Complexity of Makespan Minimization for Pipeline Transportation of Petroleum Products [1]

Ruy Luiz Milidiú
milidiu@inf.puc-rio.br
Artur Alves Pessoa
artur@inf.puc-rio.br
Eduardo Sany Laber
laber@inf.puc-rio.br

PUC-RioInf.MCC02/02 March, 2002

Abstract: PTP is a new model for the problem of transporting petroleum products through pipelines with no due dates. This model uses a directed graph G, where arcs represent pipes and nodes represent locations, and a set L of transportation orders. Since pipelines must always be full, we define a subset $F \subset L$ of *further* orders that do not need to reach the corresponding destination nodes, being used only to move the pipeline contents. The problem of finding a feasible solution to PTP is known to be $\mathcal{NP}$-hard, even if G is acyclic.

The Synchronous PTP (SPTP) is a special case of PTP where all orders in F are initially stored at nodes. In this paper, we analyze the complexity of finding a minimum makespan solution to SPTP. This problem is called the SPTMP. We prove that, for any fixed $\epsilon > 0$, there is no $\eta^{1-\epsilon}$-approximate algorithm for SPTMP unless $\mathcal{P} = \mathcal{NP}$, where η is the input size. This result also holds if the graph G is both planar and acyclic.

Keywords: Petroleum, Transportation, Oil pipeline, Complexity, Approximability

Resumo: PTP é um novo modelo para o transporte de derivados de petróleo através de oleodutos. Este modelo utiliza um grafo dirigido G, onde os arcos representam trechos de duto e os nós representam áreas operacionais, e um conjunto L de ordens de transporte. Como os oleodutos devem permanecer cheios de produto, definimos um subconjunto $F \subset L$ de ordens *futuras* que podem não chegar aos seus respectivos nós de destino, sendo utilizadas apenas para deslocar o conteúdo dos dutos. Sabe-se que encontrar uma solução viável para o PTP é um problema $\mathcal{NP}$-difícil, mesmo quando G é acíclico.

SPTP (o PTP síncrono) é um caso particular do PTP onde todas as ordens futuras estão inicialmente armazenadas em nós. Neste documento, analisamos a complexidade de encontrar uma solução para o SPTP que minimiza o tempo total de movimentação dos oleodutos (*makespan*). Este problema é chamado de SPTMP. Demonstramos que, para qualquer valor fixo de $\epsilon > 0$, o SPTMP não admite nenhum algoritmo $\eta^{1-\epsilon}$ aproximado a menos que $\mathcal{P} = \mathcal{NP}$, onde η é o tamanho da entrada, em bits. Este resultado também vale se o grafo G é planar e acíclico.

Palavras-Chave: Petróleo, Transporte, Oleoduto, Complexidade, Aproximabilidade

[1]Sponsored by CTPetro/FINEP, CENPES/Petrobras, and FAPERJ.

1 Introduction

Petroleum products are typically transported through pipelines. Pipelines are different from all other transportation methods since they use stationary carriers whose cargo moves rather than moving carriers of stationary cargo. An important characteristic of pipelines is that they must be always full. Hence, assuming incompressible fluids, an elementary pipeline operation is the following: pump an amount of product into the pipeline and remove the same amount of product from the opposite side. Typically, each pipeline is 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 mixing at liquid boundaries.

Optimizing the transportation through pipelines is a problem of high relevance, since a non negligible component of a petroleum derivative's price depends on its transportation cost. Nevertheless, as far as we know, just a few authors have specifically addressed this problem [HR95, Cam95, MLPR99, MPL02]. Let us define an *order* as a requirement to transport a given amount of some product from one location to another. In [HR95], Hane and Ratliff present a model that assumes cyclic orders. In this case, the same orders always repeat after the completion of a given period of time. In [MPL00, MPL02], Milidiú, Pessoa, and Laber propose a model for pipeline transportation of petroleum products with non-cyclic orders. Let us refer to this model as the PTP (Pipeline Transportation Problem) model. PTP models a pipeline system through a directed graph G, where each of the n nodes represents a location and each of the m directed arcs represents a pipeline, with a corresponding flow direction. In this sense, PTP is more general than Hane's model, where the pipeline system must be represented by a directed tree. As in Hane's model, the flow inside each pipeline is assumed to be unidirectional.

Throughout this paper, we use the term *batch* to denote the amount of product that corresponds to a given order. Each batch is defined by both its initial position and its associated destination node, and cannot be split during transportation. The initial position of a batch may be either a node or a pipeline. In the Petroleum industry terminology, batches with fixed destination nodes are usually called *proprietary batches*. PTP assumes that all batches are proprietary, and have an unitary volume. In general, PTP allows multiple batches corresponding to the same order. In this paper, we assume an one-to-one correspondence between batches and orders. Observe that this assumption makes our lower bounds stronger since they apply to a more restricted model.

Let L be a set of r batches. Since pipelines must always be full, some of the batches cannot be delivered. In fact, they are only used to move the pipeline contents. Hence, we define a subset $F \subset L$ of *further* batches that are not necessarily delivered at the end of a feasible pumping sequence. As result, a feasible solution is a pumping sequence that delivers all *non-further* batches in $L - F$.

In [MPL02], the problem of finding a feasible solution to PTP is proved to be $\mathcal{NP}$-hard, even if G is acyclic. Moreover, the authors introduce the Synchronous PTP (SPTP), a special case of PTP where all batches in F are initially stored at nodes. The problem of finding a minimum pumping cost solution to SPTP is called SPTOP. In a previous work, we introduce the BPA algorithm, that finds feasible solutions to SPTP in a polynomial time. If G is acyclic, then these solutions are also optimal for the SPTOP.

In this paper, we analyze the complexity of finding minimum makespan solution to SPTP. This problem is called the SPTMP (Synchronous Pipeline Transportation Makespan Problem). We prove that, for any fixed $\epsilon > 0$, there is no $\eta^{1-\epsilon}$-approximate algorithm for SPTMP unless $\mathcal{P} = \mathcal{NP}$, where η is the input size. This result also holds if the graph G is both planar and acyclic. Roughly speaking, we prove that it is $\mathcal{NP}$-complete to decide whether the execution of a given pipeline operation depends on the execution of another one. After that, we show that it leads to the previous approximability bound. We believe that this approach could be used to obtain similar lower bounds on the approximability of other scheduling problems.

This paper is organized as follows. In Section 2, we describe the SPTMP. In Section 3, we prove our approximability bound. In the last section, we present our final remarks.

2 The SPTMP model

In this section, we describe the SPTMP model. Our description includes its pipeline system, orders, pipeline contents, allowed operations, and objective function.

Pipeline System

Let $G = (N, A)$ be a directed graph, where N is the set of n nodes and A is the set of m arcs. Given an arc $a = (i, j) \in A$, we say that i is the start node of a and j is the end node of a. Arcs represent pipes and nodes represent locations. Each arc $a \in A$ has an associated integer capacity $v(a)$. Moreover, we define the set of pipeline positions $A' = \{(a, l) | a \in A \text{ and } l \in \{1, \ldots, v(a)\}\}$.

Orders

Let L be a set of r unitary volume batches, where $F \subset L$ is the subset of *further* batches and $L - F$ is the subset of *non-further* batches. For each $b \in L$, it corresponds an order to deliver b at $d(b) \in N$.

Pipeline Contents

Now, let us define our representation of the pipeline contents. Here, we only consider the instants where each arc $a \in A$ contains exactly $v(a)$ integral batches. As a result, any solution to this model generates a discrete sequence of states, where the positions of all batches are well-defined.

Let us use $p_t(b)$ to denote the position of batch b at state t. If $p_t(b) = (a, l) \in A'$, then batch b is located at the lth position of arc a at state t. Otherwise, if $p_t(b) = i \in N$, then batch b is stored at node i. Furthermore, the content of a given arc a at a given state t is represented by a list of batches $[b_1, b_2, \ldots, b_{v(a)}]$, where b_l is the batch such that $p_t(b_l) = (a, l)$, for $l = 1, 2, \ldots, v(a)$.

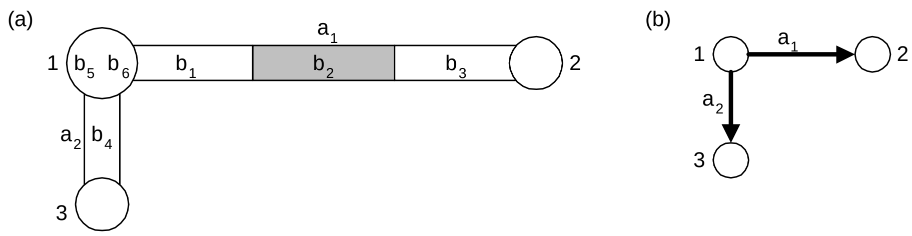

Figure 1: (a) The contents of a pipeline system; (b) the corresponding graph.

As an example, Figure 1.(a) represents the pipeline contents corresponding to the graph of Figure 1.(b). Observe that the system has two pipelines $a_1 = (1, 2)$ and $a_2 = (1, 3)$, whose flow direction is indicated by the corresponding arcs. The capacities of a_1 and a_2 are $v(a_1) = 3$ and $v(a_2) = 1$, respectively. Let us assume that Figure 1.(a) corresponds to state t. In this case, we have $p_t(b_1) = (a_1, 1)$, $p_t(b_2) = (a_1, 2)$, $p_t(b_3) = (a_1, 3)$, and $p_t(b_4) = (a_2, 1)$, since the contents of a_1 and a_2 are respectively represented by the lists $[b_1, b_2, b_3]$ and $[b_4]$. Furthermore, we have $p_t(b_5) = p_t(b_6) = 1$ since both b_5 and b_6 are stored at node 1.

At the initial state (state 0), the position $p_0(b)$ of each batch b is given. As in the SPTP, we assume that every further batch b has $p_0(b) \in N$.

Operations

A solution for the model is a set Q of elementary pipeline operations (EPO), defined as follows. Let $a = (i, j)$ be an arc of G, whose contents at a given state t are given by the list $[b_1, b_2, \ldots, b_{v(a)}]$. Moreover, let b be a batch stored at node i at this moment. An EPO (b, a, t) is to pump b into a during the time interval $[t, t+1)$. As a result of this operation, the contents of a at state $t+1$ are given by the list $[b, b_1, b_2, \ldots, b_{v(a)-1}]$ and $b_{v(a)}$ is stored at the node j. We point out that some EPO's may be simultaneously executed. Formally, given two EPO's (b_1, a_1, t_1) and (b_2, a_2, t_2), if we have $t_1 = t_2$, then we must have $b_1 \neq b_2$ and $a_1 \neq a_2$.

Let $q = \max\{t+1 | (b, a, t) \in Q\}$. Q is feasible when the following two conditions hold:

1. every batch $b \in L - F$ is stored in node $d(b)$, when the state is q;
2. for every batch $b \in F$ there is a path in G containing $p_q(b)$ and terminating at node $d(b)$.

Objective Function

The SPTMP is to find a minimum makespan set Q of EPO's. Hence, the value of q shall be minimum.

3 Complexity of SPTMP

In this section, analyze the complexity of SPTMP. Here, we also assume that the graph G is both acyclic and planar, what makes our lower bounds stronger.

Now, let us introduce some terminology. Let us use the term *source (tail) node* of $p_t(b)$ to denote:

1. the start (end) node of a if $p_t(b) = (a, l) \in A'$;
2. the node i, if $p_t(b) = i \in N$.

We say that $p_q(b)$ is a *valid final position* for $b \in F$ when there is a path in G connecting the tail node of $p_0(b)$ to the source node of $p_q(b)$ and another path connecting the tail node of $p_q(b)$ to $d(b)$. If $b \in L - F$ then $d(b)$ is the only *valid final position* for b. A *valid final state* for a pipeline system is a state where each batch $b \in L$ is located at a valid final position. Now, we are ready to present a theorem proved in [MPL02].

Theorem 1 *An instance I of the SPTMP is feasible if and only if there exists an assignment from F to A' with the following two properties:*

1. *to each pipeline position $(a, l) \in A'$, it is assigned exactly one batch of F;*

2. for every batch b assigned to $(a, l) \in A'$, (a, l) is a valid final position for b.

The previous theorem constructs an instance B of the Assignment Problem [AMO93] that is feasible if and only if I is feasible. Observe that any feasible assignment X of B gives a valid final position of each batch $b \in F$ as follows: if b is assigned to $(a, l) \in A'$, then $p_q(b) = (a, l)$. Moreover, any valid final state for I corresponds to an assignment of B. This fact is used in some of our proofs.

Finally, we say that an arc a is *reachable* to a batch b when there are both a path from $p_0(b)$ to the start node of a and another path from the end node of a to $d(b)$. Observe that a batch b can be pumped only into reachable arcs.

3.1 Hardness of Operations Dependence

Here, we prove that, for a given instance I of the SPTMP, deciding whether the execution of a given operation depends on the execution of another one is a $\mathcal{NP}$-complete problem. Formally, given an instance I of the SPTMP, two batches $b_1, b_2 \in L$, and two arcs $a_1, a_2 \in A$, the *Dependence Pipeline Problem* (DPP) is to find a feasible solution Q to I containing both (a_1, b_1, t_1) and (a_2, b_2, t_2), for $t_1 \geq t_2$. In the next theorem, we prove that DPP is $\mathcal{NP}$-complete problem by showing a polynomial reduction from the well-known Vertex Cover Problem (VCP) to DPP.

Given an undirected graph $G' = (V', E')$ and a positive integer $k' < |V'|$, VCP is to find a subset $S' \subset V'$ of vertices with $|S'| \leq k'$ such that, for all $e = (i, j) \in E'$, either $i \in S'$ or $j \in S'$ (or both). Here, we consider a special case of VCP (say 3-VCP) where every vertex degree in G' is at most 3. We point out that 3-VCP is also $\mathcal{NP}$-complete [GJ79].

Theorem 2 *DPP is $\mathcal{NP}$-complete.*

Proof: First, we prove that DPP belongs to $\mathcal{NP}$. Let I be an instance of the SPTMP. Since G is acyclic[2], for any feasible solution Q to I, each batch can be pumped into at most m arcs. Hence, Q has no more than rm EPO's. Let I' be an instance of DPP given by I, $b_1, b_2 \in L$, and $a_1, a_2 \in A$. Since any certificate to I' is also a feasible solution to I, DPP belongs to $\mathcal{NP}$.

Now, let us consider an instance of 3-VCP represented by both G' and k'. For the sake of simplicity, we number the vertices of G' from 3 to $|V'| + 2$ and the edges of G' from $|V'| + 4$ to $|E'| + |V'| + 3$. We construct a corresponding instance I' of DPP as follows:

[2]If G has one or more cycles, then whether DPP belongs to $\mathcal{NP}$ or not is an open question.

1. create $|E'| + |V'| + 6$ nodes in N, numbered from 1 to $|E'| + |V'| + 3$;

2. create the following arcs in A:

   (a) $a_1 = (1, 2)$ with initial content $[b_1^1, b_2^1, \dots, b_{|V'|-k'}^1]$, where $d(b_j^1) = 2$, for $j = 1, \dots, |V'| - k'$;

   (b) $(2, i)$ and $(i, |V'| + 3)$, for all $i \in V'$, where:

      i. $(2, i)$ has initial content $[b_{2i-5}^2, b_{2i-4}^2]$;

      ii. $(i, |V'| + 3)$ has initial content $[b_i^3]$;

      iii. each edge adjacent to i has the same number as one of $d(b_{2i-5}^2)$, $d(b_{2i-4}^2)$ and $d(b_i^3)$;

      iv. if i has degree δ, then $3 - \delta$ of $d(b_{2i-5}^2)$, $d(b_{2i-4}^2)$ and $d(b_i^3)$ are the node $|V'| + 3$;

   (c) $(|V'| + 3, e)$, for all $e \in E'$, with initial content $[b_e^4]$, where $d(b_e^4) = |E'| + |V'| + 4$;

   (d) $(e, |E'| + |V'| + 4)$, for all $e \in E'$, with initial content $[b_e^5]$, where $d(b_e^5) = |E'| + |V'| + 5$;

   (e) $a_6 = (|E'| + |V'| + 4, |E'| + |V'| + 5)$ with initial content $[b_1^6, b_2^6, \dots, b_{|E'|}^6]$, where $d(b_1^6) = d(b_2^6) = \dots = d(b_{|E'|-1}^6) = |E'| + |V'| + 5$ and $d(b_{|E'|}^6) = |E'| + |V'| + 6$;

   (f) $a_2 = (|E'| + |V'| + 5, |E'| + |V'| + 6)$, with initial content $[b^7]$, where $d(b^7) = |E'| + |V'| + 6$;

3. for each arc $a \in A$, create $v(a)$ further batches $b_1^a, b_2^a, \dots, b_{v(a)}^a$ initially stored at the start node of a and destinated to its end node;

4. create $|V'|$ batches $b_3^0, b_4^0, \dots, b_{|V'|+2}^0$ in $L - F$, initially stored at node 1, such that $d(b_i^0) = i$.

5. create the batch b_1 in $L - F$, initially stored at node 1, such that $d(b_1) = 2$.

6. set $b_2 = b_1^6$;

Recall that no batch initially contained in an arc is a further batch.

Figure 2.(a) shows an example of a graph G'. For $k' = 2$, figure 2.(a) shows the corresponding instance I' of DPP constructed as previously described. This figure represents nodes, arcs and arc contents as in figure 1.(a). The number of each node is inside the corresponding ellipse. Each batch b contained in an arc is labeled by $d(b)$. The initial

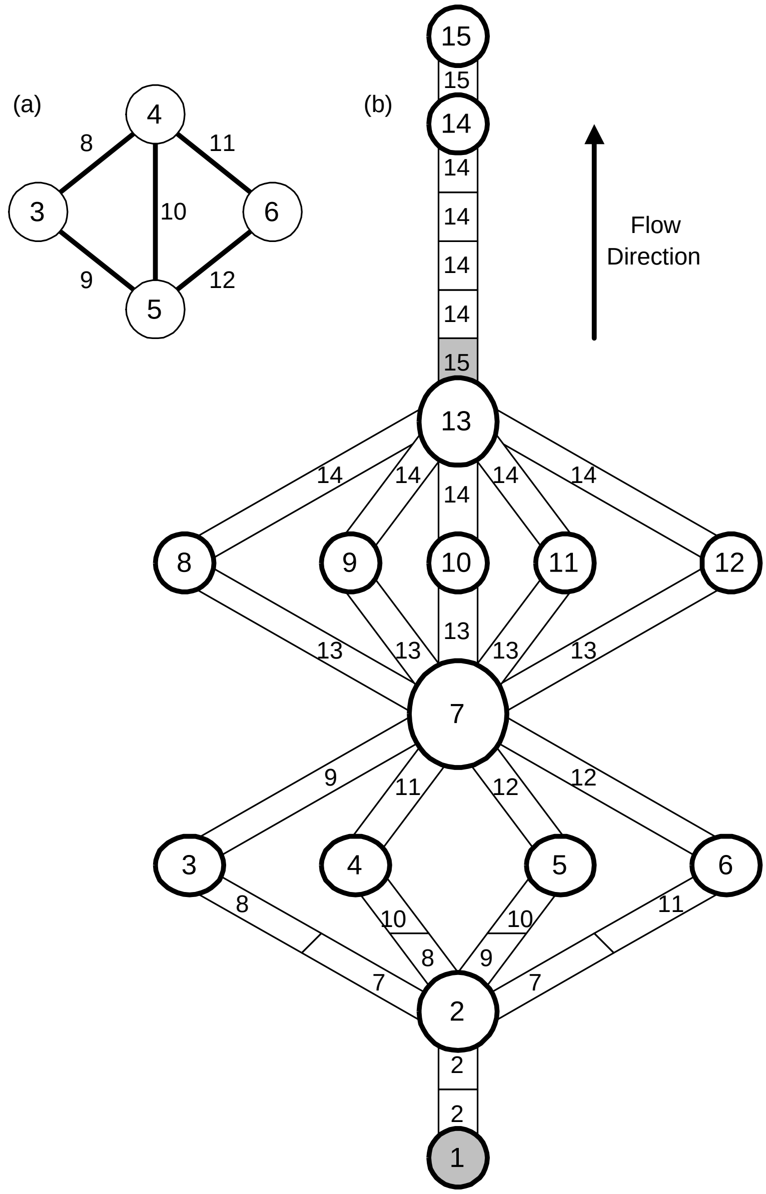

Figure 2: (a) an example of a graph G'; (b) the corresponding instance I' of DPP, for $k' = 2$.

positions of b_1 and b_2 are represented in gray. Finally, the flow directions of all pipelines are indicated by a single arrow. Clearly, the created graph G is both acyclic and planar.

Now, let $S' = \{i_1, i_2, \ldots, i_{k'}\}$ be vertex cover to G' and $V' - S' = \{i_{k'+1}, i_{k'+2}, \ldots, i_{|V'|}\}$. If there is a vertex cover with less than k' vertices to G', than we simply insert other vertices in it to obtain S'. In this case, we construct a corresponding certificate Q to I' as follows:

1. for $t = 1, 2, \ldots, |V'|$, create the EPO $(b^0_{i_t}, (1, 2), t)$;

2. for $t = 1, 2, \ldots, k'$, create the EPO's $(b^0_{i_t}, (2, i_t), t + |V'| - k')$, $(b_1^{(2,i_t)}, (2, i_t), t + |V'| - k' + 1)$, and $(b_2^{(2,i_t)}, (2, i_t), t + |V'| - k' + 2)$;

3. for $t = 1, 2, \ldots, k'$, create the EPO's $(b^2_{2i_t-4}, (i_t, |V'|+3), t+|V'|-k'+1)$, $(b^2_{2i_t-5}, (i_t, |V'|+3), t + |V'| - k' + 2)$, and $(b_1^{(i_t,|V'|+3)}, (i_t, |V'| + 3), t + |V'| - k' + 3)$;

4. for $t = 1, 2, \ldots, k'$, create the EPO's $(b^3_{i_t}, (|V'|+3, d(b^3_{i_t})), t+|V'|-k'+2)$, $(b^2_{2i_t-4}, (|V'|+3, d(b^2_{2i_t-4})), t + |V'| + 2)$, and $(b^2_{2i_t-5}, (|V'| + 3, d(b^2_{2i_t-5})), t + |V'| + k' + 2)$;

5. for all $e \in E'$, create the EPO's $(b^4_e, (e, |E'|+|V'|+4), |V'|+2k'+3)$, and $(b_1^{(e,|E'|+|V'|+4)}, (e, |E'| + |V'| + 4), |V'| + 2k' + 4)$;

6. for all $e \in E'$, create the EPO's $(b^5_e, a_6, e + 2k')$, and $(b^{a_6}_{e-|V'|-3}, a_6, e + |E'| + 2k')$;

7. create the EPO's $(b_1, a_1, |E'| + |V'| + 2k' + 4)$, $(b_2, a_2, |E'| + |V'| + 2k' + 4)$, and $(b_1^{a_2}, a_2, |E'| + |V'| + 2k' + 5)$;

8. for $t = 1, 2, \ldots, |V'| - k'$, create the EPO's $(b_t^{a_1}, a_1, t + |E'| + |V'| + 2k' + 4)$;

9. for $t = k' + 1, k' + 2, \ldots, |V'|$, create the EPO's $(b^0_{i_t}, (2, i_t), t + |E'| + |V'| + k' + 5)$, $(b_1^{(2,i_t)}, (2, i_t), t + |E'| + |V'| + k' + 6)$, and $(b_2^{(2,i_t)}, (2, i_t), t + |E'| + |V'| + k' + 7)$;

10. for $t = k'+1, k'+2, \ldots, |V'|$, create the EPO's $(b^2_{2i_t-4}, (i_t, |V'|+3), t+|E'|+|V'|+k'+6)$, $(b^2_{2i_t-5}, (i_t, |V'| + 3), t + |E'| + |V'| + k' + 7)$, and $(b_1^{(i_t,|V'|+3)}, (i_t, |V'| + 3), t + |E'| + |V'| + k' + 8)$;

11. for $t = k'+1, k'+2, \ldots, |V'|$, create the EPO's $(b^3_{i_t}, (|V'|+3, d(b^3_{i_t})), t+|E'|+|V'|+k'+7)$, $(b^2_{2i_t-4}, (|V'| + 3, d(b^2_{2i_t-4})), t + |E'| + 2|V'| + 7)$, and $(b^2_{2i_t-5}, (|V'| + 3, d(b^2_{2i_t-5})), t + |E'| + 3|V'| - k' + 7)$;

12. for all $e \in E'$, create the EPO $(b_1^{(|V'|+3,e)}, (|V'| + 3, e), |E'| + 4|V'| - k' + 8)$;

It is easy to verify that Q is a certificate to I'.

Now, let us consider the feasibility condition of Theorem 1. In every assignment from F to A' with the two properties of this theorem, the batch b^a_i is assigned to some position of arc a, for all $a \in A$ and $i = 1, 2, \ldots, v(a)$. To see this, one must assign batches to the outgoing arcs of each node in G following a topological node order [CLR92]. Hence, we obtain that $p_q(b^a_i) = a$ in any certificate to I', for all $a \in A$ and $i = 1, 2, \ldots, v(a)$. As a result, since G is acyclic, $b^a_1, b^a_2, \ldots, b^a_{v(a)}$ are necessarily the last $v(a)$ batches pumped into a.

Next, we show that, if G' has no vertex cover with k' vertices, then there is no certificate to I'. Let us assume that every vertex cover in G' has more than k' vertices. In this case, it is enough to prove that, for any feasible solution to I, b_2 reaches the start node of a_2 only after b_1 is pumped into a_1. First, observe that b_2 reaches the start node $|E'|+|V'|+5$ of a_2 only after pumping $|E'|$ batches into a_6. Moreover, as a consequence of Theorem 1, $b^5_{|V'|+4}, b^5_{|V'|+5}, \ldots, b^5_{|E'|+|V'|+3}$ are necessarily pumped into a_6 before $b^{a_6}_1, b^{a_6}_2, \ldots, b^{a_6}_{|E'|}$. In addition, a_6 is reachable to no other batch. However, b^5_e reaches the start node of a_6 only after pumping one batch into $(e, |E'|+|V'|+4)$, for each $e \in E'$. As a result, b_2 reaches the start node of a_2 only after pumping one batch into $(e, |E'|+|V'|+4)$, for all $e \in E'$. Again, as a consequence of Theorem 1, b^4_e must be pumped into $(e, |E'|+|V'|+4)$ before $b^{(e,|E'|+|V'|+4)}_1$, for each $e \in E'$. Moreover, $(e, |E'|+|V'|+4)$ is reachable to no other batch. Since b^4_e is initially contained at the arc $(|V'|+3, e)$, we obtain that b_2 reaches the start node of a_2 only after pumping one batch into $(|V'|+3, e)$, for all $e \in E'$. As a consequence of Theorem 1, every batch $b \in L - F$ with $d(b) = e$ must be pumped into $(|V'|+3, e)$ before $b^{(|V'|+3,e)}_1$, for each $e \in E'$. By construction, there are exactly two such batches. For each $e = (i, j) \in E'$, $(|V'|+3, e)$ is reachable to two batches that must pass by $(i, |V'|+3)$ and $(j, |V'|+3)$, respectively. Moreover, $(|V'|+3, e)$ is reachable to no other batch. Since every vertex cover in G' has more than k' vertices, the contents of at least $k'+1$ of $(3, |V'|+3), (4, |V'|+3), \ldots, (|V'|+2, |V'|+3)$ must move before b_2 reaches the start node. For each $i \in V'$, as a consequence of Theorem 1, $b^{(i,|V'|+3)}_1$ is pumped into $(i, |V'|+3)$ only after pumping both b^2_{2i-5} and b^2_{2i-4} into $(i, |V'|+3)$. Since no other batch is initially stored at node i of G, for each $i \in V'$, we obtain that b_2 reaches the start node of a_2 only after moving the contents of at least $k'+1$ of $(2,3), (2,4), \ldots, (2, |V'|+2)$. Furthermore, as a consequence of Theorem 1, $b^{(2,i)}_1$ is pumped into $(2, i)$ only after pumping b^0_i into $(2, i)$, for each $i \in V'$. Since $(2, i)$ is reachable to no other batch, for each $i \in V'$, we obtain that at least $k'+1$ of $b^0_3, b^0_4, \ldots, b^0_{|V'|+2}$ must cross the arc a_1 before b_2 reaches the start node of a_2. Hence, at least $v(a_1)+k'+1 = |V'|+1$ batches must be pumped into a_1 before that. Finally, as a consequence of Theorem 1, $b^{a_1}_1, b^{a_1}_2, \ldots, b^{a_1}_{|V'|-k'}$ are pumped into a_1 only after pumping b_1 into a_1. As a result, we must pump $b^0_3, b^0_4, \ldots, b^0_{|V'|+2}$, and b_1 into a_1 before b_2 reaches the start node of a_2, and we are done. ■

3.2 Approximability Bound

In this section, we prove our lower bound on the approximability of SPTMP. For that, we use the following approach. For any instance J of SPTMP, let us use $|J|$ to denote the number of bits required to represent J. Given an instance of 3-VCP represented by both G' and k', and a corresponding instance I of SPTMP constructed as in the proof of theorem 2, we construct an instance I^α of SPTMP by *cascading* α copies of I. Later, we explain this construction. After that, we prove that, if G' has a vertex cover with no more than k' vertices, then I^α has a feasible solution with a makespan equal to $t(|I|) = O(|I|)$. Otherwise, I^α has no feasible solution with makespan smaller than α. We also show that $|I^\alpha|$ is $O(\alpha|I|)$. Now, let us consider an $|J|^{1-\epsilon}$-approximation algorithm A that runs in $O(|J|^k)$ time, for any instance J of SPTMP and a given constant k. For $\alpha = |I|^{(3/\epsilon)-1}$, we have that A finds an $O(|I|^{(3/\epsilon)-3})$-approximate solution to I^α in $O(|I|^{3k/\epsilon})$ time. Since $\alpha/t(|I|) = \Omega(|I|^{(3/\epsilon)-2})$, if $|I|$ is sufficiently large, then A can be used to decide whether G' has a vertex cover with no more than k' vertices.

Hence, we have the following theorem.

Theorem 3 *For any fixed $\epsilon > 0$, there is no $\eta^{1-\epsilon}$-approximate algorithm for SPTMP unless $\mathcal{P} = \mathcal{NP}$, where η is the input size. This result also holds if the graph G is both planar and acyclic.*

Proof: By the previous discussion, it is enough to construct an instance I^α with the following three properties:

1. if G' has a vertex cover with no more than k' vertices, then I^α has a feasible solution with an $O(|I|)$ makespan;

2. if every vertex cover to G' has more than k' vertices, then I^α has no feasible solution with makespan smaller than α;

3. $|I^\alpha|$ is $O(\alpha|I|)$.

Now, let $I^{(1)}, I^{(2)}, \ldots, I^{(\alpha)}$ be α copies of I. I^α is constructed as follows:

1. for $j = 1, 2, \ldots, \alpha - 1$, remove from $I^{(j)}$ the following:

   (a) the node $|E'| + |V'| + 6$;

   (b) the arc a_2;

   (c) the two batches b^7 and $b_1^{a_2}$;

2. for $j = 1, 2, \ldots, \alpha-1$, connect the two pipeline networks of $I^{(j)}$ and $I^{(j+1)}$ by replacing both the node $|E'| + |V'| + 5$ of $I^{(j)}$ and the node 1 of $I^{(j+1)}$ by a single node;

3. for $j = 1, 2, \ldots, \alpha - 1$, replace both the batch b_2 of $I^{(j)}$ and the batch b_1 of $I^{(j+1)}$ by a single batch $b^{(j+1)}$ initially contained at the first position of the arc a_6 of $I^{(j)}$, and destinated to the node 2 of $I^{(j+1)}$.

Clearly, I^α satisfies Property 3. Let $a^{(j)}$ be the arc $(1, 2)$ of $I^{(j)}$, for $j = 1, 2, \ldots, \alpha$. Let also $b^{(1)}$ be the batch b_1 of $I^{(1)}$. Figure 3 represents the connections between the instances $I^{(j-1)}$, $I^{(j)}$, and $I^{(j+1)}$, in I^α. In this figure, circles represent nodes, rectangles represent pipelines, and each cloud represents the remaining of the pipeline network corresponding to each copy of I. In addition, the three batches $b^{(j)}$, $b^{(j+1)}$, and $b^{(j+2)}$ are gray colored.

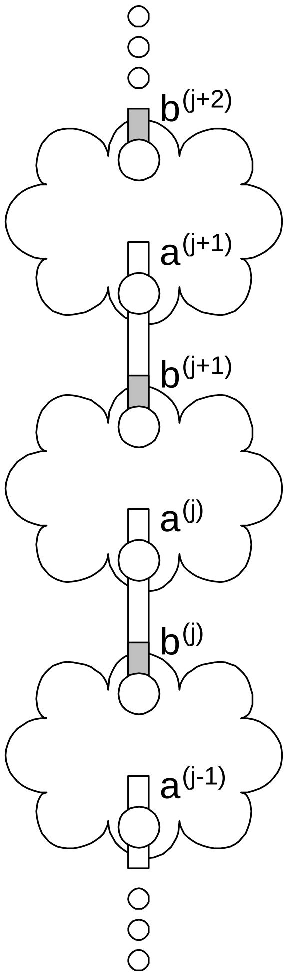

Figure 3: Connections between the instances $I^{(j-1)}$, $I^{(j)}$, and $I^{(j+1)}$, in I^α.

Observe that $b^{(j)}$ represents both the batch b_2 of $I^{(j-1)}$ and the batch b_1 of $I^{(j)}$, for $j = 2, 3, \ldots, \alpha$. Moreover, $a^{(j)}$ represents both the arc a_2 of $I^{(j-1)}$ and the arc a_1 of $I^{(j)}$. Hence, we claim that, if every vertex cover to G' has more than k' vertices, then any feasible solution to I^α pumps $b^{(j)}$ into $a^{(j)}$ before pumping $b^{(j+1)}$ into $a^{(j+1)}$, for $j = 1, 2, \ldots, \alpha - 1$. Observe that Property 2 of I^α immediately follows from this claim. A proof for this claim is analogous to that of theorem 2.

Now, let $Q^{(j)}$ be a feasible solution to $I^{(j)}$ constructed as in the proof of theorem 2. If G' has a vertex cover with no more than k' vertices, then a feasible solution Q to I^α with an $O(|I|)$ makespan is constructed as follows:

1. for $j = 1, 2, \ldots, \alpha$, remove from $Q^{(j)}$ the EPO $(b_1^{a_2}, a_2, |E'| + |V'| + 2k' + 5)$;

2. for $j = 1, 2, \ldots, \alpha - 1$, replace both the EPO $(b_2, a_2, |E'| + |V'| + 2k' + 4)$ of $Q^{(j)}$ and the EPO $(b_1, a_1, |E'| + |V'| + 2k' + 4)$ of $Q^{(j+1)}$ by a single EPO $(b^{(j)}, a^{(j)}, |E'| + |V'| + 2k' + 4)$;

3. $Q = Q^{(1)} \cup Q^{(2)} \cup \cdots \cup Q^{(\alpha)}$.

It follows from this construction that I^α satisfies Property 1, what completes our proof. ■

4 Final Remarks

In this paper, we prove that, for any fixed $\epsilon > 0$, there is no $\eta^{1-\epsilon}$-approximate algorithm for SPTMP unless $\mathcal{P} = \mathcal{NP}$, where η is the input size. In [Kan94], V. Kann investigates the class of *polynomially bounded minimization problems* (Min PB). The author shows that Min PB-complete problems cannot be approximated within η^ϵ, for some $\epsilon > 0$. Moreover, some of these problems are proved to have the same approximability bound as SPTMP. Hence, whether SPTMP is Min PB-complete is an interesting open question.

References

[AMO93] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.

[Cam95] 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.

[CLR92] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*. MIT Press and McGraw-Hill Book Company, 6th edition, 1992.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[HR95] Christopher A. Hane and H. Donald Ratliff. Sequencing inputs to multi-commodity pipelines. *Annals of Operations Research*, 57, 1995. Mathematics of Industrial Systems I.

[Kan94] Viggo Kann. Polynomially bounded minimization problems that are hard to approximate. *Nordic Journal of Computing*, 1(3):317–331, 1994.

[MLPR99] Ruy L. Milidiú, Eduardo S. Laber, Artur A. Pessoa, and Pablo A. Rey. Petroleum products scheduling in pipelines. In *The International Workshop on Harbour, Maritime & Industrial Logistics Modeling and Simulation*, september 1999.

[MPL00] Ruy L. Milidiú, Artur A. Pessoa, and Eduardo S. Laber. Transporting petroleum products in pipelines (abstract). In *ISMP 2000 – 17th International Symposium on Mathematical Programming*, pages 134–135, Atlanta, Georgia, USA, August 2000.

[MPL02] Ruy L. Milidiú, Artur A. Pessoa, and Eduardo S. Laber. Pipeline transportation of petroleum products. In *Proceedings of the LATIN'2002 (to appear)*, Canún, Mexico, april 2002.