



The complexity of makespan minimization for pipeline transportation[☆]

Ruy Luiz Milidiú, Artur Alves Pessoa^{*,1}, Eduardo Sany Laber²

*Informatics Department of PUC-Rio, Brazil, Rua Marquês de São Vicente 225, EPLF, 13º andar,
CEP 22453-900, Rio de Janeiro - RJ, Brazil*

Received 31 October 2002; received in revised form 7 April 2003; accepted 11 April 2003

Communicated by O. Ibarra

Abstract

SPTP is a model for the pipeline transportation of petroleum products. It uses a directed graph G , where arcs represent pipes and nodes represent locations. In this paper, we analyze the complexity of finding a minimum makespan solution to SPTP. This problem is called SPTMP. We prove that, for any fixed $\varepsilon > 0$, there is no $\eta^{1-\varepsilon}$ -approximate algorithm for the SPTMP unless $\mathcal{P} = \mathcal{NP}$, where η is the input size. This result also holds if G is both planar and acyclic. If G is acyclic, then we give a m -approximate algorithm to SPTMP, where m is the number of arcs in G .

© 2003 Elsevier B.V. All rights reserved.

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

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

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

* Corresponding author. Tel.: +55-21-2540-6915; fax: +55-21-2511-5645.

E-mail addresses: milidiu@inf.puc-rio.br (R.L. Milidiú), artur@inf.puc-rio.br (A.A. Pessoa), laber@inf.puc-rio.br (E.S. Laber).

¹ This work was partially supported by FAPERJ, Bolsa de Doutorado Nota 10.

² This work was partially supported by CNPq, Bolsa de Produtividade, process 300428/99-5.

pipeline and remove the same amount of product from the opposite side. Typically, each oil 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 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 this problem [1,3,5,7]. Let us define an *order* as a requirement to transport a given amount of some product from one location to another. In [3], Hane and Ratliff present a model that assumes cyclic orders. In this case, the same orders always repeat after the completion of a given time period. In [6,7], the Pipeline Transportation Problem (PTP) model is proposed for the pipeline transportation of petroleum products with non-cyclic orders. 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. The initial position of a batch may be either a node or a pipeline. Moreover, PTP assumes that all batches have unitary volumes and that no batch can be split during its transportation. 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 the set of r batches. Since pipelines must always be full, some batches must be used to fill the pipelines at the end of the schedule. Observe that these batches are not delivered. Due to this fact, PTP defines a subset $F \subset L$ of *further* batches that are not necessarily delivered at the end of a feasible pumping sequence. As a result, a feasible solution is a pumping sequence that delivers all *non-further* batches in $L - F$.

In [7], 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 this work, the authors also introduce the BPA algorithm, that finds feasible solutions to SPTP in 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 Synchronous Pipeline Transportation Makespan Problem (SPTMP). We prove that, for any fixed $\varepsilon > 0$, there is no $\eta^{1-\varepsilon}$ -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. To prove this result, we propose the Precedence Pipeline Problem (PPP). In this problem, we are given a special instance of the SPTMP where two pipeline operations π_1 and π_2 must be executed. Then, we must find a feasible solution where π_1 is executed not before π_2 . We also show that PPP is \mathcal{NP} -complete. A preliminary version of this paper was presented in [8].

Next, we give an overview of our proof for the inapproximability of SPTMP. Let I be an instance of PPP. First, we prove that PPP is \mathcal{NP} -complete. Then, we propose a construction of another an instance I^α of SPTMP by *chaining* α copies of I . This construction is such that the operation π_2 for the i th copy of I is the same as the operation π_1 for the $(i + 1)$ th copy, for $i = 1, 2, \dots, \alpha - 1$. Hence, any solution for I^α with makespan smaller than α , does not execute π_1 before π_2 in at least one copy of I . This provides a certificate for I . On the other hand, the construction of I^α assures that it has a feasible solution with makespan $O(|I|)$ whenever I has a certificate, where $|I|$ is the number of bits required to represent I . Our approximation lower bound is obtained by assigning an appropriate value to α as a function of $|I|$.

For completeness, we also show that the BPA algorithm can be modified to find a m -approximate solution to SPTMP, for acyclic graphs. Although this approximation factor is very high, the lower bound proved in this paper prevents one to do much better unless $\mathcal{P} = \mathcal{NP}$.

This paper is organized as follows. In Section 2, we describe the SPTMP. In Section 3, we prove our approximability bounds. In Section 4, 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.

2.1. 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 divide each arc a into $v(a)$ pipeline positions. We also define the set of all pipeline positions $A' = \{(a, l) \mid a \in A \text{ and } l \in \{1, \dots, v(a)\}\}$.

2.2. Orders

Let L be a set of r unitary volume batches. Each $b \in L$ corresponds to a transportation order which is a commitment to deliver b at $d(b) \in N$. We define a subset $F \subset L$ that is called the subset of *further* batches and, similarly, $L - F$ is called the subset of *non-further* batches.

2.3. Pipeline contents

Pumping a batch into a pipeline requires a non-negligible amount of time. However, we only consider the instants where each arc $a \in A$ contains exactly $v(a)$ integral

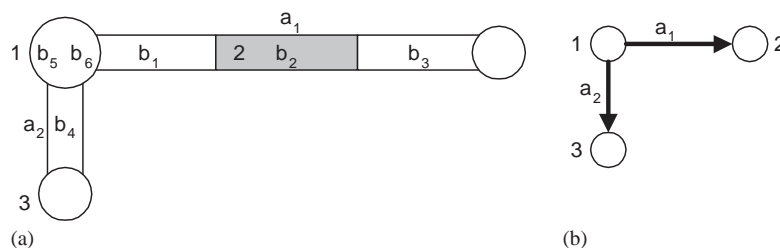


Fig. 1. (a) The contents of a pipeline system; (b) the corresponding graph.

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 l th 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 $[b_1, b_2, \dots, b_{v(a)}]$. In this case, b_l is a batch such that $p_t(b_l) = (a, l)$, for $l = 1, 2, \dots, v(a)$.

As an example, Fig. 1(a) represents the pipeline contents corresponding to the graph of Fig. 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 Fig. 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 $p_0(b) \in N$.

2.4. 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 $[b_1, b_2, \dots, b_{v(a)}]$. 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 $[b, b_1, b_2, \dots, b_{v(a)-1}]$ and $b_{v(a)}$ is stored at the node j . We point out that some EPO's may be simultaneously executed. Formally, given two different EPOs (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 $q = \max\{t + 1 \mid (b, a, t) \in Q\}$. 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 $d(b)$.

2.5. Objective function

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

3. Complexity of SPTMP

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

First, 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 $p_t(b) = i \in N$.

Moreover, we say that an arc a is *allowed* 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 allowed arcs.

3.1. Tight instances

Now, let us define a special subset of instances of the SPTMP that we use to prove our complexity results. We refer to instances in this subset as *tight instances*.

Definition 1. An instance I of the SPTMP is tight when the following four conditions hold:

- (1) G is acyclic;
- (2) for every batch $b \in L$, there is exactly one path in G connecting the tail node of $p_0(b)$ to $d(b)$;
- (3) for every arc $a = (i, j) \in A$, there are exactly $v(a)$ further batches initially stored at node i and destined to node j ;
- (4) there is no other further batch.

For $l = 1, 2, \dots, v(a)$, let us use the notation $b_l^{(i,j)}$ to denote the l th further batch initially stored at node i and destined to node j . Let us refer to these batches as the *fillers* of the arc (i, j) .

For example, Fig. 2 represents a tight instance of the SPTMP. This figure represents nodes, arcs and arc contents as in Fig. 1(a). The number of each node is inside the

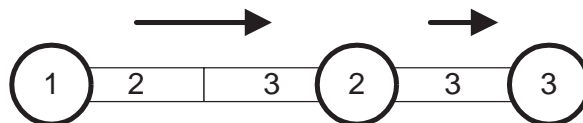


Fig. 2. An example of a tight instance of the SPTMP.

corresponding circle. Each batch b contained in an arc is labeled inside by $d(b)$. The two arrows indicate the flow directions into the arcs. By the definition of tight instances, observe that $F = \{b_1^{(1,2)}, b_2^{(1,2)}, b_1^{(2,3)}\}$, where $p_0(b_1^{(1,2)}) = p_0(b_2^{(1,2)}) = 1$, $p_0(b_1^{(2,3)}) = 2$, $d(b_1^{(1,2)}) = d(b_2^{(1,2)}) = 2$, and $d(b_1^{(2,3)}) = 3$. Hence, this information is not explicitly represented in Fig. 2. Throughout this paper, we shall represent tight instances as in this figure, that is, with no indication of the corresponding arc fillers.

Now, we have the following lemma.

Lemma 1. *For any feasible solution to a tight instance of the SPTMP, each arc $a \in A$ must contain all its fillers at the final state.*

Proof. First, observe that the filler of an arc $a = (i, j)$ is initially stored at node i and destined to node j . Since G is acyclic, we have that this filler is not allowed to any arc other than a . Since all further batches are fillers in a tight instance, we obtain that only the fillers of a can be contained in this arc at the final state. Moreover, we have exactly $v(a)$ fillers for each arc a . As a result, every filler must be pumped into the corresponding arc, in order to fill it at the final state. \square

In the instance of Fig. 2, observe that the batch initially contained at the second pipeline position of the arc $(1, 2)$ must be pumped into the arc $(2, 3)$ before the further batch $b_1^{(2,3)}$. The next theorem generalizes this observation to every feasible solution to a tight instance of the SPTMP.

Theorem 1. *Let Q be a feasible solution to a tight instance I of the SPTMP. In this case, for every batch $b \in L - F$ and every arc a in the path that connects the tail node of $p_0(b)$ to $d(b)$, b must be pumped into a before any filler of a .*

Proof. First, we show that b must be pumped into a . This is true because b must reach $d(b)$ and a belongs to the only path that connects the tail node of $p_0(b)$ to $d(b)$. Moreover, the last $v(a)$ batches pumped into a are exactly the batches contained in this arc at the final state. By Lemma 1, these batches must be the $v(a)$ fillers of a . Since G is acyclic, no batch can be pumped twice into the same arc. Hence, b must be pumped into a before any filler of a . \square

3.2. Precedence Pipeline Problem

Here, we prove that, for a given instance I of the SPTMP and two given EPOs π_1 and π_2 , finding a feasible solution to I where π_1 is not executed before π_2 is a \mathcal{NP} -complete problem.

Formally, given an instance I of the SPTMP, two batches $\bar{b}_1, \bar{b}_2 \in L$, and two arcs $\bar{a}_1, \bar{a}_2 \in A$, the *Precedence Pipeline Problem* (PPP) is to find a feasible solution Q to I containing both the EPO's $\pi_1 = (\bar{a}_1, \bar{b}_1, t_1)$ and $\pi_2 = (\bar{a}_2, \bar{b}_2, t_2)$, for some $t_1 \geq t_2$.

In the next theorem, we prove that PPP is a \mathcal{NP} -complete problem by showing a polynomial reduction from the Vertex Cover Problem (VCP) to PPP. In this proof,

instances of the PPP that correspond to tight instances of the SPTMP are also referred to as tight instances.

Given an undirected graph $\mathcal{G} = (V, E)$ and a positive integer $k < |V|$, the 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 \mathcal{G} is at most 3. We point out that 3-VCP is also \mathcal{NP} -complete [2].

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

Proof. This proof is divided into four parts. In the first part, we prove that PPP belongs to \mathcal{NP} . In the second part, we present a polynomial reduction from 3-VCP to PPP. In the third part, we show that a certificate to PPP leads to a certificate to 3-VCP. Finally, in the last part, we show that a certificate to 3-VCP leads to a certificate to PPP.

Part I: PPP belongs to \mathcal{NP} : Let I be an instance of the SPTMP, with a corresponding graph G . Since G is acyclic,³ for any feasible solution Q to I , each batch can be pumped into at most m arcs. Hence, Q has no more than rm EPOs. Let I' be an instance of PPP given by I , $\bar{b}_1, \bar{b}_2 \in L$, and $\bar{a}_1, \bar{a}_2 \in A$. Since any certificate to I' is also a feasible solution to I , PPP belongs to \mathcal{NP} .

Part II: a reduction from 3-VCP to PPP: Next, we show a polynomial reduction from an instance of 3-VCP represented by both \mathcal{G} and k to a tight instance I' of PPP. For the sake of simplicity, the notation used to denote each vertex or edge in \mathcal{G} is also used to denote the corresponding node in G .

Fig. 3(a) shows an example of a graph \mathcal{G} . For $k = 2$, Fig. 3(b) shows the corresponding instance I' of PPP. Later, we explain the construction of I' . The initial positions of \bar{b}_1 and \bar{b}_2 are represented in gray. The arcs $\bar{a}_1 = (1, 2)$, $\bar{a}_2 = (5, 6)$ and $a_6 = (4, 5)$ are also indicated. The flow directions of all pipelines are defined by a single arrow. Finally, this figure shows the notation used for each group of non-further batches on the right side of the pipeline system. Clearly, the graph G that corresponds to the pipeline system of Fig. 3(b) is both acyclic and planar.

Now, let us consider a general instance of 3-VCP represented by both $\mathcal{G} = (V, E)$ and k , where $V = \{s_1, s_2, \dots, s_{|V|}\}$ and $E = \{e_1, e_2, \dots, e_{|E|}\}$. We construct a corresponding tight instance I' of PPP as follows:

- (1) let $N = \{1, 2, \dots, 6\} \cup \{s_1, s_2, \dots, s_{|V|}\} \cup \{e_1, e_2, \dots, e_{|E|}\}$;
- (2) create the following arcs in A :
 - (a) $\bar{a}_1 = (1, 2)$ 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) $(2, s_i)$ and $(s_i, 3)$, for $i = 1, 2, \dots, |V|$, where:
 - (i) $(2, s_i)$ has initial content $[b_{2i-1}^2, b_{2i}^2]$;
 - (ii) $(s_i, 3)$ has initial content $[b_i^3]$;
 - (iii) $d(b_{2i-1}^2) = e_j$, $d(b_{2i}^2) = e_k$ and $d(b_i^3) = e_l$, where $e_j, e_k, e_l \in E$ are the three edges adjacent to s_i in \mathcal{G} ;

³ If G has one or more cycles, then whether PPP belongs to \mathcal{NP} or not is an open question.

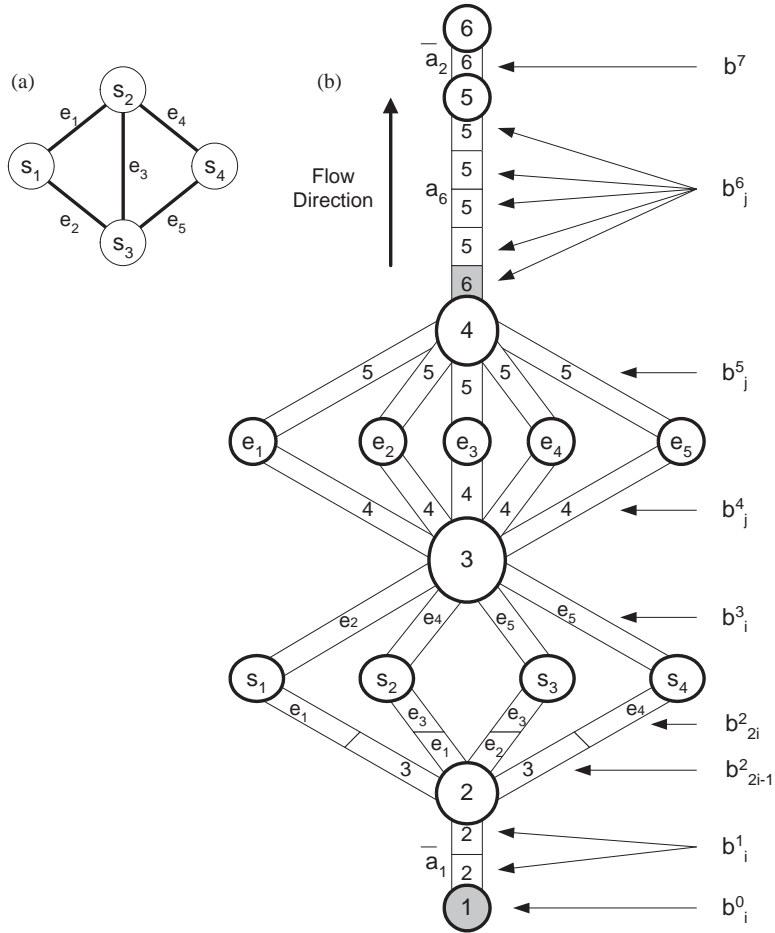


Fig. 3. (a) An example of a graph \mathcal{G} ; (b) the corresponding instance I' of PPP, for $k=2$.

- (iv) if i has degree $\delta < 3$, then the remaining $3 - \delta$ batches are destined to the node 3;
- (c) $(3, e_j)$, for $j = 1, 2, \dots, |E|$, with initial content $[b^4_j]$, where $d(b^4_j) = 4$;
- (d) $(e_j, 4)$, for $j = 1, 2, \dots, |E|$, with initial content $[b^5_j]$, where $d(b^5_j) = 5$;
- (e) $a_6 = (4, 5)$ with initial content $[\bar{b}_2, b^6_2, b^6_3, \dots, b^6_{|E|}]$, where $d(b^6_2) = d(b^6_3) = \dots = d(b^6_{|E|}) = 5$ and $d(\bar{b}_2) = 6$;
- (f) $\bar{a}_2 = (5, 6)$, with initial content $[b^7]$, where $d(b^7) = 6$;
- (3) for each arc $a \in A$, create the corresponding fillers;
- (4) for $i = 1, 2, \dots, |V|$, create the non-further batch b^0_i initially stored at node 1, with $d(b^0_i) = s_i$.
- (5) create the non-further batch \bar{b}_1 , initially stored at node 1, with $d(\bar{b}_1) = 2$.

Observe that the previous instance is tight. Next, we use Theorem 1 to show that any certificate to I' gives a vertex cover to \mathcal{G} with no more than k vertices.

Part III: a certificate to PPP leads to a certificate to 3-VCP: Let Q be a certificate to I' . We consider, from the batches $b_1^0, b_2^0, \dots, b_{|V|}^0$, which ones leave the arc \bar{a}_1 before \bar{b}_2 is pumped into \bar{a}_2 , according to Q . Let us refer to these batches as the *selected batches*. Observe that exactly $|V| - k$ batches must stay at \bar{a}_1 to keep it filled before \bar{b}_1 and the corresponding fillers are pumped into this arc. Moreover, by Theorem 1, the fillers of \bar{a}_1 cannot be pumped before \bar{b}_1 . Since \bar{b}_1 cannot be pumped before \bar{b}_2 is pumped into \bar{a}_2 , we obtain that at most k batches can leave \bar{a}_1 before \bar{b}_2 is pumped into \bar{a}_2 . As a result, we have no more than k selected batches.

Let us refer to the vertices of \mathcal{G} that correspond to the selected batches as the *selected vertices*. Next, we prove that the set of selected vertices is a vertex cover of \mathcal{G} .

First, observe that at least $|E|$ batches must be pumped into a_6 before \bar{b}_2 reaches the start node of \bar{a}_2 . Moreover, as a consequence of Theorem 1, $b_1^5, b_2^5, \dots, b_{|E|}^5$ are necessarily pumped into a_6 before the fillers of this arc. In addition, a_6 is not allowed to any other batch. Hence, at least one batch must be pumped into $(e_j, 4)$, for $j = 1, 2, \dots, |E|$, before \bar{b}_2 reaches the start node of \bar{a}_2 . By an analogous argument, we obtain that at least one batch must be pumped into $(3, e_j)$, for $j = 1, 2, \dots, |E|$, before \bar{b}_2 reaches the start node of \bar{a}_2 . However, as a consequence of Theorem 1, the filler of $(3, e_j)$ cannot be pumped into this arc before any non-further batch b with $d(b) = e_j$. By construction, for each $e = (s_i, s_j) \in E$, we have exactly two non-further batches destined to the node e . These two batches must pass respectively by the arc $(s_i, 3)$ and the arc $(s_j, 3)$ before reaching node 3. Since $(3, e)$ is not allowed to any other batch, we have that, for all $e = (i, j) \in E$, either the content of $(s_i, 3)$ or the content of $(s_j, 3)$ must move before \bar{b}_2 reaches the start node of \bar{a}_2 . Let $S \subset V$ be a set containing every index i such that the content of $(s_i, 3)$ moves before \bar{b}_2 reaches the start node of \bar{a}_2 , according to Q . By the previous discussion, we have that S is a vertex cover of \mathcal{G} . Moreover, by Theorem 1, the content of $(2, s_i)$ must move before the content of $(s_i, 3)$ moves, for $i = 1, 2, \dots, |V|$. By the same theorem, b_i^0 must reach node 2 before that. As a result, for all i such that $s_i \in S$, b_i^0 must leave the arc \bar{a}_1 , before \bar{b}_2 reaches the start node of \bar{a}_2 . Hence, every vertex in S is a selected vertex. As a consequence, the set of selected vertices is also a vertex cover of \mathcal{G} .

Part IV: a certificate to 3-VCP leads to a certificate to PPP: If there is a vertex cover with less than k vertices in \mathcal{G} , than a vertex cover with exactly k vertices can be obtained by arbitrarily inserting other vertices in it. Hence, let us assume without loss of generality that $S = \{s_1, s_2, \dots, s_k\}$ is a vertex cover of \mathcal{G} . In this case, we construct a corresponding certificate Q to I' as follows:

- (1) For $t = 1, 2, \dots, |V|$, create the EPO $(b_t^0, (1, 2), t - 1)$. Since $v((1, 2)) = |V| - k$, b_t^0 is stored at the node 2 at the state $t + |V| - k$, for $t = 1, 2, \dots, k$.
- (2) For $t = 1, 2, \dots, k$, create the EPOs $(b_t^0, (2, s_t), t + |V| - k)$, $(b_1^{(2, s_t)}, (2, s_t), t + |V| - k + 1)$, and $(b_2^{(2, s_t)}, (2, s_t), t + |V| - k + 2)$.
- (3) For $t = 1, 2, \dots, k$, create the EPOs $(b_{2t}^2, (s_t, 3), t + |V| - k + 1)$, $(b_{2t-1}^2, (s_t, 3), t + |V| - k + 2)$, and $(b_1^{(s_t, 3)}, (s_t, 3), t + |V| - k + 3)$.

- (4) For $t = 1, 2, \dots, k$, create the EPOs $(b_t^3, (3, d(b_t^3)), t + |V| - k + 2)$, $(b_{2t}^2, (3, d(b_{2t}^2)), t + |V| + 2)$, and $(b_{2t-1}^2, (3, d(b_{2t-1}^2)), t + |V| + k + 2)$. Since S is a vertex cover for \mathcal{G} , at least one batch is pumped into $(3, e_j)$, for $j = 1, 2, \dots, |E|$. Hence, b_j^4 is stored at node e_j , at the state $|V| + 2k + 3$.
- (5) For $j = 1, 2, \dots, |E|$, create the EPO's $(b_j^4, (e_j, 4), |V| + 2k + 3)$, and $(b_1^{(e_j, 4)}, (e_j, 4), |V| + 2k + 4)$.
- (6) For $j = 1, 2, \dots, |E|$, create the EPO's $(b_j^5, a_6, |V| + 2k + 3 + j)$, and $(b_j^{a_6}, a_6, |E| + |V| + 2k + 3 + j)$.
- (7) Create the EPOs $\pi_1 = (\bar{b}_1, \bar{a}_1, |E| + |V| + 2k + 4)$, $\pi_2 = (\bar{b}_2, \bar{a}_2, |E| + |V| + 2k + 4)$, and $(b_1^{\bar{a}_2}, \bar{a}_2, |E| + |V| + 2k + 5)$.
- (8) For $t = 1, 2, \dots, |V| - k$, create the EPOs $(b_t^{\bar{a}_1}, \bar{a}_1, t + |E| + |V| + 2k + 4)$. As a result, b_t^0 is stored at the node 2 at the state $t + |E| + |V| + k + 5$, for $t = k + 1, k + 2, \dots, |V|$.
- (9) For $t = k + 1, k + 2, \dots, |V|$, create the EPOs $(b_t^0, (2, s_t), t + |E| + |V| + k + 5)$, $(b_1^{(2, s_t)}, (2, s_t), t + |E| + |V| + k + 6)$, and $(b_2^{(2, s_t)}, (2, s_t), t + |E| + |V| + k + 7)$.
- (10) For $t = k + 1, k + 2, \dots, |V|$, create the EPO's $(b_{2t}^2, (s_t, 3), t + |E| + |V| + k + 6)$, $(b_{2t-1}^2, (s_t, 3), t + |E| + |V| + k + 7)$, and $(b_1^{(s_t, 3)}, (s_t, 3), t + |E| + |V| + k + 8)$.
- (11) For $t = k + 1, k + 2, \dots, |V|$, create the EPO's $(b_t^3, (3, d(b_t^3)), t + |E| + |V| + k + 7)$, $(b_{2t}^2, (3, d(b_{2t}^2)), t + |E| + 2|V| + 7)$, and $(b_{2t-1}^2, (3, d(b_{2t-1}^2)), t + |E| + 3|V| - k + 7)$.
- (12) For $j = 1, 2, \dots, |E|$, create the EPO $(b_1^{(3, e_j)}, (3, e_j), |E| + 4|V| - k + 8)$.

It can be verified that Q is a certificate to I' , and we are done. \square

3.3. Approximability lower 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 \mathcal{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 *chaining* α copies of I . Later, we explain this construction. After that, we prove that, if \mathcal{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 $|J|^{1-\varepsilon}$ -approximation algorithm \mathcal{A} that runs in $O(|J|^c)$ time, for any instance J of SPTMP and a given constant c . For $\alpha = |I|^{(3/\varepsilon)-1}$, we have that \mathcal{A} finds an $O(|I|^{(3/\varepsilon)-3})$ -approximate solution to I^α in $O(|I|^{3c/\varepsilon})$ time. Since $\alpha/t(I) = \Omega(|I|^{(3/\varepsilon)-2})$, if $|I|$ is sufficiently large, then \mathcal{A} can be used to decide whether \mathcal{G} has a vertex cover with no more than k vertices.

Hence, we have the following theorem.

Theorem 3. *For any fixed $\varepsilon > 0$, there is no $\eta^{1-\varepsilon}$ -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:

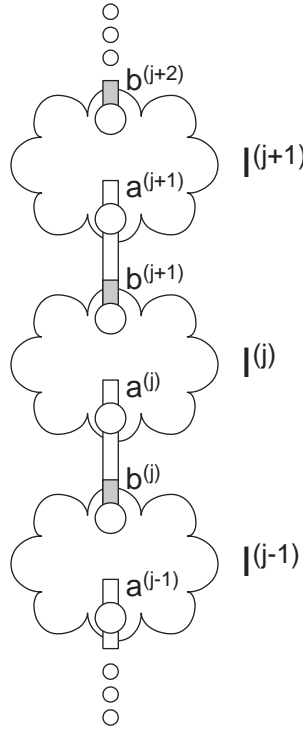


Fig. 4. Connections between the instances $I^{(j-1)}$, $I^{(j)}$, and $I^{(j+1)}$, in I^α .

- (1) if \mathcal{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 \mathcal{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 $I^{(1)}, I^{(2)}, \dots, I^{(\alpha)}$ be α copies of I . In order to construct I^α , for each $j = 1, 2, \dots, \alpha - 1$, do the following five steps:

- (1) remove from $I^{(j)}$ the following:
 - (a) the node 6;
 - (b) the arc \bar{a}_2 ;
 - (c) the two batches b^7 and $b_1^{\bar{a}_2}$;
- (2) connect the two pipeline networks of $I^{(j)}$ and $I^{(j+1)}$ by replacing both node 5 of $I^{(j)}$ and the node 1 of $I^{(j+1)}$ by a single node;
- (3) let the arc \bar{a}_1 of $I^{(j+1)}$ be also the new arc \bar{a}_2 for $I^{(j)}$;
- (4) remove the batch \bar{b}_1 of $I^{(j+1)}$;
- (5) let the batch \bar{b}_2 of $I^{(j)}$ be also the new batch \bar{b}_1 for $I^{(j+1)}$;
- (6) destinate this batch to the node 2 of $I^{(j+1)}$.

Clearly, I^α satisfies property 3. Let us use $b^{(j)}$ and $a^{(j)}$ to denote the batch \bar{b}_1 and the arc \bar{a}_1 for $I^{(j)}$, respectively, for $j = 1, 2, \dots, \alpha$. Let also $b^{(\alpha+1)}$ and $a^{(\alpha+1)}$ be respectively the batch \bar{b}_2 and the arc \bar{a}_2 for $I^{(\alpha)}$. Fig. 4 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.

Observe that $b^{(j)}$ represents both the batch \bar{b}_2 for $I^{(j-1)}$ and the batch \bar{b}_1 for $I^{(j)}$, for $j = 2, 3, \dots, \alpha$. Moreover, $a^{(j)}$ represents both the arc \bar{a}_2 for $I^{(j-1)}$ and the arc \bar{a}_1 for $I^{(j)}$. Furthermore, pumping $b^{(j)}$ into $a^{(j)}$ before pumping $b^{(j+1)}$ into $a^{(j+1)}$ is essentially the same as pumping \bar{b}_1 into \bar{a}_1 before pumping \bar{b}_2 into \bar{a}_2 , in I . Hence, by Theorem 2, if every vertex cover to \mathcal{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 $j = 1, 2, \dots, \alpha - 1$. Observe that Property 2 of I^α immediately follows from this claim.

Now, let $Q^{(j)}$ be a feasible solution to $I^{(j)}$ constructed as in the proof of Theorem 2. If \mathcal{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 $j = 1, 2, \dots, \alpha$, remove from $Q^{(j)}$ the EPO $(b_1^{\bar{a}_2}, \bar{a}_2, |E| + |V| + 2k + 5)$;
- (2) for $j = 1, 2, \dots, \alpha - 1$, replace both the EPO $(\bar{b}_2, \bar{a}_2, |E| + |V| + 2k + 4)$ of $Q^{(j)}$ and the EPO $(\bar{b}_1, \bar{a}_1, |E| + |V| + 2k + 4)$ of $Q^{(j+1)}$ by a single EPO $(b^{(j)}, a^{(j)}, |E| + |V| + 2k + 4)$;
- (3) $Q = Q^{(1)} \cup Q^{(2)} \cup \dots \cup Q^{(\alpha)}$.

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

3.4. Approximability upper bound

Here, we show that the BPA algorithm [7] can be modified to find a m -approximate solution to SPTMP, for acyclic graphs.

BPA assumes that every order has a corresponding batch weight. If G is acyclic, then BPA finds a minimum cost solution to SPTMP, where the cost of an EPO (b, a, t) is equal to the sum of the weights of all batches contained in a during the execution of this EPO. In this case, both the batch that enters a and the batch that leaves a have only one half of their weights added to this cost. Moreover, BPA generates EPOs that are sequentially executed. Hence, the obtained solution has a makespan equal to the number of generated EPOs.

Now, let us consider that cost of any EPO is exactly one. In this case, we point out that BPA can still be used to find a minimum cost solution to SPTMP, with minor modifications. Let \hat{q} be the minimum number of EPOs for an instance I of SPTMP. Since each arc can execute at most one EPO per time unit, we obtain that any feasible solution to I has a makespan not smaller than \hat{q}/m . Moreover, since BPA gives a solution to I with makespan \hat{q} , this solution is m -approximate.

4. Final remarks

In this paper, we prove that, for any fixed $\varepsilon > 0$, there is no $\eta^{1-\varepsilon}$ -approximate algorithm for SPTMP unless $\mathcal{P} = \mathcal{NP}$, where η is the input size. In [4], Kann investi-

gates 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 $\varepsilon > 0$. 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.

For completeness, we also give a m -approximate algorithm for the SPTMP, for acyclic graphs. An interesting open problem is to design an $O(\delta)$ -approximate algorithm for the SPTMP, where δ is the maximum number of arcs in a simple path of G . Observe that such algorithm does not conflict with the previous lower bound.

References

- [1] E. 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] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, 1979.
- [3] C.A. Hane, H.D. Ratliff, Sequencing inputs to multi-commodity pipelines, Ann. Oper. Res. 57, Mathematics of Industrial Systems I, 1995; paper available on the web "<http://gunther.smeal.psu.edu/13915.html>".
- [4] V. Kann, Polynomially bounded minimization problems that are hard to approximate, Nordic J. Comput. 1 (3) (1994) 317–331.
- [5] R.L. Milidiú, E.S. Laber, A.A. Pessoa, P.A. Rey, Petroleum products scheduling in pipelines, in: The Internat. Workshop on Harbour, Maritime & Industrial Logistics Modeling and Simulation, 1999.
- [6] R.L. Milidiú, A.A. Pessoa, E.S. Laber, Transporting petroleum products in pipelines, in: ISMP 2000—17th Internat. Symp. on Mathematical Programming, Atlanta, Georgia, USA, 2000, pp. 134–135 (abstract).
- [7] R.L. Milidiú, A.A. Pessoa, E.S. Laber, Pipeline transportation of petroleum products with no due dates, in: Proc. of the LATIN'2002, Cancún, Mexico, 2002, pp. 248–262.
- [8] R.L. Milidiú, A.A. Pessoa, E.S. Laber, Complexity of makespan minimization for pipeline transportation of petroleum products, in: Proc. of the APPROX'2002, Rome, Italy, 2002, pp. 243–255.