{p,d} \). The model to be optimized in this case is:

\[
\begin{align*}
\text{Min } & \ Z_1 \\
\text{s.t.} & \ (3),(4),(6),(7),(8),(9),(10)
\end{align*}
\]

5.4 Scenario 4

In this scenario, we consider the same costs involved in the Scenario 1. We also suppose that the sum of the demands in each depot (total demand) is greater than the horizon planning, and the constraints related to the maximum contiguous pack size are added. In this case, the optimization model is described below.

\[
\begin{align*}
\text{Minimize } & \ Z_2 \\
\text{s.t.} & \ (3),(5),(6),(7),(8),(9),(10),(11)
\end{align*}
\]

In the next section we present the computational results obtained to these scenarios.

6. Computational Results

We have used the MOSEL language to model and the XPRESS IP pure Branch-and-Bound solver to solve our IP model. The VNS metaheuristic was implemented in C language. The computational tests were performed over an AMD Athon of 2.00 GHz with 1 GB of memory RAM. The number of integer variables is 8400 and the number of constraints is around 9000. The stopping condition for the VNS and the IP model was two hours of computation.

In the first scenario the solution founded by the VNS was worse than the IP solution and the magnitude of this difference was of 0.28%. However, the processing time of the VNS was smaller than IP solution. The gap between the lower bound and the IP solution was 0.34%.

In the second scenario the solution founded by the VNS was better than the IP solution and the magnitude of this difference was of the order of 0.26%. The processing time of the VNS was also better (smaller). The gap between the lower bound and the IP solution was of 0.80%.
In the third scenario the solution founded by the VNS was better than the IP solution and the magnitude of this difference was of the order of 0.17%. The processing time of the VNS was also better (smaller). The gap between the lower bound and the IP solution was of 0.86%.

In the fourth scenario the solution founded by the VNS was better than the IP solution and the magnitude of this difference was of the order of 0.16%. The processing time of the VNS was also better (smaller). The gap between the lower bound and the IP solution was of 0.75%.

Table 1 summarizes the results obtained to the scenarios 1, 2, 3 and 4.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>VNS Solution</th>
<th>VNS Time (s)</th>
<th>IP Solution</th>
<th>IP Time (s)</th>
<th>IP Lower Bound (LB)</th>
<th>VNS distance to LB (%)</th>
<th>IP distance to LB (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>2.782.483,95</td>
<td>1368,40</td>
<td>2.774.820,30</td>
<td>7200</td>
<td>2.765.284,00</td>
<td>0,62</td>
<td>0,34</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>2.816.588,45</td>
<td>1742,69</td>
<td>2.824.055,61</td>
<td>7200</td>
<td>2.801.578,75</td>
<td>0,54</td>
<td>0,80</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>2.784.383,95</td>
<td>2647,25</td>
<td>2.789.094,25</td>
<td>7200</td>
<td>2.765.284,00</td>
<td>0,69</td>
<td>0,86</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>2.818.088,50</td>
<td>4064,89</td>
<td>2.822.558,82</td>
<td>7200</td>
<td>2.801.578,75</td>
<td>0,59</td>
<td>0,75</td>
</tr>
</tbody>
</table>

7. Conclusions and Future Work

The pipeline transportation problem has been very studied in the face of its practical importance. To get an idea, in 2008 have been published at least 4 articles on the subject. In this context, we proposed an integer linear programming (IP) formulation and a Variable Neighborhood Search (VNS) metaheuristic in order to compare the performances of the exact and heuristic approaches to the problem. In our preliminary studies, we obtained feasible solutions to the multiproduct pipeline product problem on both methodologies in a reasonable computational time. The results achieved by VNS and IP model are very close to each other and also very closer to the lower bound given by the Xpress-MP© B&B algorithm.

The intention in this work was to validate the use of the VNS metaheuristic to solve the problem approximately. Since we did not have any lower bound on the problem it was impossible to evaluate the quality of the VNS solutions. Therefore, with this first IP approach via the pure B&B we were also able to evaluate the quality of the bounds generated by the IP model. It is important to emphasize that, in despite of the apparently small dimension of the case study (5 depots and 5 products), we have 8400 integer variables and around 9000 constraints.

The inclusion of the constraints related to the maximum contiguous pack size turns the obtaining of feasible solutions much more difficult. Thus, in a future work it is indispensable to implement a branch-and-cut and to study the integer programming polyhedron, in order to solve the real problem to optimality. Another relevant topic is to examine other different operational settings.

Acknowledges

The authors are supported by CNPq (Universal 474927/2007-5); second author is also supported by FAPERJ (Proc. 2007/170.479).

References


