Planning of Pipeline Oil Transportation with Interface Restrictions is a Difficult Problem

Ruy Luiz Milidiú

milidiu@inf.puc-rio.br

Frederico dos Santos Liporace

liporace@inf.puc-rio.br

PUC-RioInf.MCC56/03 December, 2003

Abstract: An important constrain when developing a schedule for the operation of an oil pipeline is the interface between adjacent products. Due to the resulting quality loss, some products are not allowed to be adjacent inside the pipeline. The S-PPI decision problem asks about the existence of a feasible pipeline operations sequence that takes into account this additional restriction. We show that S-PPI is $\mathcal{NP}$-complete. An immediate implication from this finding is that the planning of pipeline transportation with interface restrictions for general topology pipeline networks is also difficult.

Keywords: pipeline transportation, complexity, algorithms, interface, planning, pipesworld

Resumo: Uma importante característica a ser considerada durante a geração de uma programação para uma rede de oleodutos é a interface entre produtos adjacentes. Como alguns produtos reconhecidamente geram perdas consideráveis, estes não devem ser adjacentes no oleoduto. O problema de decisão S-PPI pergunta sobre a existência de uma seqüência viável de operações em um oleoduto, considerando as restrições de interface. Nós mostramos que o S-PPI é $\mathcal{NP}$-completo. Uma implicação imediata deste fato é que o planejamento de transporte em oleodutos, considerando as restrições de interface, é também difícil para redes de topologia genérica.

Palavras-Chave: transporte em oleodutos, complexidade, algoritmos, interface, planejamento, pipesworld

1 Introduction

Pipelines play an important role in the transportation of Petroleum and its derivatives, since it is the most effective way to transport large volumes over large distances.

The main components of a pipeline network are operational areas and pipeline segments. Operational areas may be distribution centers, ports or refineries. These areas are connected by one or more pipeline segments. The oil derivatives are moved between the areas through the pipelines.

These pipeline networks may be very long and complex. An example of a company dedicated to pipeline management is Transpetro, the transportation company of Petrobras, the Brazilian state-run oil company. Altogether, Transpetro operates more than 6700 kilometers of pipelines. Figure 1 shows the topology of a pipeline network with thirteen areas and twenty eight pipeline segments operated by Transpetro.

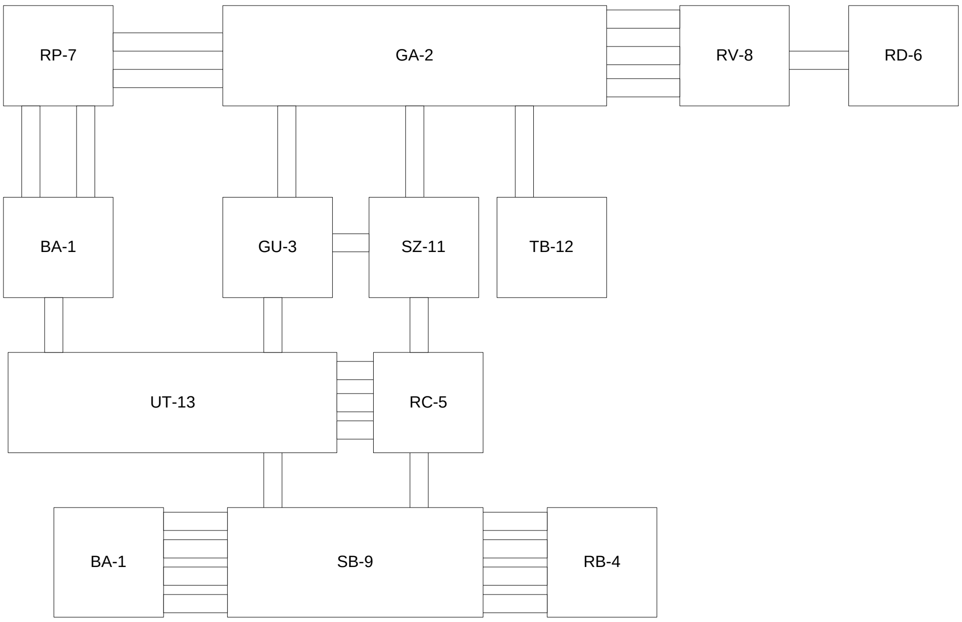

Figure 1: Example of a pipeline network operated by Transpetro.

Typically, oil pipelines are a few inches wide and several miles long. As a result, reasonable amounts of distinct products can be transported through the same pipeline with a very small loss due to the mixing at liquid boundaries.

Optimizing the transportation through oil pipelines is a problem of high relevance, since a non negligible component of a petroleum product's price depends on its transportation cost. Nevertheless, as far as we know, just a few authors have specifically addressed the problem [5, 1, 9]. In [7] they show that finding a feasible solution for a model that includes due dates is $\mathcal{NP}$-hard. They also propose a polynomial time algorithm to solve instances, where all the products to be delivered are initially located in the areas, not inside the pipelines. In [6] they discuss the approximability of the problem of finding the smaller set of pumping operations that move the pipeline to a goal state, and propose an approximate algorithm if the pipeline network is acyclic.

A PDDL [3] formulation for the pipeline transportation problem that uses unitary batches has been proposed [8], and is being considered to be used for a benchmark for the upcoming International Planning Competition [2].

A valid operation plan for an oil pipeline must typically consider restrictions such as the maximum and minimum storage for products and due dates for production and delivery. In this paper we focus on the interface restriction, and introduce the S-PPI problem, which considers only the interface restriction for a two nodes pipeline, as shown in Figure 2. We show that S-PPI is $\mathcal{NP}$-complete.

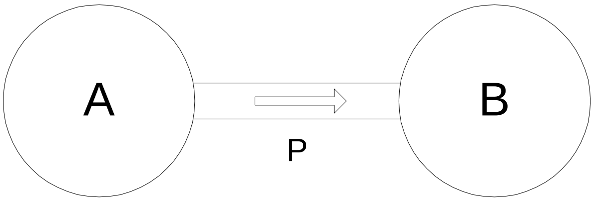

Figure 2: Pipeline network example.

This paper is organized as follows. Section 2 details the pipeline transportation problem. In section 3, we define the S-PPI problem. In section 4, we show that S-PPI is $\mathcal{NP}$-complete. Finally, in section 5, we present our conclusions.

2 Pipeline Transportation

The pipeline transportation problem has an unique characteristic, what distinguishes it from other transportation methods: it uses stationary carriers whose cargo moves rather than the more usual moving carriers of stationary cargo.

Pipelines may be divided in two major groups, based on the nature of their cargo: those that transport liquid and those that transport gas. Here, we focus on the liquid pipeline transportation problem. More specifically, we examine multi-commodity liquid pipelines, where more than one product may be transported. We use the term product to refer to petroleum derivatives, such as diesel, gasoline and aviation kerosene.

2.1 Pressurized pipelines

For safety reasons, liquid pipelines must be always pressurized. That is, they must be completely full of liquid. A typical operation in a pipeline segment P that connects areas A and B is pumping a product from one area to the other. Figure 2 illustrates the pumping of product from A to B. Assuming incompressible fluids, the same amount of liquid pumped from A is received in B.

Since the pipeline segment may be filled with distinct products, the product that area B receives is not necessarily the same one that area A pumps. For instance, if the pipeline segment is initially filled with diesel, and area A pumps some amount of gasoline, B initially receives the diesel that is stored in the pipeline segment. Moreover, B only begins to receive the gasoline originated from A after receiving all the diesel volume that is initially in S.

2.2 Interface

When distinct products have direct contact inside the pipeline segment, there is some unavoidable product quality loss due to the mixture in the interface between them. These interface losses are a major concern in pipeline operation. The mixed products can not be simply discarded, they must go through a special treatment that usually involves sending them back to a refinery where they require special tanks.

The severity of interface losses depends on the products that interface inside the pipeline segment. If two products are known to generate high interface losses, the pipeline schedule must not place them adjacently into the segment.

The interface restrictions model the interface requirements. An interface graph G is defined for the products that are moved through the pipeline. In this graph, each product P_i is represented by a node. An edge $\{P_i, P_j\}$ in G indicates that the interface losses between P_i and P_j are acceptable. Otherwise, if the edge $\{P_i, P_j\}$ does not exist in G, it means that the interface losses between P_i and P_j are unacceptable. We say that $P_i \sim P_j$ iff P_i may interface with P_j.

Figure 3 presents an interface graph example. In the example, the allowable interfaces are given by $P_1 \sim P_2$, $P_1 \sim P_3$, $P_2 \sim P_3$, $P_3 \sim P_4$ and $P_3 \sim P_5$.

3 The S-PPI problem

This section introduces the Simple Pipeline Planning with Interface (S-PPI) model. In this problem we have two areas, namely A and B, connected by the single pipeline segment P, as shown in Figure 4.

The flow direction in P is from A to B, as indicated by the arrow above P.

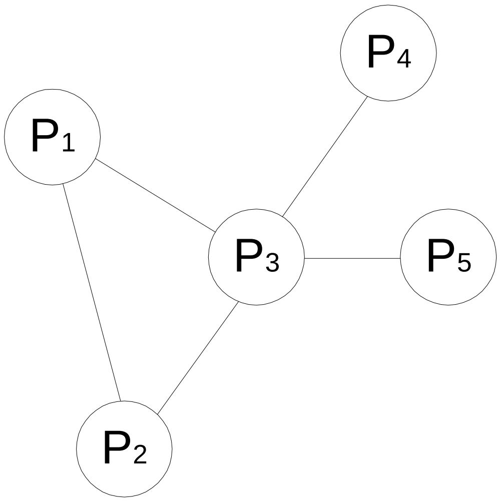

Figure 3: An interface graph example.

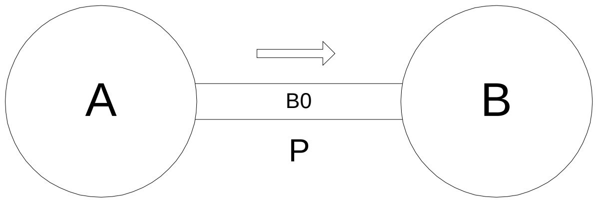

Figure 4: The S-PPI model initial state.

The S-PPI problem also has a set $\mathcal{B}= \{b_0, \ldots, b_n\}$ of batches, with a different product for each batch. The batch volumes are represented by v_i, with $i = 0, \ldots, n$, and the pipeline segment volume as V. In S-PPI, we take the *unitary volumes assumption*, that is, all batches have the same volume. We go further and define the batch volumes to be equal to the pipeline segment volume, that is, $v_i = V$ for $i = 0, \ldots, n$. We say that $b_i \sim b_j$ iff b_i may interface with b_j. Batches b_0 and b_n are special, since their interface restrictions are $b_0 \sim b_k$ and $b_n \sim b_k$, for $k = 0, \ldots, n$.

In S-PPI, the pipeline state is represented by $S = (\beta_A, b_P, \beta_B)$, where β_A and β_B are the set of batches that are located in areas A and B, respectively, and b_P is the batch that is inside P.

Pipeline states may be changed by an operation named $Pump(S, b)$, that represents pumping batch b from area A into P. Given a state $S_j = (\beta_A, b_P, \beta_B)$ and a batch $b \in \beta_A$, the $Pump(S_j, b)$ operation changes the system to a state S_{j+1} such that $S_{j+1} = (\beta_A - \{b\}, b, \beta_B \cup \{b_P\})$. Since all the batch volumes are equal to the pipeline volume, at the end of each $Pump$ operation the batch b_P that was previously inside P is completely moved to B, and the pumped batch b is completely inside P. In addition, $b \sim b_p$ must hold to satisfy the interface restriction.

The S-PPI problem is to find a sequence L of pumping operations that move the system from the initial state $S_0 = (\{b_1, \dots, b_n\}, b_0, \emptyset)$ to the goal state $S_g = (\emptyset, b_n, \{b_0, \dots, b_{n-1}\})$.

The S-PPI decision problem is to answer the question "Is there a sequence L of pumping operations that moves the system from S_0 to S_g?"

4 S-PPI is $\mathcal{NP}$-Complete

In this section, we show that S-PPI contains the Hamiltonian Path, a known $\mathcal{NP}$-complete problem [4], as a special case. First, let us show that S-PPI is in $\mathcal{NP}$.

Theorem 1 *S-PPI is in $\mathcal{NP}$.*

Proof: The number of *Pump* operations in any certificate C is always equal to the number of batches in the instance minus one. Hence, the size of C grows linearly with the input size. To verify C, we must simply check if the interface restrictions are satisfied for all *Pump* operations and if the resulting final state is equal to S_g. Therefore, the overall checking takes polynomial time. ∎

In the Hamiltonian Path (HP) problem, we are given a graph $G = (V, E)$ and must answer if G contains or not a Hamiltonian path, that is, a simple path that contains all vertices $(i_1, i_2, \dots, i_k)$ in G such that $\{i_l, i_{l+1}\} \in E$, for $l = 1, \dots, k-1$.

Theorem 2 *HP $\propto$ S-PPI*

Proof: Given a graph $G = (V, E)$, with $|V| = n$, we build a S-PPI instance Π as follows. For each $i \in V$, with $i = 1, \dots, n$ we create a batch b_i in Π. We also define two extra batches b_0 and b_{n+1}. The initial state is defined as $S_0 = (\{b_1, \dots, b_{n+1}\}, b_0, \emptyset)$, and the goal state is $S_g = (\emptyset, b_{n+1}, \{b_0, \dots, b_n\})$. The allowable interface between batches in Π is such that $b_i \sim b_j$ iff $\{i, j\} \in E$. In addition, batches b_0 and b_{n+1} may interface with all the other batches, that is, $b_0 \sim b_i, b_{n+1} \sim b_i$, for $i = 1, \dots, n$.

Figure 5 shows an example on how the set of batches in Π is created from G.

To show that this transformation can be performed in polynomial time, it suffices to observe that the number of batches and interface restrictions in Π is bounded by a polynomial in $|V|$ and $|E|$, and their construction follows directly from a traversal of G.

We now show that Π admits a solution L if and only if G has a Hamiltonian path.

Suppose first that we have a solution L for Π. Hence, we show that G has a Hamiltonian path. For all operations in L, it must be observed that the inserted batch b_i must be allowed to interface with the batch b_P that was previously in P, with b_i replacing b_P in P after the operation is completed.

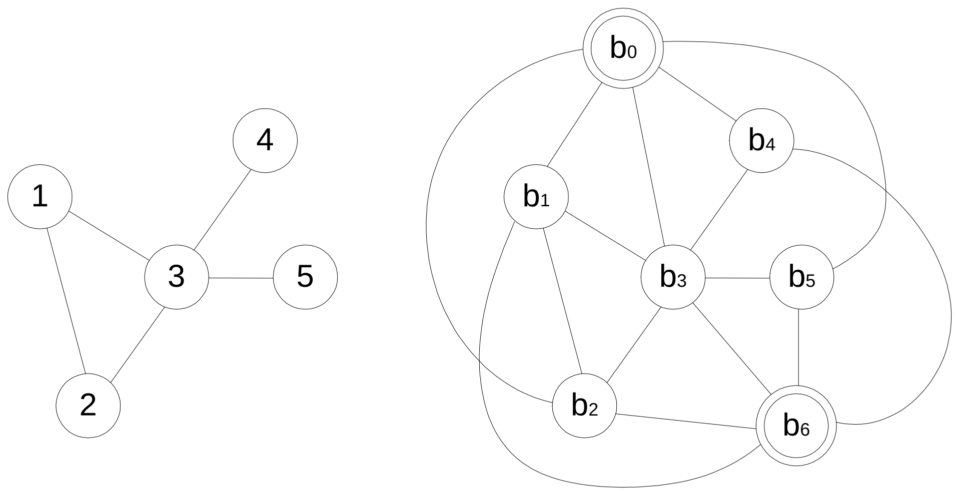

Figure 5: Example of Π batch set construction. G is shown on the left, and the Π batch set is shown on the right.

The first operation is always possible, since b_0 interfaces to all batches in Π.

Since batches $b_1, \ldots, b_n$ must be placed in B, the last batch to be pumped in L must be b_{n+1}, in order to push the previously inserted batch from P to B.

Each batch b_i pumped in an operation in L, except for the first and last ones, may be viewed as taking the arc $\{i, j\}$ in E, where b_j is the batch pumped in the previous operation. Since all batches must be moved, and every batch is composed by a distinct product, all nodes in G must be visited. Since the batches can not be split, each node must be visited only once. Therefore, these operations define a Hamiltonian path in G.

Now we show that if G contains a Hamiltonian path then Π has a solution L. We take $Pump(S_0, b_i)$, where b_i is the batch associated to the first node in the Hamiltonian path, as the first operation in L. This operation is always possible since b_0, inside P at start time, may interface to all batches. We proceed by pumping the remaining batches, in the same order that their counterpart node in G appears in the Hamiltonian path. These operations also do not violate the interface restriction. Finally, the last operation in L is pumping b_{n+1}. The final state is the S-PPI goal state, $S_g = (\emptyset, b_{n+1}, \{b_0, \ldots, b_n\})$. ■

Theorem 3 *S-PPI is $\mathcal{NP}$-complete*

Proof: S-PPI is $\mathcal{NP}$-complete if the following two conditions hold: (i) S-PPI is in $\mathcal{NP}$; (ii) S-PPI is $\mathcal{NP}$-hard. Theorem 1 states (i), whereas Theorem 2 states (ii), what concludes the proof. ■

An immediate implication from this finding is that the planning of pipeline transportation with interface restrictions for general topology networks is also difficult.

5 Conclusions

In this paper, we introduce the S-PPI problem, inspired on the planning of petroleum pipelines. This model captures the interface restriction, one of the many characteristics of this domain. We show that S-PPI is $\mathcal{NP}$-complete. This result, in addition to other already obtained complexity results on this domain, is an indication of the challenge posed on the task of deriving automatic planners for the domain. An interesting approach to achieve this goal is to develop a dedicated solver, with domain specific heuristics.

Some open problems include finding the complexity of formulations that consider only tank capacity restrictions and that allow reversions in the pipeline flow.

Furthermore, the development of alternative PDDL models for the domain and the corresponding measurement of the performance of state-of-the-art general purpose solvers for them could give some insight on the problem difficulty.

References

[1] Eduardo Camponogara. A-teams para um problema de transporte de derivados de petróleo. Master's thesis, Departamento de Ciência da Computação, IMECC - UNICAMP, December 1995.

[2] Stefan Edelkamp and Jörg Hoffmann. Quo vadis, IPC-4? — proposals for the classical part of the 4th international. In *Workshop on the Competition, Trento, Italy*, June 2003.

[3] M. Fox and D. Long. PDDL 2.1: An extension to PDDL for expressing temporal planning domains, 2001.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

[5] Christopher A. Hane and H. Donald Ratliff. Sequencing inputs to multi-commodity pipelines. *Annals of Operations Research*, 57, 1995. Mathematics of Industrial Systems I.

[6] Ruy L. Milidiú, Artur A. Pessoa, and Eduardo S. Laber. Complexity of makespan minimization for pipeline transportation of petroleum products. In *Proceedings of the APPROX'2002*, pages 243–255, Rome, Italy, 2002. Accepted for publication in TCSA.

[7] Ruy L. Milidiú, Artur A. Pessoa, and Eduardo S. Laber. Pipeline transportation of petroleum products with no due dates. In *Proceedings of the LATIN'2002*, pages 248–262, Cancún, Mexico, April 2002.

[8] Ruy Luiz Milidiú, Frederico dos Santos Liporace, and Carlos José P. de Lucena. Pipesworld: Planning pipeline transportation of petroleum derivatives. In *Workshop on the Competition, Trento, Italy*, June 2003.

[9] Artur Alves Pessoa. *Dois problemas de otimização em grafos: Transporte em redes de dutos e Busca com custos de acesso*. PhD thesis, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, May 2003.