



HAL
open science

Parameterized complexity and approximability of coverability problems in weighted Petri nets

Dimitri Watel, Marc-Antoine Weisser, Dominique Barth

► **To cite this version:**

Dimitri Watel, Marc-Antoine Weisser, Dominique Barth. Parameterized complexity and approximability of coverability problems in weighted Petri nets. International Conference on Application and Theory of Petri Nets and Concurrency, Jun 2017, Zaragoza, Spain. pp.330-349, 10.1007/978-3-319-57861-3_19 . hal-01459206

HAL Id: hal-01459206

<https://hal.science/hal-01459206v1>

Submitted on 7 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parameterized complexity and approximability of coverability problems in weighted Petri nets

Dimitri Watel¹, Marc-Antoine Weisser², and Dominique Barth³

¹ SAMOVAR, Evry, France

² LRI, CentraleSupélec, Université Paris-Saclay, Orsay, France

³ DAVID, Université Versailles Saint-Quentin-en-Yvelines, Versailles, France

This paper is submitted as a Regular paper.

Abstract. Many databases have been filled with the chemical reactions found in scientific publications and the information associated with (efficiency, chemical products involved...). They can be used to define functions representing costs such as the ecological impact of the reactions. A major challenge is to use computer driven optimization in order to improve synthesis process. The objective is to provide algorithms to help determining a pathway (series of reactions) for the synthesis of a molecule. Usually, a chemist proposes a pathway adapted from an existing one. Our goal is to determine optimum pathways from the reactions recorded in the database.

As, the classical Petri nets do not allows us to consider the optimization component, a weighted model has to be defined and the complexity of the associated problems studied. In this paper we introduce the weighted Petri nets in which each transition is associated with a weight. We define the Minimum Weight Synthesis Problem: find a minimum weight series of transitions to fire to produce a given target component. It mainly differ from classical coverability as it is an optimization problem.

We prove that this problem is decidable but EXPSpace-hard and that there is no polynomial approximation even when both in and outdegree are fixed to two and the target state is a single component. We also consider a more constraint version of the problem limiting the number of fired transitions. We prove this problem falls into PSPACE and the parametrized versions into XP but it remains not approximable.

Keywords: Petri net coverability problem, Minimum weight synthesis problem, Parameterized complexity, Approximability

1 Introduction

The development of decision support tools for the synthesis of new molecules is a major challenge for organic chemistry and biochemistry [7,17], through the use of databases of reactions that are regularly updated in these scientific fields (such as REAXYS, CHEBI [1,13]). Indeed, the objective is to provide algorithms to help determining a reaction pathway (serie of consecutive reactions) for the synthesis of new molecules by using such reaction databases [7,9]. Usually, a chemist proposes a reaction pathway from a synthesis in the database

of a molecule sufficiently similar to the target molecule [16,21]. It is therefore important to determine the optimum reaction pathway of this similar molecule from the reactions recorded in this target database in which each reaction can be associated with a cost depending on the complexity of the reactions or the solvents involved. Thus, the initial molecules in a reaction pathway are usually easy to synthesize or to be bought. Given the cost of these potentially initial molecules and the list of all referenced chemical reactions, the problem is to identify best process to obtain the target molecules.

In this article, we are considering to optimize the cost to obtain or synthesize the target molecules but it can be any additive function such as the ecological footprint if it can be estimated or the time needed if there is no parallelization.

As we say previously, many such chemical databases are used and updated (for example [1,13]), filled with million of reactions found in scientific publications. These databases are maintaining many informations such as the efficiency or the chemical products involved in the reaction which may not be reactive but nevertheless necessary (solvent for example). They can be used to define functions representing costs such as environmental impact of the reactions.

This challenge of improving synthesis process by using computer driven optimisation is not specific to chemistry. It can be found in all manufacturing areas as long as one can describe the process as a set of transformations and evaluate the cost of buying resources and the cost of transforming them. We can even find this problem in games. In most of the MMORPG (massively multiplayer online role-playing game), the player have the ability to craft items using resources they find or buy in the game.

Petri nets were introduced by Petri [22] and is a classical model to describe chemical and biological processes [6,12,14,15,19,24]. More generally, it is a way to model transformations of molecular components. Each transformation destroys some components (substrates) and generates some others (products). Considering a quantity (or stock) of such components, represented by a vector containing for each component a non-negative quantity, a transformation may be fired to change this stock into another one.

One of the classical problems with Petri net related to our synthesis problem is the *Coverability* of a stock : starting with an initial stock, is it possible, with a finite list of transformations, to get another stock containing a given target stock? The Coverability problem is decidable [10,11] but requires at least an exponential space complexity in general nets [18,20]. However, in this context, classical Petri nets are not adapted anymore, a weighted model has to be defined in order to take into account the costs of buying molecules and making the reactions.

To represent this crafting problem with a Petri net, we introduce the weighted Petri nets where each transformation is associated with a non-negative weight. This weight allows us to represent both the cost of a transformation and the cost of purchasing a component: a transition transforming an empty set of component into a non empty set. These are two important differences with the coverability problem, thus to avoid confusion, we refer to synthesis problem. The *Minimum Weight Synthesis Problem* (Min-WSP) consists in determining, starting with no

component in the stock, which transformation should be fired in order to get, at minimum weight, a stock containing a given target stock.

We have given the practical reasons for our interest in this problem. However, we think that given its simplicity, in terms of definition, it is a problem that deserves to be studied even if given the complexity of coverability problems in classical Petri nets, one can not expect anything that hardness results. Finally, as we have seen with crafting in MMORPG, many optimization problems are linked with the Min-WSP.

Related works. The Priced Petri nets are an extension of standard Petri nets [2,3,4]. It consists in a net where transitions have a cost and possibly an aging effect on the current stock. When a token ages, a conservation cost must also be paid. Given an input stock and a target stock, the Priced Coverability problem consists in the search for a minimum cost sequence of transitions transforming the input stock into a stock containing the target stock. This associated decision problem is decidable if and only if the costs are non-negative. This is an extension of our weighted synthesis problem as the conservation cost of the molecules is null and we do not take time into account. Abdulla *et al.* studied the decidability of the priced Petri nets problems. Our work considers the approximability and parameterized complexity of our model.

In the works of Abdulla *et al.* or ours, we introduce an optimization component into the problem which try to capture a chemical property. This has also been done in recent work. For example, [5] introduces an NP-Complete problem in which, given an initial stock and some target place, we search for a reachable stock containing a maximum number of tokens in the target place.

Our contribution. In this paper, we focus on two optimization versions of the Petri Net Coverability problem. The first one is the above-mentioned *Minimum Weight Synthesis Problem* (Min-WSP). We study the complexity, the approximability from a chemist point of view in a sense that we also consider firstly the fact that a chemical reaction has neither a high number ρ of reactants nor a high number π of products and secondly the fact that the number C of target molecules is small. Consequently, in addition to the classical complexity and approximability studies, we also deal with the parameterized complexity and approximability with respect to ρ , π and C .

In Section 3, we prove that Min-WSP is decidable and that determining if an instance of Min-WSP has a feasible solution is polynomial. However, we also prove this problem to be EXPSPACE-hard and not polynomially approximated to within a polylogarithmic ratio even if we consider the parameterized versions where $\pi = \rho = 2$ and $C = 1$.

In the second problem, called *Minimum Limited sequence Weighted Synthesis problem* (Min-LWSP), we also search for a minimum transformation cost to cover a target stock. However, contrary to Min-WSP, we consider a human constraint in the sense that we limit the number of transitions we can fire so that the number l of chemical reactions we need and the number of molecules we have

to buy in order to synthesis a molecule is reasonable. We study how this new parameter l affects the results of the first problem.

Table 1 summarizes the complexity results for Min-WSP and Min-LWSP given in Sections 3 and 4.

Table 1. Summary of the results of the paper. The four first lines concern Min-WSP and the others concern Min-LWSP. The four first columns indicate how the parameters of the problems are considered for the complexity: a unary entry, a binary entry, a fixed constant or a fixed value. Note that l is not defined for Min-WSP thus for the four first lines. The columns Opt. sol. and Feas. sol. respectively specify if the results stands for the search of an optimal solution, the search of the existence of a feasible solution or both.

	l	ρ	π	C	Opt. sol.	Feas. sol.	Result	
Min-WSP	-				×		Decidable	Th. 1
	-	2	2	1	×		EXSPACE-Hard	Th. 2
	-	2	2	1	×		Polylog-Inapprox	Th. 5
	-						P	Th. 3
Min-LWSP	Binary	2	2	1	×	×	PSPACE-Complete	Th. 7
	Unary	2	2	1	×	×	NP-Complete	Th. 8
	<i>ct.</i>				×	×	XP	Th. 9
	<i>ct.</i>	<i>ct.</i>			×	×	W[2]-Hard	Th. 10
	<i>ct.</i>			<i>ct.</i>	×	×	W[2]-Hard	Th. 10
	<i>ct.</i>		<i>ct.</i>			×	W[1]-Complete	Th. 12
	<i>ct.</i>	<i>ct.</i>	<i>ct.</i>	<i>ct.</i>	×		W[1]-Hard	Th. 11

The hardness results for deciding the existence of a feasible solution of an instance of Min-LWSP immediately proves hardness of approximability for the optimization problem.

The next section is dedicated to the formal definitions of the terminology. In addition, it gives drawing conventions of this paper and details the parameters we focus on in the parameterized complexity study. Sections 3 and 4 are respectively dedicated to the studies of Min-WSP and Min-LWSP.

2 Definitions

This section is dedicated to the formal definitions of the terminology we use and of the problems we study in this paper.

2.1 The Petri Net Coverability problem

A Petri net is a triplet $(\mathcal{P}, \mathcal{T}, M_0)$ where \mathcal{P} is the finite set of *places* and \mathcal{T} is the finite set of *transitions*. Each place may contain zero, one or more *tokens*. A *state* M maps \mathcal{P} to \mathbb{N} , each place x is associated with a non-negative number

$M(x)$ of tokens. We write, using classical vector notation [18], $M = \sum_{x \in \mathcal{P}} a_x \cdot x$ where $a_x = M(x)$. We may possibly remove a place x from the sum if a_x is null. Particularly, an empty state is denoted by \emptyset . M_0 is called the *initial state* of the Petri net: it defines the initial number of tokens in each place. Each transition $t \in \mathcal{T}$ is a couple of states $t^- = \sum_{x \in \mathcal{P}} t^-(x) \cdot x$ and $t^+ = \sum_{x \in \mathcal{P}} t^+(x) \cdot x$, respectively called the *input state* and the *output state* of t . The set of places for which $t^-(x) > 0$ and the set of places for which $t^+(x) > 0$ are respectively called the *input states* and the *output states*. We denote the transition by $t = t^- \rightarrow t^+$.

We say a state M' *covers* another state M if and only if, for all $x \in \mathcal{P}$, $M'(x) \geq M(x)$. Considering a state M of the Petri net, one can *fire* a transition $t \in \mathcal{T}$ if M covers the input state t^- . We say t is *enabled* at M . In that case, the resulting state is $M' = \sum_{x \in \mathcal{P}} (M(x) - t^-(x) + t^+(x)) \cdot x$. We denote this by $M \Rightarrow^t M'$. In other word, firing a transition means transforming some tokens in the input places into tokens in the output places. Similarly, we can define a *sequence* $T = (t_1, t_2, \dots, t_{|T|})$. This sequence is *enabled at* M if we can successively fire all the transitions of T from t_1 to $t_{|T|}$. In that case, and if firing this sequence produce the state M' , we write $M \Rightarrow^T M'$. Note that the sequence T may contain a transition t more than once.

We can now formally define the classical coverability problem in Petri nets.

Problem 1 (Coverability). Given a Petri net $(\mathcal{P}, \mathcal{T}, M_0)$ and a *target state* \mathcal{C} , is there an enabled sequence T at M_0 such that $M_0 \Rightarrow^T M$ and M covers \mathcal{C} ?

We represent an instance of this problem by a bipartite directed graph in which the places are circle nodes and the transitions are squared nodes. For each transition nodes t , we represent the input and output state using incoming and respectively outgoing arcs. An arc (x, t) (resp. (t, x)) is associated with the value $t^-(x)$ (resp. $t^+(x)$). We omit this value when it is 1 and we omit the arc when it is 0. For each target place, i.e. a place x for which $\mathcal{C}(x) \neq 0$, is drawn with a double circle. A dashed outgoing arc, from x , is associated with the number $\mathcal{C}(x)$. Given a state M , $M(x)$ is represented by a value inside the circle node x . An example is given on Figure 1.

2.2 The Weighted Petri Net Synthesis problem

The Minimum Weighted Synthesis Problem (Min-WSP) insert an optimization part to the previous problem. Each transition $t \in \mathcal{T}$ is associated with a non-negative weight $\omega(t) \in \mathbb{R}^+$. In addition, the initial state M_0 is always \emptyset , there is not any token in any place. However, note that the Petri net may contain transitions for which the input state is empty.

We define the weight $\omega(T)$ of a sequence T of transitions of \mathcal{T} as the sum $\omega(T) = \sum_{t \in T} \omega(t)$.

Problem 2 (Minimum Weighted Synthesis, Min-WSP). Given a Petri net $(\mathcal{P}, \mathcal{T}, \emptyset)$, a weight function $\omega : \mathcal{T} \rightarrow \mathbb{R}^+$ over the transitions and a *target state* \mathcal{C} , return

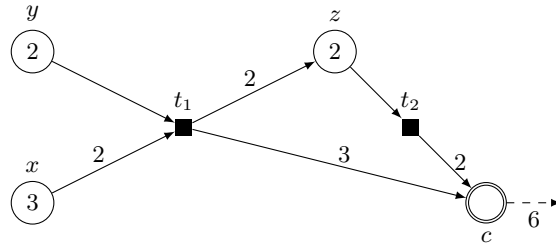


Fig. 1. This figure illustrates an instance of the coverability problem: a Petri Net with four places x, y, z, c and two transitions $t_1 = 2x + y \rightarrow 2z + 3c$ and $t_2 = z \rightarrow 2c$. The initial state M_0 is $3x + 2y + 2z$. After firing t_1, t_2, t_2 we obtain the state $M = x + 7c$ which is covering the target state $6c$.

a enabled sequence T at \emptyset of \mathcal{T} such that $\emptyset \Rightarrow^T M$, M covers \mathcal{C} , and $\omega(T)$ is minimum.

In addition, we denote by FS-MWSP the problem of determining if there exists a feasible solution for an instance of Min-WSP.

We use a representation similar to the classical Petri nets. We add a table indicating the weight of each transition. An example of a Min-WSP instance if is given in figure 2.

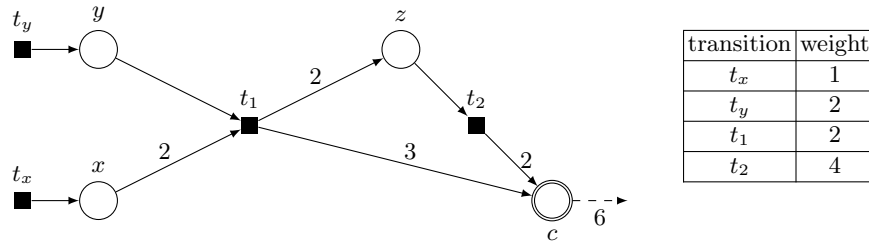


Fig. 2. This figure illustrates a instance of Min-WSP. It is a Petri Net with four places x, y, z, c and four transitions $t_x = \emptyset \rightarrow x$, $t_y = \emptyset \rightarrow y$, $t_1 = 2x + y \rightarrow 2z + 3c$ and $t_2 = z \rightarrow 2c$. The initial state M_0 is \emptyset . After firing $(t_y, t_y, t_x, t_x, t_x, t_x, t_1, t_1)$ of weight 12 we obtain the state $M = 7c$ which is covering the target state $6c$.

2.3 Parametrized version of Min-WSP

In order to analyze the parametrized complexity of the Petri Net Synthesis problem, we introduce four different parameters: ρ and π , respectively the maximum number of input and output tokens among all the transitions of \mathcal{T} , C the size of the target state and l the number of transitions fired in a feasible solution. As

the three first are constraints on the instances, the last is a constraint on feasible solutions so we will treat it independently.

The two first parameters, ρ and π , limit the maximum number of input and output tokens. More formally,

$$\rho = \max_{t \in \mathcal{T}} \sum_{x \in \mathcal{P}} t^-(x)$$

$$\pi = \max_{t \in \mathcal{T}} \sum_{x \in \mathcal{P}} t^+(x)$$

The third parameter on which we focus is the size $C = \sum_{x \in \mathcal{P}} \mathcal{C}(x)$ of the target state \mathcal{C} . From a chemical perspective, these parameters model the limited number of reactants, products of reactions and the number of molecules we want to synthesize, which is likely to be small.

Finally, we define a fourth parameter l to limit the number of molecules we can buy and the number of reactions we can perform to synthesize the target molecules. As introducing this parameter changes the set of feasible solutions for an instance (see Example 1), we define an independent problem: **Minimum Limited sequence Weighted Synthesis problem (Min-LWSP)**.

Example 1. We constraint the instance of Figure 1 by introducing a parameter $l = 6$ to limit the number of fired transitions. As a consequence the sequence $T = (t_y, t_y, t_x, t_x, t_x, t_x, t_1, t_1)$ is not a feasible solution anymore as it contains 8 transitions. This constraint instance contains only one feasible solution: the sequence $T' = (t_y, t_x, t_x, t_1, t_2, t_2)$ of weight 14.

Problem 3. Limited sequence Weighted Petri Net Synthesis problem (Min-LWSP). Given a Petri net $(\mathcal{P}, \mathcal{T}, \emptyset)$, a weight function $\omega : \mathcal{T} \rightarrow \mathbb{R}^+$ over the transitions, a *target state* \mathcal{C} and an integer l , return a enabled sequence T at \emptyset of \mathcal{T} such that $\emptyset \Rightarrow^T M$, M covers \mathcal{C} , $|T| \leq l$ and $\omega(T)$ is minimum.

Similarly to FS-MWSP, we define FS-MLWSP the problem of determining if there exists a feasible solution for an instance of Min-LWSP.

3 Min-WSP

In this section, we study the complexity, the approximability and the parameterized approximability of Min-WSP.

3.1 Complexity and parameterized complexity

The two next theorems give an upper bound and a lower bound of the complexity of Min-WSP: the problem is decidable but EXPSPACE-hard.

Theorem 1 ([2]). *Min-WSP is decidable.*

Theorem 2. *Min-WSP is EXPSPACE-hard even if $\pi = \rho = 2$ and $C = 1$.*

Remark 1. The theorem 2 shows that Min-WSP is not easier if we fix the three parameters π, ρ and C . As a consequence, there is no parameterized results considering those three parameters.

Proof. The proof of EXPSPACE-hardness is a reduction from the Petri net Coverability problem. This problem is EXPSPACE-hard [18,20]. Note that this result remains true if the Petri net satisfies $\pi = \rho = 2$ and if the target state contains a polynomial number C of tokens. We can create a equivalent instance with only one target token and $\pi = \rho = 2$ by firstly adding a new place c and a new transition using all the C target tokens to create one token in c . Finally, transform this transition into $C - 1$ transitions and add $C - 2$ new places as shown in Figure 3 so that ρ remains 2. As C is polynomial, this transformation is polynomial too.

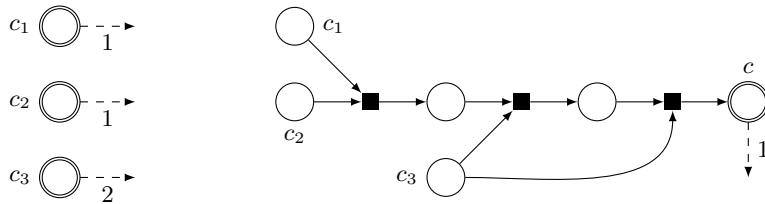


Fig. 3. Example of transformation from an instance (on the left) in which the target state $\mathcal{C} = c_1 + c_2 + 2 \cdot c_3$ (and then $C = 4$) to an instance (on the right) satisfying $C = 1$. The weight of every transition is 0.

Let $\mathcal{I} = ((\mathcal{P}, \mathcal{T}, M_0), \mathcal{C})$ be an instance of the Petri net Coverability problem. We create an instance \mathcal{J} of Min-WSP in which the Petri net is $(\mathcal{P}, \mathcal{T} \cup \{t\}, \emptyset)$ in which we add a transition t which creates $M_0(x)$ tokens in x for every $x \in \mathcal{P}$ from nothing. The weight of t is 1 and the weight of each other transition is 0. Finally, the target state is \mathcal{C} . Then \mathcal{C} is coverable in \mathcal{I} if and only if there is a feasible solution of weight 1 in \mathcal{J} . As a consequence the decision version of Min-WSP is EXPSPACE-hard. \square

This last theorem shows that the problem of determining if there exists a feasible solution for an instance of Min-WSP is polynomial.

Theorem 3. *FS-MWSP is polynomial.*

Proof. In order to find a feasible solution, for each transition t for which the input state is empty, we mark all the output places of t . Then, for each transition t for which all the input places are marked, we mark all the output places. We repeat this marking operation either until no new place is marked. There is a feasible solution if and only if all the target places are marked. \square

This means that it is conceivable to build approximation algorithms for this problem. However, we show in the next subsection a high inapproximability ratio. Moreover, note that Theorem 3 does not explain how to build a feasible solution. In fact, even if we can easily adapt the algorithm given in the proof to return such a solution, for some instances, no feasible solution contains a polynomial number of transitions. See, for example, the instance on Figure 4.

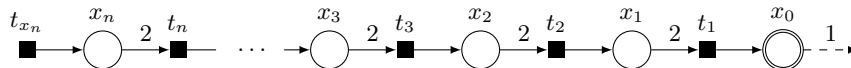


Fig. 4. Example of instance for which the number of transitions is exponential. Indeed, in order to put a token in the place x_0 , we need to buy 2^n tokens for x_n using the transition t_{x_n} , then fire 2^{n-1} times the transition t_n , then fire 2^{n-2} times the transition t_{n-1} , and so on until we fire once the transition t_1 . This sequence contains then 2^n transitions.

3.2 Approximability

We show in this subsection that Min-WSP cannot be approximated within a polylogarithmic ratio in polynomial time. Basically, the best ratio an approximation algorithm may achieve is at least polynomial in the size of the instance. In order to prove this inapproximability result for Min-WSP, we build, in this subsection, a reduction from the MMSA_h problem.

Problem 4. Minimum Monotone Satisfying Assignment (MMSA_h). Given a set Y of boolean variables and a monotone boolean formula φ of depth h (this formula has h levels of alternating AND and OR gates, where the top levels is an AND gate), minimize the number of true variables such that φ is satisfied.

We respectively call $D(\varphi)$ and $C(\varphi)$ the set of OR and AND gates in the formula.

Theorem 4 ([26]). *Unless $P = NP$, there is no polynomial approximation for MMSA_3 with a ratio $Q(\log(|Y|), \log(|D(\varphi)|), \log(|C(\varphi)|))$ where Q is any polynomial.*

We now detail and prove a reduction from MMSA_h to Min-WSP. An example is given on Figure 5. Let $\mathcal{I} = (Y, \varphi)$ be an instance of MMSA_h , we build the following instance \mathcal{J} :

- add one place x_0 and a transition $t_0 = \emptyset \rightarrow x_0$;
- for each variable y , add one place x_y and a transition $t_y = x_0 \rightarrow l_y \cdot x_y$ where l_y is the number of literals y in φ ;
- add one place x_g for each gate g ;

- for each OR gate $g = \bigvee_i \varphi_i$, where φ_i are either gates or variables, add, for each i , a transition $t_g^i = x_{\varphi_i} \rightarrow x_g$;
- for each AND gate $g = \bigwedge_i \varphi_i$, where φ_i are either gates or variables, add a transition $t_g = \sum_i x_{\varphi_i} \rightarrow x_g$;
- set the target state to $\mathcal{C} = x_g$ where g is the top level AND gate;
- set the weight of t_0 to 1 and the weight of every other transition to 0.

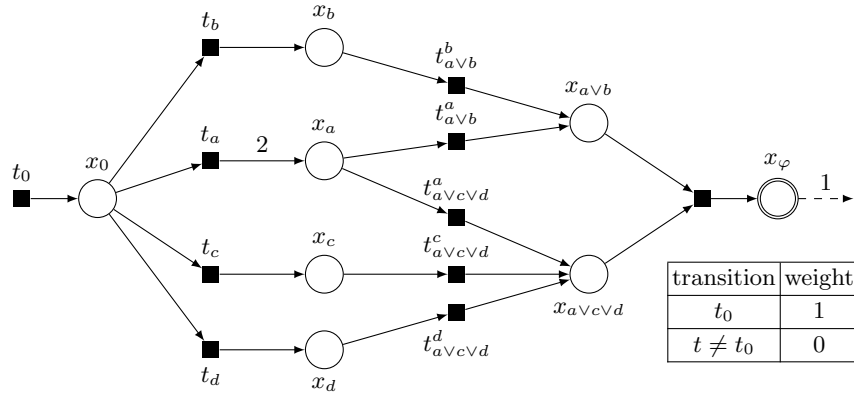


Fig. 5. Example of reduction from MMSA_2 to Min-WSP with the boolean formula $\varphi = (a \vee b) \wedge (a \vee c \vee d)$.

Lemma 1. *If there is a feasible solution for \mathcal{I} with $\omega \geq 1$ true variables, there is a feasible solution for \mathcal{J} of weight ω .*

Proof. The solution consists in firing ω times t_0 (this way, we put ω tokens in the place x_0) and then using all the transitions of the true gates and variables in the formula, starting with the variables and the lower gates and terminating with the top level AND gate. The proof that, when a transition is fired, all the input places contain a token (and then that the solution is feasible) can be done by induction on the height of each gate. \square

Lemma 2. *If there is a feasible solution for \mathcal{J} of weight $\omega \geq 1$, there is a feasible solution for \mathcal{I} with at most ω true variables.*

Proof. Let $T = (t_1, t_2, \dots, t_{|T|})$ be a feasible solution for \mathcal{J} of cost $\omega \geq 1$. We now prove that assigning true to all the variables $y \in Y$ such that $t_y \in T$ is a feasible solution for \mathcal{I} of cost at most ω . Let f be the boolean function associating to each variable and each gate the value *true* or *false*. We have to check that $f(g)$ is true where g is the top level AND gate.

Firstly, there are at most ω distinct variables for which the transition is fired in T . Indeed, in order to fire the transition t_y associated with the variable $y \in Y$,

we have to fire the transition t_0 of weight 1 first in order to put a token in x_0 . As the weight of T is ω , there are at most ω variables set to true.

Secondly, for each $1 \leq i \leq |T|$ such that $t_i \neq t_0$, the transition t_i is associated either with a variable y or a gate g . Considering how we set the true and false variables in \mathcal{I} , if the transition is associated with a variable y , then $f(y)$ is true. We now prove by induction on i that, on the other case, $f(g)$ is true. Let $\varphi(t)$ be the variable or the gate associated with the transition t , for $t \neq t_0$. Similarly, $\varphi(x)$ is the variable or the gate associated to the place x , if $x \neq x_0$. Note that the first transition t_i for which $t_i \neq t_0$ is necessarily associated with a variable. We now assume that, for every $j \leq i$, either $t_j = t_0$ or $f(\varphi(t_j))$ is true. We also assume, without loss of generality that $t_{i+1} \neq t_0$ and that $g = \varphi(t_{i+1})$ is an OR gate: $g = \varphi_1 \vee \varphi_2 \vee \dots \vee \varphi_k$. Thus, t_{i+1} has one input place $x \neq x_0$ such that $\varphi(x) = \varphi_\kappa$ for some $\kappa \in \llbracket 1; k \rrbracket$ (and $t_{i+1} = t_g^{\varphi_\kappa}$). Before this transition is fired, x contains a token otherwise T would not be a feasible solution for \mathcal{J} . As $x \neq x_0$, the token in x was added by a reaction $t_j \neq t_0$ for some $j \leq i$ and $\varphi(x) = \varphi(t_j) = \varphi_\kappa$. By the inductive hypothesis, $f(\varphi(t_j))$ is true, then $f(g)$ is true. A similar argument occurs if g is an AND gate. Consequently, there is a feasible solution for \mathcal{I} of cost at most ω . \square

Theorem 5. *Unless $P = NP$, there is no polynomial approximation for Min-WSP with a ratio $Q(\log(|\mathcal{P}|), \log(|\mathcal{T}|))$ even if $\pi = \rho = 2$ and $C = 1$ where Q is any polynomial.*

Proof. Let Q be a polynomial. We assume that the cost of an optimal solution for \mathcal{I} is ω^* . We can use the previous reduction to build an instance \mathcal{J} . By Lemmas 1 and 2, the cost of an optimal solution for \mathcal{J} is ω^* . If there is a $Q(\log(n), \log(m))$ -approximation for Min-WSP, we can use it to build a solution for \mathcal{J} of cost at most $Q(\log(n), \log(m)) \cdot \omega^*$, and then use Lemma 2 to build a feasible solution for \mathcal{I} of cost at most $Q(\log(n), \log(m)) \cdot \omega^*$.

Finally note that, in the reduction $m = |C(\phi)| + |D(\phi)|(|D(\phi)| + |C(\phi)| + |Y|) + |Y|$ et $n = 1 + |C(\phi)| + |D(\phi)| + |Y|$, thus there is a polynomial Q' such that $Q(\log(n), \log(m)) \leq Q'(\log(|Y|), \log(|D(\phi)|), \log(|C(\phi)|))$. As a consequence, we have built a $Q'(\log(|Y|), \log(|D(\phi)|), \log(|C(\phi)|))$ -approximation for MMSA_h . By Theorem 4, this is a contradiction. \square

4 Min-LWSP

In this section, we study the complexity, the approximability and the parameterized approximability of Min-LWSP.

4.1 Complexity

Theorem 6. *The decision versions of Min-LWSP and FS-MLWSP are in PSPACE.*

Proof. Let K be any integer. We search for the existence of a feasible solution of weight less than K . To achieve this, we provide a non-deterministic algorithm that runs in polynomial space.

We start with a state where all the places are empty. At each iteration, we non-deterministically choose a transition t . If t is not enabled at the current state or if the maximum weight K is lower than the current weight, then return NO, otherwise, fire the t . If $l + 1$ transitions were fired, return NO. If the target state is covered, return YES. If none of those cases occur, we start a new iteration.

At each iteration, the algorithm must store, for each place, the number of tokens in that place. Those numbers are no more than $l \cdot \pi$ as a transition can produce at most π tokens. It must also store the current total weight and the number of the current transition, which are respectively no more than K and l . Consequently, this algorithm runs in polynomial space and Min-LWSP is in NPSPACE.

As PSPACE = NPSPACE [25], Min-LWSP is in PSPACE. The proof for FS-MLWSP is exactly the same except that we do not have to consider the weight. \square

Theorem 7. *The decision versions of Min-LWSP and FS-MLWSP are PSPACE-complete, even if $\rho = \pi = 2$ and $C = 1$.*

Proof. The proof of Theorem 1 of [18], page 287, gives a reduction from the Turing halting problem in polynomial space to the Reachability problem (and the Coverability problem) for 1-Conservative Petri nets with $\rho = \pi = 2$ and $C = 1$. The same reduction can be used to prove our theorem. Some states encode the tape and others encode the states of the automata. Each transition function is encoded by a transition of the Petri net of weight 0. As the machine cannot use more than $P(|x|)$ cells of the tape, where P is a polynomial and x is the input word, it cannot use more than $2^{P(|x|)}|Q|$ transitions where $|Q|$ is the size of the automata otherwise the machine loops indefinitely. Thus the Petri net must cover the target state by firing at most that number of transitions. The initial state of the Petri net of the proof of [18] encodes the word x written on the tape and the initial state of the automata. In this version, we add a transition t_0 of weight 1 that creates that initial state from nothing. We can then transform t_0 into multiple transitions as we did in Theorem 2 so that π remains 2. The machine halts without using more than $P(|x|)$ cells if and only if the Petri net covers the target state with at most $1 + 2^{P(|x|)}|Q|$ transitions of total weight at most 1.

Thus the decision version of Min-LWSP is PSPACE-hard. By Theorem 6, the decision version of Min-LWSP is PSPACE-Complete, even if $\rho = \pi = 2$ and $C = 1$.

In order to prove that FS-MLWSP is also PSPACE-hard, we must prevent the transition t_0 from being fired twice in a feasible solution. To do so, we add a place x_0 as input of t_0 and a construction as the one given in Figure 4 so that we need to fire at least $2^{P(|x|)+1}|Q|$ transitions in order to put a token in the place x_0 . We then set $l = 2^{P(|x|)+1}|Q| + 2^{P(|x|)}|Q| + 1$ so that any feasible solution cannot place more than one token in x_0 and then cannot fire t_0 twice. \square

The same ideas show the following theorem.

Theorem 8. *If the encoding of l is unary, the decision versions of Min-LWSP and FS-MLWSP are NP-Complete, even if $\rho = \pi = 2$ and $C = 1$.*

4.2 Parameterized complexity and approximability

We present in this section four theorems describing the parameterized complexity of Min-LWSP and FS-MLWSP. There are proved using reduction from and to the set cover problem and the partitioned clique problem.

Problem 5. Set Cover. Given an integer K , a set E and a set $S \subset 2^E$, find a subset S' of S covering E such that $|S'| \leq K$.

Problem 6. Partitioned clique Given an undirected graph $G = (V, E)$ and a partition V_1, V_2, \dots, V_k of V , find a clique of size k with exactly one node in each set V_i .

Set Cover problem is W[2]-Complete with respect to the parameter K [8]. Partitioned clique problem is W[1]-complete with respect to the parameter k [23].

Theorem 9. *Min-LWSP is XP with respect to the parameter l .*

Proof. Indeed, one can search for an optimal solution by exhaustively enumerate all the m^l sequences of transitions. \square

Theorem 10. *FS-MLWSP is W[2]-hard with respect to the parameters l and ρ and is W[2]-hard with respect to the parameters l and C .*

Proof of Theorem 10. This proof is an FPT reduction from the set cover problem parameterized with K . Let $\mathcal{I} = (K, E, S)$ be an instance of Set Cover. We now build an instance \mathcal{J} of FS-MLWSP parameterized with l and ρ .

We create for each element e in E a place x_e . For each set s in S , we create a transition $t_s = \emptyset \rightarrow \sum_{e \in s} x_e$. The target state is $\mathcal{C} = \sum_{e \in E} x_e$. We finally set $l = K$.

This reduction satisfies $\rho = 0$. A feasible solution S' for \mathcal{I} can be transformed into a feasible solution for \mathcal{J} by firing every transition t_s for $s \in S'$ and, conversely, from a feasible solution for \mathcal{J} , we can build a feasible solution for \mathcal{I} by selecting the sets for which the transition is fired. As this reduction is FPT with respect to K , FS-MLWSP is W[2] with respect to l and ρ .

If we now add to this instance a new place c and a new transition $\sum_{e \in E} x_e \rightarrow c$; and if the target state is $\mathcal{C} = c$, we get an instance of FS-MLWSP in which $C = 1$. Note that ρ does not polynomially depend on the parameter K . As this reduction is also FPT with respect to K , FS-MLWSP is W[2] with respect to l and C . \square

Theorem 11. *FS-MLWSP is W[1]-hard with respect to the parameters l , ρ , π and C .*

Proof. Given an instance $\mathcal{I} = (G, V_1, V_2, \dots, V_k)$ of Partitioned Clique, we now build the following instance \mathcal{J} of FS-MLWSP parameterized with l , ρ , π and C :

- add one place x_0 and a transition $t_0 = \emptyset \rightarrow x_0$;
- add one place x_v and a transition $t_v = x_0 \rightarrow (k-1) \cdot x_v$ for each node $v \in V$;
- add one place $c_{i,j}$ for each $i < j \in \llbracket 1; k \rrbracket$;
- add, for each edge $e = (u, v)$ such that $u \in V_i, v \in V_j$ and $i < j$, a transition $t_e = x_u + x_v \rightarrow c_{i,j}$;
- set the target state to $\mathcal{C} = \sum_{i=1}^{k-1} \sum_{j=i+1}^k c_{i,j}$;
- set l to $2k + \frac{k \cdot (k-1)}{2}$.

An example is given Figure 6.

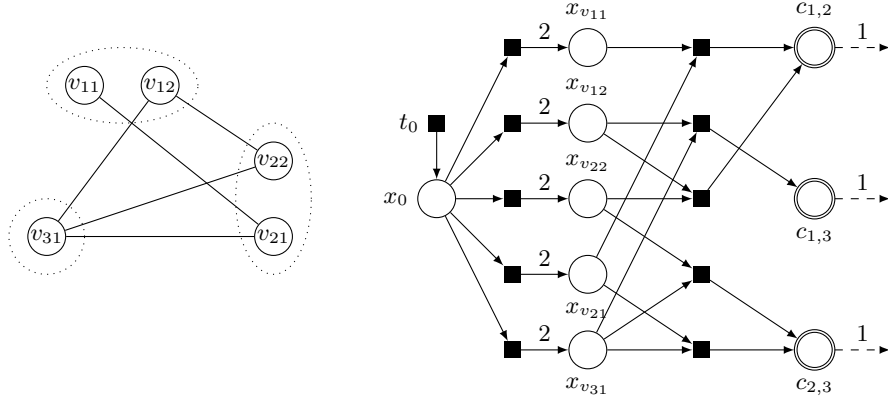


Fig. 6. Example of reduction from Partitioned Clique to FS-MLWSP. On the left, the three dashed ellipses describe the partition of the nodes.

This reduction is polynomial and l , C , ρ and π depends only on k . Thus it is FPT with respect to k .

If there is a clique (v_1, v_2, \dots, v_k) of size k in G satisfying $v_i \in V_i$ for $i \in \llbracket 1; k \rrbracket$, there is a feasible solution for \mathcal{J} consisting in buying k tokens for the place x_0 with the transition t_0 , firing the k transitions t_{v_i} for $i \in \llbracket 1; k \rrbracket$ and then firing the $\frac{k \cdot (k-1)}{2}$ transitions $t_{(v_i, v_j)}$ for $i < j \in \llbracket 1; k \rrbracket$. Those transitions exist and are enabled as (v_1, v_2, \dots, v_k) is a clique.

We now assume there is a feasible solution $T = (t_1, t_2, \dots, t_{|T|})$ for \mathcal{J} . It satisfies $|T| \leq l$. Let \mathcal{T}_V and \mathcal{T}_E be respectively the set of transitions $\{t_v, v \in V\}$ and $\{t_e, e \in E\}$. Let $(M_0 = \emptyset, M_1, M_2, \dots, M_{|T|})$ be the successive states of \mathcal{J} after firing the transitions of T . Note that $\emptyset \Rightarrow^T M_{|T|}$, and, as a consequence, this state $M_{|T|}$ contains at least $\mathcal{C}(c_{i,j}) = 1$ token in each place $c_{i,j}$ for $i < j \in \llbracket 1; k \rrbracket$. Only the transitions of \mathcal{T}_E can create such tokens. As none of those transitions create more than one such token, there are, in T , at least $\frac{k \cdot (k-1)}{2}$ transitions of

\mathcal{T}_E . As $l = 2 \cdot k + \frac{k \cdot (k-1)}{2}$, there are at most $2k$ transitions of $\{t_0\} \cup \mathcal{T}_V$ in T . As there must be a token in x_0 before a transition of \mathcal{T}_V can be fired, there are at most k transitions of \mathcal{T}_V in T .

In addition, there is, for each $i \in \llbracket 1; k \rrbracket$, at least one node v_i in V_i and an integer l such that $M_i(x_{v_i}) \geq 1$. Indeed, otherwise, no transition of \mathcal{T}_E placing a token in c_{ji} or c_{ij} for all $j \neq i$ could have been fired. Thus, there are in T exactly k transitions of \mathcal{T}_V . Each one is associated with one node $v_i \in V_i$ for $i \in \llbracket 1; k \rrbracket$.

Finally, (v_1, v_2, \dots, v_k) is a clique, otherwise, some transition $t_{(v_i, v_j)}$ would not exist and consequently no token could have been placed in $c_{i,j}$.

This concludes the reduction. FS-MLWSP is W[1]-Hard with respect to the parameters l, C, π and ρ . \square

We finally prove that FS-MLWSP belongs to the W[1] class if parameterized with l and π . We first give an intermediate lemma proving that, if l and π are fixed, we can consider that ρ and C are fixed too.

Lemma 3. *There exists an FPT-reduction from FS-MLWSP parameterized with l and π to FS-MLWSP parameterized with l, π, ρ and C .*

Proof. As there are at most l transitions in a feasible solution and as every transition may produce at most π tokens, a feasible solution may produce at most $l \cdot \pi$ tokens. Consequently, if $C > l \cdot \pi$ there is no feasible solution and the problem is solved. In addition, if a transition needs more than $l \cdot \pi$ input tokens, this transition may be removed from the instance as it is not possible to fire it. We then necessarily get an instance where $C \leq l \cdot \pi$ and $\rho \leq l \cdot \pi$. \square

Theorem 12. *FS-MLWSP is W[1]-complete with respect to the parameters l and π .*

Due to its length, the complete proof is given in Appendix A. It is based on a reduction to the partitioned clique problem.

Remark 2. Note that Theorem 12 does not prove any parameterized result on Min-LWSP. Determining if Min-LWSP is W[1]-Complete with respect to l and π and with respect to l, ρ, π and C are open questions.

Finally, the complexity of the determining if there exists feasible solutions for Min-LWSP allows us to conclude this section with the following corollary.

Corollary 1. *Let Q be a polynomial.*

- Unless $P = PSPACE$, there is no polynomial approximation for Min-LWSP with a ratio $2^{Q(|\mathcal{P}|, |\mathcal{T}|)}$ even if $\rho = \pi = 2$ and $C = 1$.
- Unless $P = NP$, there is no polynomial approximation for Min-LWSP with a ratio $2^{Q(|\mathcal{P}|, |\mathcal{T}|)}$ even if $\rho = \pi = 2$ and $C = 1$ and if the encoding of l is unary.
- Unless $FPT = W[1]$, there is no FPT approximation for Min-LWSP with ratio $2^{Q(|\mathcal{P}|, |\mathcal{T}|)}$, with respect to l, ρ, π and C .

Proof. Those results are respectively deduced from Theorem 7, 8 and 11. \square

5 Conclusion

In this paper we present the Minimum Weight Synthesis problem, a variant of the classical Coverability problem introducing weights on transitions. We provide a deep analysis of the complexity and parametrized complexity. Our results are summarized in table 1. They illustrate how hard this problem is even when most parameters are constrained. In the last section, we prove that using approximation algorithm for solving this problem is not adequate as there is no constant ratio or small variable ratio approximation algorithm.

From a theoretical perspective, there are a few cases of parameterized complexity which are still open questions. In particular, theorems 10 and 11 provide $W[1]$ -hard and $W[2]$ -hard results. It would be interesting to find others reductions to achieve completeness results.

From a more practical perspective, as the problem is hard to solve and to approximate, we should now study the different heuristic algorithms.

References

1. Reaxys. <https://www.elsevier.com/solutions/reaxys>, 2016. Elsevier.
2. ABDULLA, P. A., AND MAYR, R. Minimal cost reachability/coverability in priced timed petri nets. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5504 LNCS* (2009), 348–363.
3. ABDULLA, P. A., AND MAYR, R. Computing optimal coverability costs in priced timed petri nets. *Proceedings - Symposium on Logic in Computer Science* (2011), 399–408.
4. ABDULLA, P. A., AND MAYR, R. Priced timed petri nets. *Logical Methods in Computer Science* 9, 4 (2013), 1–51.
5. ANDERSEN, J. L., FLAMM, C., MERKLE, D., AND STADLER, P. F. Maximizing output and recognizing autocatalysis in chemical reaction networks is NP-complete. *Journal of Systems Chemistry* 3, 1 (2012), 1–9.
6. ANGELI, D., DE LEENHEER, P., AND SONTAG, E. A Petri Net approach to the study of persistence in chemical reaction networks. *Mathematical Biosciences* 210, 2 (2007), 598–618.
7. BØGEVIG, A., FEDERSEL, H.-J., HUERTA, F., HUTCHINGS, M. G., KRAUT, H., LANGER, T., LÖW, P., OPPAWSKY, C., REIN, T., AND SALLER, H. Route design in the 21st century: The ic synth software tool as an idea generator for synthesis prediction. *Organic Process Research & Development* 19, 2 (2015), 357–368.
8. DOWNEY, R., AND FELLOWS, M. *Parameterized complexity*, vol. 3. Springer-Verlag, 1999.
9. EIGNER-PITTO, V., HUERTA, F. F., HUTCHINGS, M. G., SALLER, H., AND LOEW, P. Reaction prediction tools for both idea generation in new synthesis route planning and for de novo molecule design. In *12th German Conference on Chemoinformatics* (2016).
10. FINKEL, A. The minimal coverability graph for Petri nets. In *Advances in Petri Nets 1993*. Springer Berlin Heidelberg, 1993, pp. 210–243.
11. FINKEL, A., AND LEROUX, J. Recent and simple algorithms for Petri nets. *Software and Systems Modeling* 14, 2 (2015), 719–725.

12. GENRICH, H., KÜFFNER, R., AND VOSS, K. Executable Petri net models for the analysis of metabolic pathways. *International Journal on Software Tools for Technology Transfer* 3, 4 (2001), 394–404.
13. HASTINGS, J., DE MATOS, P., DEKKER, A., ENNIS, M., HARSHA, B., KALE, N., MUTHUKRISHNAN, V., OWEN, G., TURNER, S., WILLIAMS, M., AND STEINBECK, C. The chebi reference database and ontology for biologically relevant chemistry: enhancements for 2013. *Nucleic Acids Research* 41, D1 (2012), D456.
14. HEINER, M. Preface: Petri nets for systems and synthetic biology. *Natural Computing* 10, 3 (2011), 987–992.
15. HOFESTÄDT, R. A petri net application to model metabolic processes. *Systems Analysis Modelling Simulation* 16, 2 (1994), 113–122.
16. HORVATH, D., AND JEANDENANS, C. Molecular similarity and virtual screening. in silico methods to retrieve active analogs in the context of discovering therapeutic compounds. *Actualité Chimique* 9 (2000), 64–67.
17. JOHNSON, A. P., AND MARSHALL, C. Starting material oriented retrosynthetic analysis in the lhasa program. 2. mapping the sm and target structures. *Journal of chemical information and computer sciences* 32, 5 (1992), 418–425.
18. JONES, N. D., LANDWEBER, L. H., AND EDMUND LIEN, Y. Complexity of some problems in Petri nets. *Theoretical Computer Science* 4, 3 (1977), 277–299.
19. KOCH, I. Petri nets—a mathematical formalism to analyze chemical reaction networks. *Molecular Informatics* 29, 12 (2010), 838–843.
20. LIPTON, R. J. The Reachability Problem Requires Exponential Space. Tech. rep., Yale Research Report #63, 1976.
21. NOULEHO, S., BARTH, D., DAVID, O., WATEL, D., AND WEISSER, M. A new definition of molecule similarity to determine molecular construction. In *Poster in 12th German Conference on Chemoinformatics* (2016).
22. PETRI, C. A. *Kommunikation mit Automaten*. PhD thesis, Universität Hamburg, 1962.
23. PIETRZAK, K. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences* 67, 4 (2003), 757–771.
24. REDDY, V. N., MAVROVOUNIOTIS, M. L., AND LIEBMAN, M. N. Petri net representations in metabolic pathways. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology* 1 (1993), 328–36.
25. SAVITCH, W. J. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences* 4 (1970), 177–192.
26. WATEL, D., AND WEISSER, M.-A. A note on the inapproximability of the Minimum Monotone Satisfying Assignment problem. Tech. rep., HAL, <https://hal.archives-ouvertes.fr/hal-01377704>, 2016.

A Proof of Theorem 12

Theorem 12. *FS-MLWSP is $W[1]$ -complete with respect to the parameters l and π .*

Proof. By Lemma 3 and Theorem 11, we have to prove that FS-MLWSP is $W[1]$ with respect to l , ρ , π and C . To do so, we describe a reduction to the partitioned clique problem. Let $\mathcal{I} = ((\mathcal{P}, \mathcal{T}, \emptyset), \omega, \mathcal{C}, l)$ be an instance of FS-MLWSP. We assume that \mathcal{I} contains a fake transition t_0 such that $t_0^- = t_0^+ = \emptyset$ so that \mathcal{I} contains a feasible solution of size lower than l if and only if it contains a feasible solution of size exactly l . We now create an instance $\mathcal{J} = (G, V_1, V_2, \dots, V_l, V_{l+1})$ of the partitioned clique problem such that \mathcal{J} contains a feasible solution if and only if \mathcal{I} does. An example is given in Figure 7.

In the remaining of this proof, given a state M , the ordered set of states $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ is called a *decomposition* of M if and only if $\sum_{j=1}^k \alpha_j = M$. The state α_j may be empty.

Let t be a transition of \mathcal{I} . For each $i \in \llbracket 2; l \rrbracket$, we create at most $(i-1)^\rho \cdot (l-(i+1))^\pi$ nodes in V_i , one per decomposition of t^- in $i-1$ states and per decomposition of t^+ in $l-i+1$ states. If $t^- = \emptyset$, we also create at most l^π nodes in V_1 , one per decomposition of t^+ into l states. Finally, we add at most l^C nodes to V_{l+1} , one for each decomposition of \mathcal{C} into l sets. For example, in Figure 7, $t_1^- = \emptyset$ and $t_1^+ = x + 2y$. We add six nodes to V_1 as there are six possible decompositions of t_1^+ into 2 states: $\{\emptyset, x + 2y\}$, $\{x, 2y\}$, $\{x + y, y\}$ and the symmetrical decompositions. Note that the number of nodes in G is FPT with respect to the parameters l , ρ , π and C .

Let $i < j \in \llbracket 1; l \rrbracket$, and $v_i \in V_i$ and $v_j \in V_j$ be two nodes of G . The node v_i is associated with a transition t_i , a decomposition of t_i^- and a decomposition $\{\alpha_{i+1}, \alpha_{i+2}, \dots, \alpha_{l+1}\}$ of t_i^+ . Similarly, the node v_j is associated with a transition t_j , a decomposition $\{\beta_1, \beta_2, \dots, \beta_{j-1}\}$ of t_j^- and a decomposition of t_j^+ . We add to G the edge (v_i, v_j) if and only if α_j covers β_i .

Similarly, let $i \in \llbracket 1; l \rrbracket$, and $v_i \in V_i$ and $v_{l+1} \in V_{l+1}$ be two nodes of G . The node v_i (respectively v_{l+1}) is associated with a transition t_i , a decomposition of t_i^- and a decomposition $\{\alpha_{i+1}, \alpha_{i+2}, \dots, \alpha_{l+1}\}$ of t_i^+ (resp. a decomposition $\{\beta_1, \beta_2, \dots, \beta_l\}$ of \mathcal{C}). We add to G the edge (v_i, v_{l+1}) if and only if α_{l+1} covers β_i .

For example, in Figure 7, the last node of V_1 and the second node of V_2 are respectively associated with the decomposition $\{x + 2y, \emptyset\}$ of t_1^+ and the decomposition $\{x + y\}$ of t_2^- . The two nodes are linked as $x + 2y$ covers $x + y$.

We now prove that \mathcal{J} contains a feasible solution if and only if \mathcal{I} contains an enabled sequence at \emptyset with exactly l transitions such that the resulting state covers \mathcal{C} .

We first assume there is a clique of G containing a node v_i of V_i for every $i \in \llbracket 1; l+1 \rrbracket$. Let $T = (t_1, t_2, \dots, t_l)$ be the sequence such that t_i is the transition associated with v_i for $i \leq l$. Let $\{\beta_{i,1}, \beta_{i,2}, \dots, \beta_{i,i-1}\}$ and $\{\alpha_{i,i+1}, \alpha_{i,i+2}, \dots, \alpha_{i,l+1}\}$ be respectively the decompositions of t_i^- and t_i^+ associated with v_i and let $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_l\}$ be the decomposition of \mathcal{C} associated with v_{l+1} .

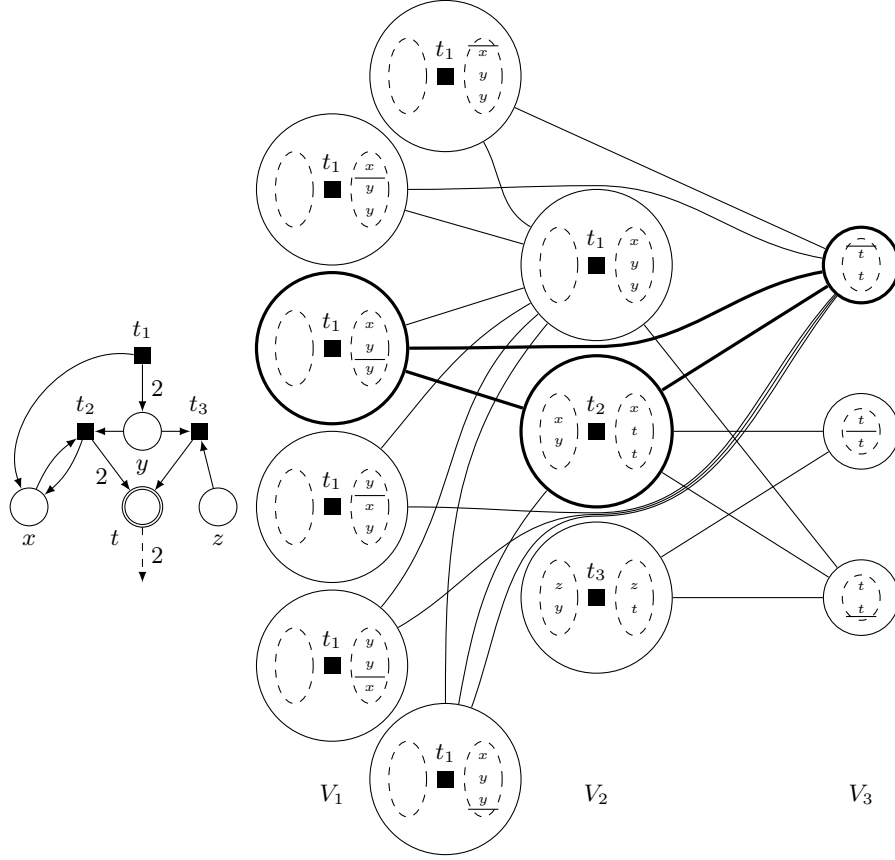


Fig. 7. Example of reduction from an instance \mathcal{I} of FS-MLWSP with $l = 2$ on the left to an instance \mathcal{J} of Partitioned Clique on the right. Due to lack of space, the fake transition is not part of \mathcal{I} . Each node of the clique instance is either a transition t with decompositions of t^- (on the left) and t^+ (on the right) or an decomposition of C into 2 states. The sequence (t_1, t_2) is a feasible solution of \mathcal{I} . The bold clique on the right is a feasible solution of \mathcal{J} associated with that sequence.

We now show that T is enabled at \emptyset . As $v_1 \in V_1$, then $t_1^- = \emptyset$ and t_1 is enabled at \emptyset . Let $i \in \llbracket 1; l-1 \rrbracket$. As $(v_1, v_2 \dots v_i, v_{i+1})$ is a clique of G , then, by construction, $\alpha_{j, i+1}$ covers $\beta_{j, i+1}$ for every $j \in \llbracket 1; i \rrbracket$. For each place x of \mathcal{P} :

$$\begin{aligned}
\sum_{j=1}^i (t_j^+(x) - t_j^-(x)) &\geq \sum_{j=1}^i \left(\alpha_{j, i+1}(x) + \sum_{k=j+1}^i \alpha_{j, k}(x) + \sum_{k=1}^{j-1} \beta_{k, j}(x) \right) \\
&\geq \sum_{j=1}^i \alpha_{j, i+1}(x) + \sum_{j < k \leq i} (\alpha_{j, k}(x) - \beta_{j, k}(x)) \\
&\geq \sum_{j=1}^i \alpha_{j, i+1}(x) = t_{i+1}^-(x)
\end{aligned}$$

Consequently, for every $i \in \llbracket 1; l \rrbracket$, (t_1, t_2, \dots, t_i) is enabled at \emptyset . Finally, we can similarly show that $\sum_{j=1}^l (t_j^+(x) - t_j^-(x)) \geq \sum_{j=1}^l \alpha_{j,l+1}(x)$ and, as (v_i, v_{l+1}) is an edge of G for $i \leq l + 1$, then, by construction, $\sum_{j=1}^l \alpha_{j,l+1}(x) \geq \sum_{j=1}^l \mathcal{C}_j(x) = \mathcal{C}(x)$ for every place x . Thus, when firing T , the resulting state covers \mathcal{C} and T is a feasible solution of \mathcal{I} .

We now assume there exists such a feasible solution $T = (t_1, t_2, \dots, t_l)$ of \mathcal{I} and prove \mathcal{J} contains a clique of size $l + 1$. Let $\gamma_{i,j}$, for $i < j \in \llbracket 1; l \rrbracket$ be the states describing the tokens produced by t_i and consumed by t_j ; let $\gamma_{i,l+1}$ be the tokens produced by t_i , not consumed by any next transition and used to cover \mathcal{C} ; and finally let $\gamma'_{i,l+1}$ be the not consumed tokens of t_i^+ that are

not used to cover \mathcal{C} . Thus, we have firstly $t_i^+ = \left(\sum_{j=i+1}^{l+1} \gamma_{i,j} \right) + \gamma'_{i,l+1}$, secondly

$t_j^- = \sum_{i=1}^{j-1} \gamma_{i,j}$ and thirdly $\sum_{i=1}^l \gamma_{i,l+1} = \mathcal{C}$. For each set V_i with $i \leq l$, we select

the node v_i associated with t_i , the decomposition $\{\gamma_{j,i}, j \leq i - 1\}$ of t_i^- and the decomposition $\{\gamma_{i,i+1}, \dots, \gamma_{i,l}, \gamma_{i,l+1} + \gamma'_{i,l+1}\}$ of t_i^+ . We finally choose the node v_{l+1} of V_{l+1} associated with the decomposition $\{\gamma_{j,l+1}, j \leq l\}$ of \mathcal{C} . If $i < j \leq l$, then $\gamma_{i,j}$ is the j -th state of the decomposition of t_i^+ and the i -th state of the decomposition of t_j^- . By construction the edge (v_i, v_j) belongs to G . Moreover, if $i \leq l$, the last state $\gamma_{i,l+1} + \gamma'_{i,l+1}$ of the decomposition of t_i^+ covers the i -th state $\gamma_{i,l+1}$ of the decomposition of \mathcal{C} . By construction the edge (v_i, v_{l+1}) belongs to G . Consequently, $\{v_1, v_2, \dots, v_{l+1}\}$ is a clique of size of $l + 1$ in \mathcal{J} .

As a result, FS-MLWSP is W[1] with respect to l and π . By Theorem 11, it is W[1]-Complete. □