On Enumerating Minimal Siphons in Petri nets using CLP and SAT solvers: Theoretical and Practical Complexity

Faten Nabli, Thierry Martinez, François Fages, Sylvain Soliman

On Enumerating Minimal Siphons in Petri nets using CLP and SAT solvers: Theoretical and Practical Complexity⋆

Faten Nabli · Thierry Martinez · Franois Fages · Sylvain Soliman

Abstract Petri nets are a simple formalism for modeling concurrent computation. They are also an interesting tool for modeling and analysing biochemical reaction systems, bridging the gap between purely qualitative and quantitative models. Biological networks can indeed be complex, large, and with many unknown kinetic parameters, which makes the development of quantitative models difficult. In this paper, we focus on the Petri net representation of biochemical reactions and on two structural properties of Petri nets, siphons and traps, that bring us information about the persistence of some molecular species, independently of the kinetics. We first study the theoretical time complexity of minimal siphon decision problems in general Petri nets, and present three new complexity results: first, we show that the existence of a siphon of a given cardinality is NP-complete; second, we prove that deciding the Siphon-Trap property is co-NP-complete; third, we prove that deciding the existence of a minimal siphon containing a given set of places, deciding the existence of a siphon of a given cardinality and deciding the Siphon-Trap property can be done in linear time in Petri nets of bounded tree-width. Then, we present a Boolean model of siphons and traps, and two methods for enumerating all minimal siphons and traps of a Petri net, by using a SAT solver and a Constraint Logic Program (CLP) respectively. On a benchmark of 345 Petri nets of hundreds of places and transitions, extracted from biological models from the BioModels repository, as well as on a benchmark composed of 80 Petri nets from the Petriweb database of industrial processes, we show that both the SAT and CLP methods are overall faster by one or two orders of magnitude compared to the state-of-the-art algorithm from the Petri net community, and are in fact able to solve all the enumeration problems of our practical benchmarks.

⋆ This paper is an extended version of a paper presented at CP 2012 [37].

EPI Contraintes, Inria Paris-Rocquencourt, France
Tel.: +33 1 39 63 56 37
Fax: +33 1 39 63 54 69
E-mail: {Thierry.Martinez, Francois.Fages, Sylvain.Soliman}@inria.fr, faten7@gmail.com

We investigate why these programs perform so well in practice, and provide some elements of explanation related to our theoretical complexity results.

Keywords Petri nets · Siphons · Traps · Systems Biology · SAT · Constraint Logic Programming

1 Introduction

Petri nets were introduced in the 60's as a simple formalism for describing and analyzing information processing systems that are characterized as being concurrent, asynchronous, non-deterministic and possibly distributed [40]. The use of Petri nets for representing biochemical reaction systems, by mapping molecular species to places and reactions to transitions, was introduced quite late in [41], together with some Petri net concepts and tools for the analysis of metabolic networks. In particular, the traditional Petri net concepts of place-invariants (P-invariants) and transition-invariants (T-invariants) have important interpretations in biochemical networks: P-invariants express structural conservation laws between molecular species, which correspond to linear invariants and possible variable eliminations in systems of ordinary differential equations, while T-invariants revealed to be equivalent to the notion of extremal fluxes in metabolic networks [49], one of the main tools for analyzing and optimizing metabolic networks [30,15,47]. Constraint logic programs have been proposed to compute P-invariants and T-invariants in [44] and [38] respectively. Constraint programming methods have also been applied successfully to many other biology related problems. For instance by Devloo et al. to discover efficiently the steady-state of large gene regulation networks [16]. Fanchon and al. use constraints to infer ranges of parameter values from observations [20] and for analysing discrete genetic regulatory networks [8]. Bockmayr and Courtois [3] use Hybrid Concurrent Constraint to model a variety of biological phenomena, such as reaching thresholds, kinetics, gene interaction or biological pathways. In [30], Larhlimi and Bockmayr take advantage of the implicit representation that constraints bring, to describe the elementary flux cone of some metabolic pathways. Backofen et al. pioneered in [1] the use of constraints and symmetry breaking for predicting the structure of proteins, etc.

In this paper, we consider the Petri net concepts of siphons and traps. A siphon is a set of places that, once unmarked, remains unmarked. A trap is a set of places that, once marked, can never loose all its tokens. Thus, siphons and traps have opposed effects on the token distribution in a Petri net. These structural properties provide sufficient conditions for reachability (whether the system can produce a given protein or reach a given state from a given initial state) and liveness (deadlock freedom from a given initial state) properties in ordinary Petri nets. It is proved that in order for a net to have all its transitions live, it is necessary that each siphon remains marked. Otherwise (i.e., once a siphon is empty), transitions having their input places in a siphon cannot be live. One way to keep each siphon marked is to have a marked trap inside it.

2

In fact, this condition is necessary and sufficient for a free-choice net to be live [40],

We first study the theoretical time complexity of siphon extraction problems in general Petri nets. It has been shown in [46] and [48] that the problems of existence of a minimal siphon of a given cardinality (k-MINIMALSIPHON), or containing a given place (Q-MINIMALSIPHON), are NP-complete, and recently in [39] that the siphon-trap property (STP) is in co-NP. Here we provide new theoretical complexity results. First we show the NP-completeness of the existence problem of a siphon of a given cardinality (k-SIPHON) and thus the NP-hardness of MINIMALCARDINALITYSIPHON. Second, we prove that deciding the siphon-trap property is in fact co-NP-complete. Third, we prove that deciding the existence of a minimal siphon containing a given set of places, deciding the existence of a siphon of a given cardinality and deciding the Siphon-Trap property are of linear time complexity in Petri nets with bounded tree-width. These latter results follow from Courcelle's theorem.

Then we consider a simple Boolean model for defining siphons and traps, and two methods for enumerating the set of all minimal siphons and traps of a Petri net. The first method iterates the resolution of the Boolean model executed with a SAT solver, while the second proceeds by backtracking with a Constraint Logic Program over Booleans (CLP(B)).

We compare this Boolean constraint solving approach to the state-of-the-art algorithms from the Petri net community described in [11] for computing minimal sets of siphons and traps, which have already been shown to outperform Mixed Integer Linear Programs previously proposed in [35,9]. On a benchmark composed of 345 curated biological models of hundreds of species and reactions each, from the BioModels [1] repository [32], and of 80 Petri nets from the Petriweb [2] [22] database of industrial processes, we show that the SAT solver MiniSAT and CLP(B) solver GNU-Prolog are both faster by one or two orders of magnitude than the dedicated algorithms, and can in fact enumerate all solutions for all the intances of those benchmarks in a seconds. Finally, we question ourselves why these programs perform so well in practice, and provide some elements of explanation related to our theoretical complexity results.

2 Preliminaries on Petri Nets

2.1 Petri Nets

Definition 1 A *Petri net graph* N is a weighted bipartite directed graph $N = (P, T, W)$, where P is a non-empty finite set of vertices called *places*, T is a non-empty finite set of vertices called *transitions*, $P \cap T = \emptyset$, and $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ is a weight function attached to the arcs.

---

[1] http://www.biomodels.net/

[2] http://www.petriweb.org/

Intuitively, the weight of a $P \times T$ arc represents the minimum number of tokens (molecules) necessary to enable a transition and the weight on a $T \times P$ arc represents the quantity produced. By abuse of notation, the weight is equal to one if it is omitted. The weight of a $P \times T$ (resp. $T \times P$) arc equals zero if there exist no arcs from place to transition (resp. from transition to place).

Definition 2 A *marking* for a Petri net graph is a mapping $m : P \to \mathbb{N}$ which assigns a number of tokens to each place. We say that a place p is *marked* if $m(p) > 0$, otherwise it is said to be *unmarked*.

Definition 3 A *Petri net* is a 4-tuple (P, T, W, m_0) where (P, T, W) is a Petri net graph and m_0 is an *initial marking*.

Let $N = (P, T, W)$ be a Petri net graph.

Definition 4 The set of *predecessors* (resp. *successors*) of a transition $t \in T$ is the set of places $^\bullet t = \{p \in P \mid W(p, t) > 0\}$ (resp. $t^\bullet = \{p \in P \mid W(t, p) > 0\}$). Similarly, the set of predecessors (resp. successors) of a place $p \in P$ is the set of transitions $^\bullet p = \{t \in T \mid W(t, p) > 0\}$ (resp. $p^\bullet = \{t \in T \mid W(p, t) > 0\}$).

The set of predecessors (resp. successors) $^\bullet S$ (resp. $S^\bullet$) of a set of places S is the union of sets of predecessors (resp. successors) of each place $p \in S$: $^\bullet S = \bigcup_{p \in S} {}^\bullet p$ (resp. $S^\bullet = \bigcup_{p \in S} p^\bullet$).

The set of predecessors (resp. successors) $^\bullet Q$ (resp. $Q^\bullet$) of a set of transitions Q is the union of sets of predecessors (resp. successors) of each transition $t \in Q$: $^\bullet Q = \bigcup_{t \in Q} {}^\bullet t$ (resp. $Q^\bullet = \bigcup_{t \in Q} t^\bullet$).

Definition 5 N is *ordinary* if for all $p \in P$ and for all $t \in T$, $W(p, t) \leq 1$ and $W(t, p) \leq 1$.

Definition 6 A transition t is *enabled* at marking m when $\forall p \in {}^\bullet t : m(p) \geq W(p, t)$.

For every two markings $m, m' : P \to \mathbb{N}$ and every transition $t \in T$, there is a *transition step* $m \xrightarrow{t} m'$ if for all $p \in P$, $m(p) \geq W(p, t)$ and $m'(p) = m(p) - W(p, t) + W(t, p)$.

$m \xrightarrow{t} m'$ means that the transition t is enabled in m and its firing leads to m'. An enabled transition may or may not fire if there are other transitions enabled.

Example 1 The classical Petri net view of a reaction model is to associate biochemical *species* to *places* and biochemical *reactions* to *transitions*. The well-known system of Michaelis-Menten enzymatic reactions can be represented by the Petri net depicted in Figure 1. It consists of three reactions that take place in two discrete steps: the first involves the reversible formation of a complex (AE) between the enzyme (E) and its substrate (A), and the second step involves an irreversible transformation of the product (B) with release of the enzyme.
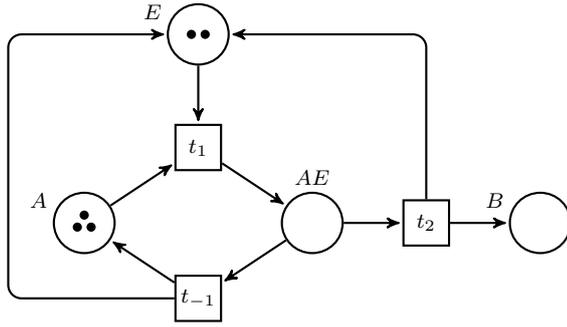
$$A + E \leftrightarrows AE \to B + E$$

Fig. 1: Petri net associated to the example of Michaelis-Menten of Example 1, displayed here with a marking enabling transition t_1.

Definition 7 Given m and m' two markings of N. A finite sequence of transitions $\sigma = (t_0 \ldots t_n)$ is a finite *firing sequence* of the Petri net if there exists a sequence of markings $m_1, \ldots, m_n$ for which $m \overset{t_0}{\to} m_1 \overset{t_1}{\to} \ldots \overset{t_{n-1}}{\to} m_n \overset{t_n}{\to} m'$. This is denoted by $m \overset{\sigma}{\to} m'$.

Definition 8 A marking m' is *reachable* from m if there exists a finite sequence σ such that $m \overset{\sigma}{\to} m'$.

Definition 9 Let $N = (P, T, W, m_0)$ be a Petri net.

 – A transition $t \in T$ is *dead* at marking m if it is not enabled in any marking m' reachable from m.
 – A marking m is *dead* if there is no transition enabled in m.
 – A Petri net is *deadlock free* (weakly live) if there is no reachable dead markings.

2.2 Siphons and Traps

Let $N = (P, T, W)$ be a Petri net graph.

Definition 10 A *trap* is a non-empty set of places $P' \subseteq P$ whose successors are also predecessors: $P'^\bullet \subseteq {}^\bullet P'$.

A *siphon* is a non-empty set of places $P' \subseteq P$ whose predecessors are also successors: ${}^\bullet P' \subseteq P'^\bullet$.

A siphon (resp. a trap) is *proper* if its predecessor set is strictly included in its successor set: ${}^\bullet P' \subsetneq P'^\bullet$ (resp. $P'^\bullet \subsetneq {}^\bullet P'$).

A siphon (resp. a trap) is *minimal* if it does not contain any other siphon (resp. trap).

It is worth remarking that a siphon in N is a trap in the dual Petri net graph, obtained by reversing the direction of all arcs in N. Note also that since

5

predecessors and successors of an union are the union of predecessors (resp. successors), the union of two siphons (resp. traps) is a siphon (resp. a trap).

The Petri net of Example 1 has two minimal siphons, $\{A, AE\}$ and $\{E, AE\}$. $\{E, AE\}$ is both a minimal siphon and a minimal trap since ${}^\bullet\{E, AE\} = \{E, AE\}^\bullet = \{t_1, t_{-1}, t_2\}$

Although siphons and traps are stable under union, it is worth noting that minimal siphons do not form a generating set of all siphons. A siphon is called a *basis* siphon if it cannot be represented as a union of other siphons [35]. Obviously, a minimal siphon is also a basis siphon, however, not all basis siphons are minimal. For instance, in Example 1, the generating set of siphons is formed by $\{A, AE\}$, $\{E, AE\}$, $\{A, AE, B\}$ and $\{E, AE, B\}$, but only $\{A, AE\}$ and $\{E, AE\}$ are minimal, the two others cannot be obtained by union of minimal siphons.

The following propositions show that traps and siphons provide a structural characterization of some particular dynamical properties on markings.

Proposition 1 *[40] For every subset $P' \subseteq P$ of places, P' is a trap if and only if for any marking $m \in \mathbb{N}^P$ with $m_p \geq 1$ for some place $p \in P'$, and any marking $m' \in \mathbb{N}^P$ such that $m \xrightarrow{\sigma} m'$ for some sequence σ of transitions, there exists a place $p' \in P'$ such that $m'_{p'} \geq 1$.*

Proposition 2 *[40] For every subset $P' \subseteq P$ of places, P' is a siphon if and only if for any marking $m \in \mathbb{N}^P$ with $m_p = 0$ for all $p \in P'$, and any marking $m' \in \mathbb{N}^P$ such that $m \xrightarrow{\sigma} m'$ for some sequence σ of transitions, we have $m'_{p'} = 0$ for all $p' \in P'$.*

2.3 Application to Deadlock Detection

One reason to consider minimal siphons is that they provide a sufficient condition for the non-existence of deadlocks. A deadlock occurs in a marked Petri net if no transition is enabled. It has been shown indeed that in a deadlocked (and marked) Petri net, all unmarked places form a siphon [6]. The siphon-based approach for deadlocks detection checks if the net contains a proper siphon that can become unmarked by some firing sequence. In parrticular the following STP property provides a sufficient condition for ordinary Petri nets to be deadlock free.

Definition 11 Given a Petri net (P, T, W, m_0), the *siphon-trap property (STP)* holds when every siphon contains a marked trap.

Theorem 1 ([7]) *An ordinary Petri net in which the STP holds is deadlock free.*
Proof. We just have to show that STP is preserved by transition firing and that a dead marking does not satisfy STP. The preservation follows from the fact that a marked trap remains marked after the firing of a transition. If a marking m is dead, then the set $S = \{p \in P \mid m(p) = 0\}$ is such that $S^\bullet = T$

since every transition should have an empty predecessor. Then $^{\bullet}S \subseteq S^{\bullet}$, and moreover S is non-empty since T is non-empty. Therefore S is a siphon and does not contain a marked trap. □

The relevance of siphons and traps for other liveness properties in systems biology are reported in [23].

2.4 Application to Systems Biology

One example of the relevance of traps and siphons in biology was given in [49] for the analysis of the production and accumulation of starch in potato tubers during growth, while starch is consumed after the tubers are deposited after the harvest. This can be seen by a purely structural analysis of the Petri net graph of the metabolic network, since starch and several of its precursors form traps in the reaction net during growth, while starch and possible intermediates of degradation form siphons after the harvest. A simplified version of this Petri net is depicted in Figure 2, where G_1 stands for glucose-1-phosphate, G_u is UDP-glucose, S is the starch, I stands for intermediary species and P_1 and P_2 represent external metabolites [45]. In this network, either the branch producing starch (t_3 and t_4) or the branch consuming it (t_5 and t_6) is operative, as it is shown in Figure 3 and Figure 4 respectively. This is realized by a switching mechanism in the gene regulatory network with synthesis of the corresponding enzymes. Two Petri nets are thus derived from this model: one Petri net where t_5 and t_6 are removed (in this Petri net, t_3 and t_4 are said to be operative) and one Petri net where t_3 and t_4 are removed.

It can be easily observed that the set $\{G_u, S\}$ is a trap when t_3 and t_4 are operative: once a token arrives in S, no transition can be fired and the token remains there independently of the evolution of the system. Dually, $\{S, I\}$ is a siphon when t_5 and t_6 are operative: once the last token is consumed from S and I, no transition can generate a new token in these places, so they remain empty.

Another interesting example of use of the concept of siphons and traps, also from [49], deals with the analysis of the role of the triosephosphate isomerase (TPI) in Trypanosoma brucei metabolism. Earlier, Helfert et al. [24] supposed that glycolysis could proceed without TPI, but unexpected evidence that all system fluxes (Pyruvate, Glycerol) decrease was found which lead the authors to build a kinetic model for explaining that phenomenon. However, a purely structural explanation for the necessary presence of TPI in glycolysis and glycerol production was provided in [49] simply by showing the existence of siphons and traps in the model.
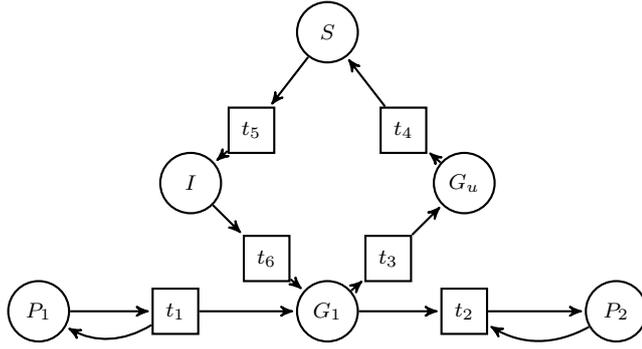
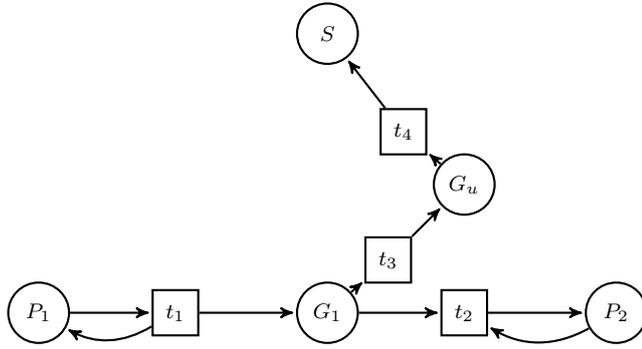Fig. 2: Petri net graph modeling the growth metabolism of the potato plant [49].

Fig. 3: Petri net graph modeling the growth metabolism of the potato plant [49] with the branch producing starch operative.

3 Theoretical Complexity of Siphon and Trap Properties

3.1 Preliminaries on SAT, FO and MSO

We shall prove some NP-completeness results using the following characterization of Boolean satisfiability. For a finite set of variables $V = \{x_1, \ldots x_m\}$, let $\neg V = \{\neg x_1, \ldots, \neg x_m\}$ denotes the set of negative literals constructed upon V. For a Boolean formula in conjunctive normal form $\phi = c_1 \wedge \cdots \wedge c_n$ over V, we have for all $1 \leq i \leq n$, $c_i = \ell_{i,1} \vee \cdots \vee \ell_{i,n_i}$, and for all $1 \leq j \leq n_i$, $\ell_{i,j} \in V \cup \neg V$. Let us write $C_\phi = \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ and $L_\phi = \{(i,j) \in \mathbb{N}^2 \mid 1 \leq i \leq n, 1 \leq j \leq n_i\}$.

Lemma 1 *A Boolean formula ϕ in conjunctive normal form is satisfiable if and only if there exists a subset $X \subseteq L_\phi$ such that*

  – for all $1 \leq i \leq n$, there exists $1 \leq j \leq n_i$ such that $(i,j) \in X$,
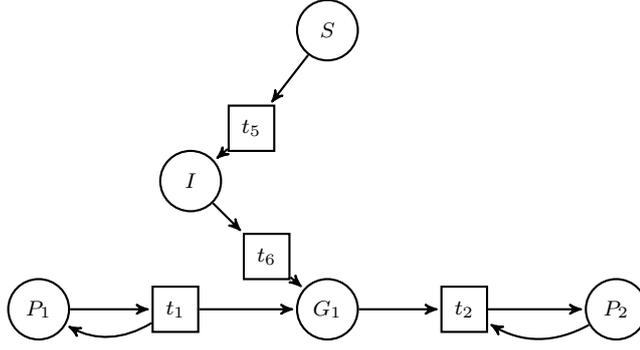
8

Fig. 4: Petri net graph modeling the growth metabolism of the potato plant [49] with the branch consuming starch operative.

 – for all $(i, j), (i', j') \in X$, $\ell_{i',j'} \neq \neg \ell_{i,j}$.

Proof. If $\mu : V \to \{0, 1\}$ satisfies ϕ, we pose for all $x \in V$, $\mu(\neg x) = 1 - \mu(x)$ and then it suffices to observe that $X = \{(i, j) \in L \mid \mu(\ell_{i,j}) = 1\}$ satisfies the conditions of the lemma. Conversely, given a subset $X \subseteq L_\phi$ satisfying these conditions, we pose $\mu : V \to \{0, 1\}$ such that for all $x \in V$, $\mu(x) = 1$ if there exists $(i, j) \in X$ such that $\ell_{i,j} = x$ and 0 otherwise. Then, we observe that μ satisfies ϕ. $\qquad\square$

We say that ϕ *is satisfied by* X if X is a subset of L_ϕ satisfying the conditions of Lemma 1.

We shall also use the language of first-order logic (FO) to express properties over finite set and graph structures assumed to be fixed. For each finite set S, we assume a unary predicate $S(x)$, also written $x \in S$ by abuse of notation, which tests membership. Non-emptyness, $S \neq \emptyset$, is expressed by the FO formula $\exists x (x \in S)$, and set inclusion, $S \subseteq S'$, by $\forall x (x \in S \to x \in S')$. Similarly for each Petri net graph, we assume two unary (set) predicates, place(x) and transition(x), which distinguish between places and transitions, and a binary predicate, edge(x, y), which tests incidence. This leads to the following characterization of siphons and traps:

Definition 12 The siphon and trap properties can be defined in FO by the following schemas of formulae:

$$\text{SIPHON}(S) : \; S \neq \emptyset \wedge \forall p \in S \; \text{place}(p)$$
$$\wedge \, \forall t (\exists p \in S \; \text{edge}(t, p) \to \exists p \in S \; \text{edge}(p, t))$$
$$\text{TRAP}(S) : \; S \neq \emptyset \wedge \forall p \in S \; \text{place}(p)$$
$$\wedge \, \forall t (\exists p \in S \; \text{edge}(p, t) \to \exists p \in S \; \text{edge}(t, p))$$

We shall also provide a series of linear time complexity results by showing that some problems can be expressed in monadic second-order logic (MSO) over

9

finite graph structures, in order to use Courcelle's theorem. MSO extends FO by the addition of second-order quantifiers over predicates, also noted $\exists$ and $\forall$, with the restriction to apply to unary predicates only, i.e. sets. An example of MSO formula is given in the proof of Theorem 4 in the next section.

Definition 13 ([42]) A *tree decomposition* of a non-oriented graph $G = (V, E)$ is a pair (X, T) where $T = (I, A)$ is a tree and $X = (X_i)_{i \in I}$ is a family of subsets of V such that

- $\bigcup_{i \in I} X_i = V$,
- for all $\{v, v'\} \in E$, there exists $i \in I$ such that $\{v, v'\} \subseteq X_i$,
- for all $i, j, k \in I$, if j lies on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The *tree-width of a tree decomposition* is $\max_{i \in I} |X_i| - 1$. The *tree-width* $\mathrm{tw}(G)$ *of* G is the minimum tree-width taken over all possible tree decompositions of G.

The tree-width of an ordinary Petri net graph $N = (P, T, W)$ follows that definition given for graphs, by reading the Petri net as the non-oriented graph $G = (P \cup T, E)$ where $E = \{\{p, t\} \subseteq P \cup T \mid W(p, t) > 0 \text{ or } W(t, p) > 0\}$.

Example 2 Let us consider the Petri net graph depicted in Figure 5. The non-oriented graph associate to this Petri net is depicted in Figure 6. Two possible tree decompositions of this non-oriented graph are shown in Figure 7. In both decompositions, each graph edge connects two vertices which belong to the same tree node. Graph vertices are adjacent only when the corresponding sub-trees intersect. In the first tree decomposition (Figure 7 on the left), each tree node contains at most four vertices, hence the width of this decomposition is three. In the second tree (Figure 7 on the right), each tree node contains at most three vertices, hence the width of this decomposition is two, which is the optimal tree-width over all possible tree decompositions. Hence, the tree-width of the Petri net graph depicted in Figure 5 equals two.
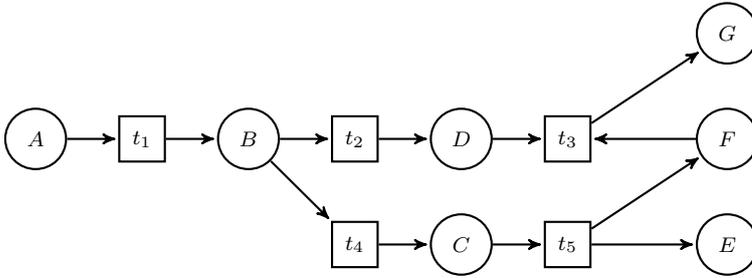

Fig. 5: Example of a Petri net with its associated non-oriented graph depicted in Figure 6 and two tree decompositions given in Figure 7.

Courcelle's theorem states that every graph property definable in MSO can be decided in linear time on graphs of bounded tree-width.
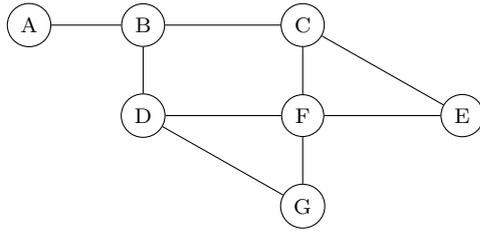
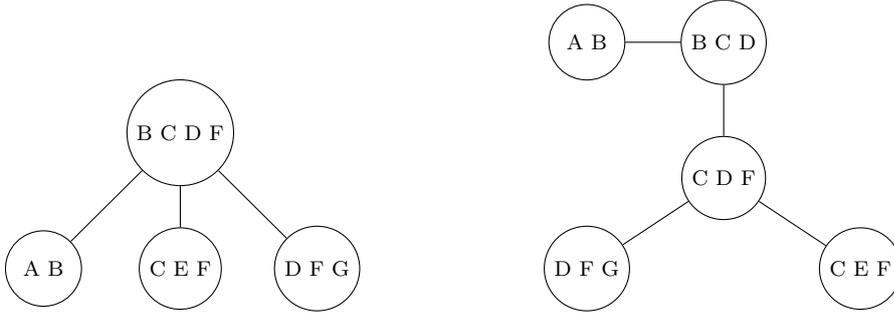Fig. 6: Non-oriented graph associated to the Petri net of Figure5

Fig. 7: Two possible tree decompositions of the graph shown in Figure 6

Theorem 2 (Courcelle [12]) *For a given formula P in monadic second-order logic on the structure of graphs and for a given positive integer k, there exists a linear time algorithm that given a finite graph G of tree-width at most k decides whether G satisfies P.*

Given a tree-automaton constructed from the formula, and a tree constructed from the graph decomposition that minimizes the tree-width, deciding if the automaton accepts the tree can be done in linear time. Courcelle's theorem shows the existence of such automata for MSO properties. However, the automata can be of hyper-exponential sizes. This makes the linear algorithm usually unusable in practice. There is some recent work in [13, 28] on fly-automata and game-theoretic methods, aiming at making this result practical for certain properties of graphs with bounded clique-width. The applicability of these new approaches to our problem is however beyond the scope of this paper.

3.2 The Q-MinimalSiphon Problem

Definition 14 The problem FindMinimal is the following problem: "*given a Petri net graph $N = (P, T, W)$ and a subset of places $Q \subseteq P$, find a minimal siphon S in N such that $Q \subseteq S$ if there is any, or fail.*"

Theorem 3 ([46]) *The problem FindMinimal is polynomial.*

11

Definition 15 Q-MINIMALSIPHON is the following decision problem: *"given a Petri net graph $N = (P, T, W)$ and a subset of places $Q \subseteq P$, does there exist a minimal siphon S in N such that $Q \subseteq S$?"*.

Theorem 4 Q-MINIMALSIPHON *is decidable in linear time for any class of Petri net graphs with bounded tree-width.*

Proof. By Definition 12, siphons and traps are expressible in FO, and thus in MSO. We just have to verify that Q-MINIMALSIPHON can be expressed in MSO as follows:

$$Q\text{-MINIMALSIPHON} : \exists S(\text{SIPHON}(S) \wedge Q \subseteq S$$
$$\wedge \, \forall S'(\text{SIPHON}(S') \wedge S' \subseteq S \rightarrow S \subseteq S'))$$

The linear time complexity then follows from Courcelle's theorem. □

In the general case, Q-MINIMALSIPHON has been shown NP-complete in [48]. We find it useful here to provide a simpler proof of this result, by showing the NP-hardness of the following equivalent problem.

Definition 16 For a given Petri net graph $N = (P, T, W)$ and $Q \subseteq P$, a *Q-hitting siphon* is a siphon $S \subseteq P$ of N such that for every siphon $S' \subseteq S$ of N, $Q \subseteq S'$.

Definition 17 Q-HITTINGSIPHON is the following decision problem: *"given a Petri net graph $N = (P, T, W)$ and a subset of places $Q \subseteq P$, does there exist a Q-hitting siphon in N?"*.

Proposition 3 *Given a Petri net graph $N = (P, T, W)$ and a subset of places $Q \subseteq P$, there exists a minimal siphon containing Q in N if and only if there exists a Q-hitting siphon in N.*
Proof. If S is a minimal siphon containing Q, then S is a Q-hitting siphon. Conversely, if S is a Q-hitting siphon, then there exists a minimal siphon $S' \subseteq S$ and since S is Q-hitting, then $Q \subseteq S'$. □

Theorem 5 Q-MINIMALSIPHON *is NP-complete.*
Proof. Q-MINIMALSIPHON is in NP since FINDMINIMAL is polynomial. We just have to show that Q-HITTINGSIPHON is NP-hard.

Let us assume a SAT instance described by a set V of variables and a Boolean formula ϕ over V in conjunctive normal form. Let $N = (P, T, W)$ be the ordinary Petri net graph, depicted in Figure 8, where

- $P = L_\phi \cup \{q\}$ and $T = V \cup \neg V \cup C_\phi$ where $q \notin L_\phi \cup T$,
- W satisfies

$$\begin{array}{ll} {}^\bullet(i,j) = \{\neg \ell_{i,j}\} & (i,j)^\bullet = \{i, \ell_{i,j}\} \quad \text{for all } (i,j) \in L_\phi \\ {}^\bullet q = C_\phi & q^\bullet = V \cup \neg V \end{array}$$
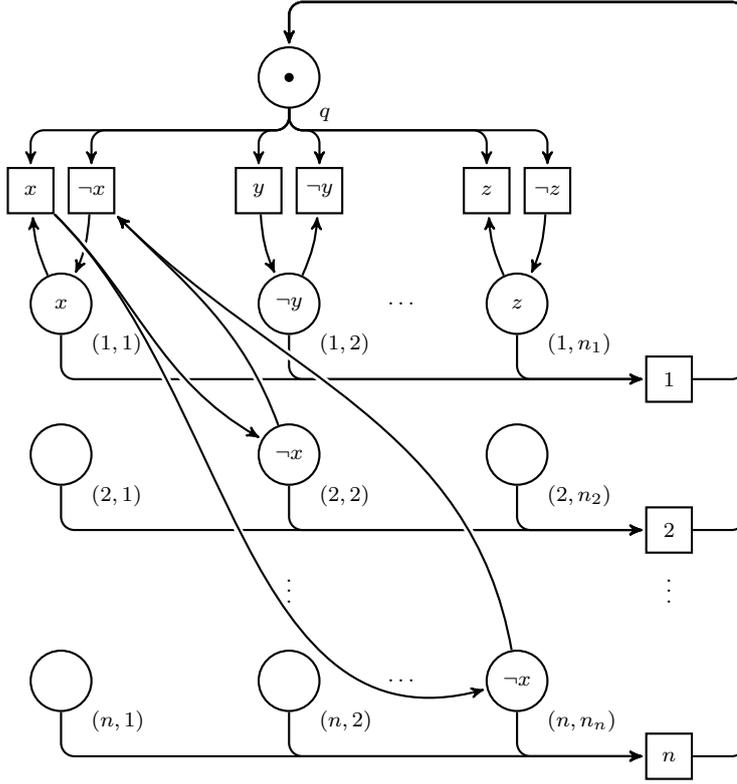
Fig. 8: Petri net used in the reduction of SAT to Q-HITTINGSIPHON in Theorem 5. In this Petri net, a siphon S contains the place q on top of the figure if and only if S contains at least one literal occurrence per clause. Moreover, S is $\{q\}$-hitting if and only if S does not contain a pair of contradictory literals.

Lemma 2 ϕ *is satisfiable if and only if there exists a $\{q\}$-hitting siphon in N, i.e., a subset $S \subseteq P$ such that $S \neq \emptyset$ and $^\bullet S \subseteq S^\bullet$ and for all $S' \subseteq S$, if $S' \neq \emptyset$ and $^\bullet S' \subseteq S'^\bullet$, then $q \in S'$.*

Proof. Suppose that ϕ is satisfied by $X \subseteq L_\phi$. We verify that $X \cup \{q\}$ is a $\{q\}$-hitting siphon. Indeed, $^\bullet X \subseteq V \cup \neg V = q^\bullet$ and, by hypotheses on X, $^\bullet q = C_\phi \subseteq X^\bullet$, therefore $^\bullet(X \cup \{q\}) \subseteq (X \cup \{q\})^\bullet$. For any $S' \subseteq S$ such that $S' \neq \emptyset$ and $^\bullet S' \subseteq S'^\bullet$, suppose that $q \notin S'$, then since $S' \neq \emptyset$, there exists $(i,j) \in S'$ and $^\bullet(i,j) = \{\neg \ell_{i,j}\} \subseteq S'^\bullet$, therefore there exists $(i',j') \in S'$ such that $\neg \ell_{i,j} \in (i',j')^\bullet$, therefore $\ell_{i',j'} = \neg \ell_{i,j}$, which contradicts that $S' \subseteq X$; thus, $q \in S'$. That proves that $X \cup \{q\}$ is a $\{q\}$-hitting siphon.

Conversely, suppose that S is a $\{q\}$-hitting siphon. We verify that ϕ is satisfied by $X = S \cap L_\phi$. Indeed, for all $1 \leq i \leq n$, $i \in {}^\bullet q$, then $i \in X^\bullet$, therefore there exists $1 \leq j \leq n_i$ such that $(i,j) \in X$. Suppose that there exist $(i,j),(i',j') \in X$ such that $\ell_{i',j'} = \neg \ell_{i,j}$, then $S' = \{(i,j),(i',j')\}$ would be

13

such that $S' \neq \emptyset$ and $^\bullet S' \subseteq S'^\bullet$, despite $q \notin S'$. Therefore, for all $(i,j),(i',j') \in X$, $\ell_{i',j'} \neq \neg \ell_{i,j}$. Thus, ϕ is satisfiable. $\qquad \square$

The theorem is then an immediate consequence of the lemma. $\qquad \square$

3.3 The k-Siphon Problem

Definition 18 The problem k-Siphon is the following decision problem: "*given a Petri net graph $N = (P,T,W)$ and a positive integer k, does there exist a siphon S in N of cardinality k?*".

The linear time complexity immediately follows from Courcelle's theorem.

Theorem 6 k-Siphon *is decidable in linear time with respect to the size of the Petri net for any class of Petri net graphs with a bounded tree-width.*
Proof. Given a Petri net graph $N = (P,T,W)$ and a positive integer k, there exists a siphon S in N of cardinality k if and only if $\exists S(\text{Siphon}(S) \wedge \text{card}_k(S))$ is satisfied, where the formula $\text{card}_k(S)$

$$\text{card}_k(S) : \exists x_1 \dots x_k \left(\bigwedge_{1 \leq i < j \leq k} x_i \neq x_j \wedge \forall x \left(x \in S \rightarrow \bigvee_{1 \leq i \leq k} x = x_i \right) \right)$$

checks that the cardinality of S is k. $\qquad \square$

We prove the NP-completeness of k-Siphon by polynomial reduction from the set covering problem, one of Karp's original NP-complete problems [26]. Let us recall that the problem k-SetCovering is the following decision problem: "*given a finite set $\mathcal{U}$ (the universe), a subset $\mathcal{S}$ of $\mathfrak{P}(\mathcal{U})$ and an integer k, does there exist a subset $S \subseteq \mathcal{S}$ of cardinality k such that $\mathcal{U} = \bigcup S$?*".

Theorem 7 k-Siphon *is NP-complete.*
Proof. k-Siphon is in NP since checking that a given set of places is a siphon of cardinality k is polynomial. NP-hardness comes by polynomial reduction from k-SetCovering: given a finite set $\mathcal{U}$ (the universe), a subset $\mathcal{S}$ of $\mathfrak{P}(\mathcal{U})$ and an integer k, let $N = (P,T,W)$ be the ordinary Petri net graph such that $P = \mathcal{S}$, $T = \mathcal{U}$ and for all $t \in T$, $t^\bullet = P$ and $^\bullet t = \{S \in \mathcal{S} \mid t \in S\}$. Then for every subset $S \subseteq P$, S is a siphon if and only if $\mathcal{U} = \bigcup S$. $\qquad \square$

This shows that the optimization problem MinimalCardinalitySiphon is NP-hard.

14

3.4 The STP Problem

Definition 19 The Siphon-Trap Property STP is the following decision problem: "*given a marked Petri net $N = (P, T, W, m_0)$, does every siphon in N include a trap that contains a marked place?*".

Theorem 8 STP *is decidable in linear time for any class of Petri nets with bounded tree-width.*

Proof. Here again, it suffices to remark that STP is expressible in MSO, with a unary predicate marked(p) for distinguishing marked places, and the following MSO formula:

$$\text{STP} : \forall S(\text{SIPHON}(S) \to \exists T(\text{TRAP}(T) \wedge T \subseteq S \wedge \exists p(\text{marked}(p) \wedge p \in T)))$$

The linear time complexity immediately follows from Courcelle's theorem. $\square$

In the general case, STP has been shown to be in co-NP in [39], by reducing $\neg$STP to SAT. Indeed, $\neg$STP expresses the existence of a siphon S such that every trap included in S does not intersect M. The encoding in SAT focuses on the maximal trap included in S (the union of all the traps included in S), which is computed by removing iteratively places in S that cannot belong to a trap.

In fact, we can prove

Theorem 9 STP *is co-NP-complete.*
Proof. Since STP is in co-NP [39], it suffices to show that $\neg$STP is NP-hard.

Let us assume a SAT instance described with a set V of variables and a Boolean formula ϕ over V in conjunctive normal form. Let $N = (P, T, W, m_0)$ be the ordinary Petri net, depicted in Figure 9, where

- $P = L_\phi \cup \{0\} \times V$ and $T = V \cup \neg V \cup C_\phi \cup \{t\}$ where $t \notin P \cup V \cup \neg V \cup C_\phi$,
- W satisfies

$$\begin{aligned} {}^\bullet(i, j) &= C_\phi \cup \{\ell_{i,j}\} & (i, j)^\bullet &= \{i, t\} & \text{for all } (i, j) \in L_\phi \\ {}^\bullet(0, x) &= \{t\} & (0, x)^\bullet &= \{x, \neg x\} & \text{for all } x \in V \end{aligned}$$

- $m_0 = \mathbb{1}_{\{0\} \times V}$.

Note that the set $\{0\} \times V$ is introduced in places to ensure that $P \cap T = \emptyset$.

Lemma 3 ϕ *is satisfiable if and only if* $(N, \{0\} \times V)$ *satisfies* $\neg$STP, *i.e., there exists a subset* $S \subseteq P$ *such that* $S \neq \emptyset$ *and* ${}^\bullet S \subseteq S^\bullet$ *and for all* $T \subseteq S$, *if* $T \neq \emptyset$ *and* $T^\bullet \subseteq {}^\bullet T$, *then* $T \cap \{0\} \times V = \emptyset$.
Proof. Suppose that ϕ is satisfied by a set $X \subseteq L_\phi$. We verify that $X \cup \{0\} \times V$ is a siphon and that it does not contain any trap intersecting $\{0\} \times V$. Indeed, by hypotheses on X, $C_\phi \subseteq X^\bullet$, and ${}^\bullet(X \cup \{0\} \times V) \subseteq C_\phi \cup V \cup \neg V \cup \{t\} \subseteq (X \cup \{0\} \times V)^\bullet$. For any $T \subseteq X \cup \{0\} \times V$ such that $T \neq \emptyset$ and $T^\bullet \subseteq {}^\bullet T$, suppose that there exists $x \in T \cap \{0\} \times V$, then $\{x, \neg x\} \subseteq T^\bullet$, therefore $\{x, \neg x\} \subseteq {}^\bullet T$ and there exist $(i, j), (i', j') \in T \cap X$ such that $\ell_{i',j'} = \neg\ell_{i,j}$,
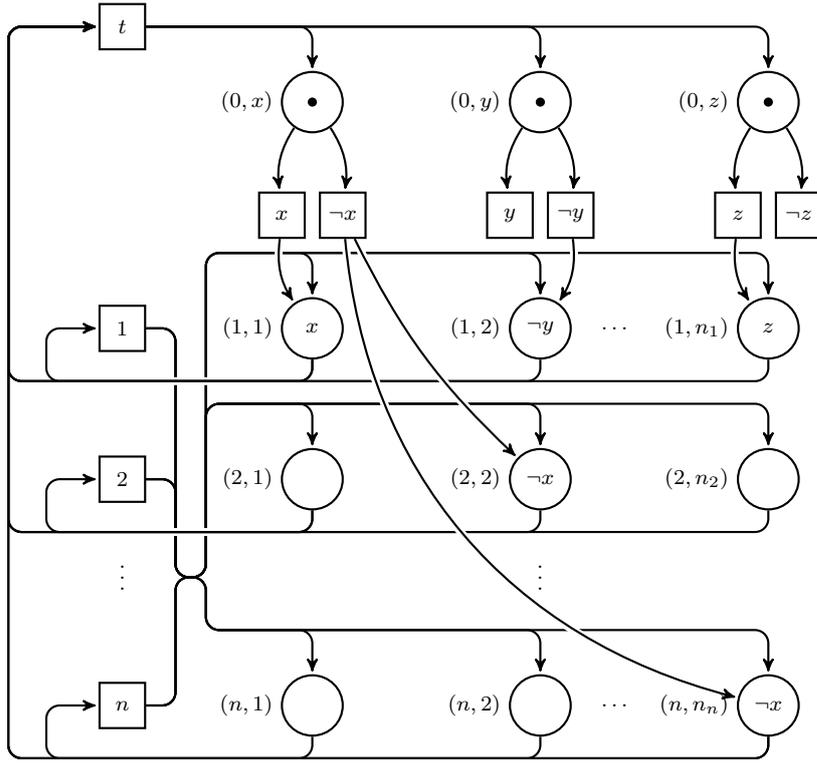
15

Fig. 9: Petri net used in Theorem 9 for the reduction of SAT to STP. In this Petri net set of places S is a siphon if and only if (1) S contains at least one literal occurrence per clause and (2) S contains all the places $(0,x)$ such that the variable x occurs in these literal occurrences; moreover, a marked place $(0,x)$ belongs to a trap included in S if and only if S contains the pair of contradictory literal occurrences x and $\neg x$.

that would contradict the hypotheses on X. Therefore $(N, \{0\} \times V)$ satisfies $\neg$STP.

Conversely, suppose that there exists a subset $S \subseteq P$ such that $S \neq \emptyset$ and $^\bullet S \subseteq S^\bullet$ and for all $T \subseteq S$, if $T \neq \emptyset$ and $T^\bullet \subseteq {}^\bullet T$, then $T \cap \{0\} \times V = \emptyset$. We verify that ϕ is satisfied by $X = S \cap L_\phi$. We have $X \neq \emptyset$ because otherwise there would exist $(0,x) \in S \cap \{0\} \times V$, then $t \in {}^\bullet S$, therefore $t \in S^\bullet$ and there would exist $(i,j) \in S \cap L_\phi$, which contradicts $X = \emptyset$. Thus, $C_\phi \subseteq {}^\bullet S \subseteq S^\bullet$ and for all $1 \leq i \leq n, i \in S^\bullet$ so there exists $1 \leq j \leq n_i$ such that $(i,j) \in X$. Suppose that there exist $(i,j),(i',j') \in X$ such that $\ell_{i',j'} = \neg \ell_{i,j}$. Then, there exists $(0,x) \in S \cap \{0\} \times V$ such that $(0,x)^\bullet = \{\ell_{i,j}, \ell_{i',j'}\}$. Then $T = \{(0,x),(i,j),(i',j')\}$ will be such that $T^\bullet = \{\ell_{i,j}, \ell_{i',j'}, i, i', t\} \subseteq C_\phi \cup \{\ell_{i,j}, \ell_{i',j'}, t\} = {}^\bullet T$, but $T \cap \{0\} \times V \neq \emptyset$. Therefore, for all $(i,j),(i',j') \in X$, $\ell_{i',j'} \neq \neg \ell_{i,j}$. $\qquad\square$

The theorem is now an immediate consequence of the lemma. □

4 Boolean Model for Minimal Siphons

The definition of siphons in FO given in Definition 12 directly leads to the following Boolean Constraint Satisfaction Problem (CSP):

Definition 20 Given a Petri net graph $N = (P, T, W)$, the CSP SIPHON(N) is the triple (V, D, C) where

- $V = P$, i.e. a variable is introduced for each place,
- $D(p) = \{0, 1\}$ for all $p \in V$, i.e. the variables are Boolean,
- $\bigvee_{p \in S} p = 1$, i.e. a siphon is not empty,
- $C = \{(p = 1 \rightarrow \bigvee_{p' \in {}^{\bullet}t} p' = 1) \mid p \in P, t \in {}^{\bullet}p\}$.

Proposition 4 *For every Petri net graph $N = (P, T, W)$, the CSP* SIPHON(N) *is satisfied by a valuation ν if and only if $\{p \in P \mid \nu(p) = 1\}$ is a siphon.*
Proof. It suffices to check that for every non-empty set of places S, we have $\forall p, \forall t \in {}^{\bullet}p, p \in S \rightarrow \bigvee_{p' \in {}^{\bullet}t} p' \in S$ if and only if ${}^{\bullet}S \subseteq S^{\bullet}$. □

The encoding in a SAT solver is short and direct. For each transition t in the set of predecessors of a place p, a clause C is added to the satisfiability problem. C is formed by a negated boolean variable associated to p or-ed with boolean variables in the set of predecessors of t. To avoid the trivial case of the empty siphon, one clause is added.

The set of all minimal siphons (*w.r.t.* set inclusion) can be enumerated in the set inclusion order, by restarting search each time a siphon S is found, with the additional constraint $\bigvee_{p \in S} p = 0$, for disallowing any superset of that siphon.

For enumerating siphons in set inclusion order, we compared two techniques: one by labeling an auxiliary cardinality variable in increasing order ([36]), one by labeling directly the Boolean variables with increasing value selection (to test first the absence, then the presence of a place in the candidate solution). The second technique has revealed to be much more efficient. The correctness of this technique comes from the fact that if a solution S' is found after a solution S, then the two paths in the search tree leading to these solutions have a least common ancestor node: this node corresponds to the labeling of a place p. By construction, p belongs to S' and not to S, thus $S' \nsubseteq S$.

We shall also consider a variant of the above CSP where the constraints $p = 1 \rightarrow \bigvee_{p' \in {}^{\bullet}t} p' = 1$ are decomposed by introducing an intermediary Boolean variable for each transition. For every Petri net graph $N = (P, T, W)$, the CSP SIPHON'(N) $= (V, D, C)$ is defined as follows.

- $V = P \cup T$, i.e. one variable is introduced for each place and each transition,

- $D(x) = \{0, 1\}$ for all $x \in V$,
- $\bigvee_{p \in S} p = 1$, i.e. a siphon is not empty,
- $C = \{(p = 1 \rightarrow t = 1) \mid p \in P, t \in {}^{\bullet}p\} \cup \{(t = 0 \rightarrow p = 0) \mid t \in T, p \in {}^{\bullet}t\}$.

It is immediate that if SIPHON'(N) is satisfied by a valuation ν, then SIPHON(N) is satisfied by $\nu|_P$, and conversely that if SIPHON(N) is satisfied by ν, then ν can be extended to a valuation satisfying SIPHON'(N).

This variant SIPHON'(N) enjoys the following property.

Proposition 5 *For every Petri net graph $N = (P, T, W)$, the CSP* SIPHON'(N) *has the same tree-width as N.*

Proof. It suffices to notice that the primal graph of SIPHON'(N) is isomorphic to the bipartite graph induced by N. $\qquad\qquad\square$

5 Implementations with SAT and CLP Solvers

This section describes two implementations of the above model with appropriate search strategy, one using an iterated SAT procedure and the other based on Constraint Logic Programming with Boolean constraints. The source code of our implementation is available on the web[3].

5.1 Iterative SAT Algorithm

The Boolean model can be directly interpreted using a SAT solver to check the existence of a siphon or trap. However, the enumeration of all minimal siphons requires a particular strategy for *iterating* the use of the SAT solver with new constraints ensuring the *enumeration* of the set of all minimal siphons.

We use sat4j[4], an efficient library of SAT solvers in Java for both satisfaction and optimization. It includes an implementation of the MiniSAT algorithm and implements the value selection strategy mentioned above: for each variable, the value 0 is tried before the value 1, for the enumeration of all solutions.

Example 3 Let us consider the Petri net depicting the enzymatic reaction of Example 1. In the first iteration, the problem amounts to solve the following encoding of Horn-dual clauses: $(\neg A \vee AE) \wedge (\neg AE \vee E \vee A) \wedge (\neg E \vee AE) \wedge (\neg E \vee AE) \wedge (\neg B \vee AE)$ The problem is satisfied with the values: $E = B = 0$ and $A = AE = 1$, which means that $\{A, AE\}$ is a minimal siphon. In the second iteration, the clause $\neg A \vee \neg AE$ is added to ensure minimality, and the problem is satisfied with $A = B = 0$ and $E = AE = 1$ meaning that $\{E, AE\}$ is also a minimal siphon. A new clause is added stating that either E or AE

18

does not belong to the siphon and no more variable assignment can satisfy the problem.

Therefore, this model contains two minimal siphons: $\{A, AE\}$ and $\{E, AE\}$. The enzyme E is a catalyst protein for the transformation of the substrate E in a product B. Such a catalyst increases the rate of the reaction but is conserved in the reaction.

5.2 Backtrack Replay CLP(B) Algorithm

The search for siphons can also be implemented with a Constraint Logic Program with Boolean constraints (CLP(B)). We use GNU-Prolog [5] [17] for its efficient low-level implementation of Boolean constraint propagators, indexicals and search heuristics.

The enumeration strategy for enumerating all minimal siphons is indeed a variation of the *branch-and-bound* algorithm, where the search is restarted to find a non-superset siphon each time a new siphon is found. We have tried two versions of the branch-and-bound algorithm: with restart from scratch and by backtracking. In the branch-and-bound with restart method, it is essential to choose a variable selection strategy which ensures diversity. Indeed, an enumeration method with a fixed variable order accumulates failures by always trying to enumerate the same sets first and these failures are only lately pruned by the non-superset constraints. As a consequence, the developed search tree gets more and more dense after each iteration since the previous forbidden sets are repeatedly tried again. It is worth nboting that this phenomenon does not exist in SAT solvers thanks to no-good recording. In CLP, random variable order selection strategy provides a good diversity and performs much better than any uniform heuristics.

However, *branch-and-bound by backtracking* gives better performance when care is taken for posting the non-superset constraint only once, since reposting it at each backtrack step proved to be inefficient. We have implemented a backtrack replay strategy, i.e. a customized branch and bound procedure where the search is performed as follows:

1. each time a siphon is found, the path leading to this solution is memorized,
2. then the search is fully backtracked to the root in order to add to the model the new non-superset constraint,
3. and then the memorized path is rolled back and replayed to continue the search at the point it was stopped.

Rolling back the memorized path is conceptually equivalent to adding a lexicographic constraint that imposes to be greater than the last siphon found. However the performance of this constraint is not as good as an explicit roll-back, because of lack of propagation.

The backtracking procedure can use any variable selection strategy, regardless of the diversity, since it ensures by construction that the already explored
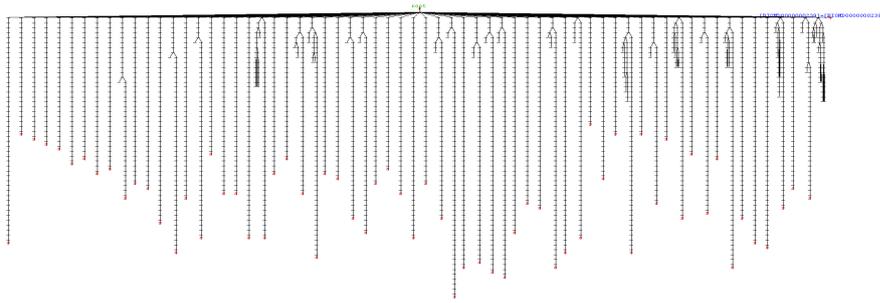
---

[5] http://www.gprolog.org/

Fig. 10: Search tree developed with the backtrack replay strategy for enumerating the 64 minimal siphons of the model 239 of BioModels which contains 51 species and 72 reactions. The branches ending with a cross are solutions. It is remarkable that very few backtracks are necessary to enumerate all solutions using the backtrack replay strategy.

sets will not be explored again. In our experiments, the most-constrained variable selection strategy is the one that provided the best performance and is therefore the one that was selected.

Figure 10 (generated with CLPGUI[6] [19]) depicts the search tree that is developed for enumerating the 64 minimal siphons of a biological model of 51 species and 72 reactions. Each sub-tree immediately connected to the root corresponds to the replay of the path with a minimality constraint added. The small number of backtrack points shows the remarkable efficiency of the backtrack replay strategy combined with a simple Boolean constraint propagator.

6 Evaluation

In the literature, many algorithms have been proposed to compute minimal siphons and traps of Petri nets. Since a siphon in a Petri net N is a trap of the dual net N', it is enough to focus on siphons, the traps are obtained by duality. Some algorithms are based on linear programming [35,9], Horn clause satisfaction [27,33] or algebraic approaches [31]. More recent state-of-the-art methods are presented in [10,11] and show the better performance of the dedicated algorithm of [11]. This algorithm uses a recursive problem partitioning procedure to reduce the original search problem to multiple simpler search sub-problems. Each sub-problem has specific additional place constraints with respect to the original problem. This algorithm can be applied to enumerate minimal siphons, place-minimal siphons, or even siphons that are minimal with respect to a given subset of places. In this section we compare our results to that dedicated algorithm.

---

[6]   http://contraintes.inria.fr/∼fages/CLPGUI

6.1 BioModels Benchmark

The BioModels [32] database is a repository of peer-reviewed, published, computational models of biological processes. These models are written in the Systems Biology Markup Language (SBML) exchange format and are mainly composed of reaction rules, with or without kinetics, and events. This resource allows biologists to store, search and retrieve models referenced in publications.

We consider the curated part of the repository BioModels version February 2013. Among these 404 curated models, 59 models do not hold any reaction (but events only), we thus consider the 345 reaction models from which a Petri net graph can be extracted from the structure of the reactions.

In addition, we consider the following complex biochemical models:

– a model of E2F/Rb signaling from [4] which contains 408 molecular species and 534 reactions, and will be shown to contain 74 minimal siphons,
– Kohn's map of the mammalian cell cycle control [29,5], which contains 509 species and 775 reactions, and will be shown to contain 80 minimal siphons.

6.2 Petriweb Benchmark

The database Petriweb [7] [22] is an attempt to make available a significant body of Petri nets examples in a public repository. They are written in the Petri Net Markup Language (PNML), an emerging standard format supported by many tools PetriWeb supports a restricted form of PNML, including flat, uncoloured nets, plus limited support for hierarchy The repository can be browsed with a web browser, and individual nets can be retrieved and uploaded in PNML.

The repository contains 80 Petri nets given with some interesting properties. The properties are defined by the repository administrator. The properties are intended to be checked automatically by external analysis tools.

6.3 Computational Results and Comparisons

In this section we compare the SAT method, using the MiniSAT solver implementation included in the SAT4J library, the CLP method, using GNU-Prolog, both described in Section 5, and the state-of-the-art dedicated algorithm of [11].

Table 1 presents the CPU times in milliseconds for enumerating the sets of all minimal siphons of the Petri nets in our benchmark in Petriweb and BioModels (except Model 175 as explained below). The CPU times have been obtained on a PC with an intel Core processor 2.20 GHz and 8 GB of memory. For each benchmark, we provide the total number of models, the minimal, maximal and average numbers of siphons, and the total computation time in milliseconds for enumerating all of them.

---

[7] http://www.petriweb.org/

Benchmark	# model	# siphons min-max (avg.)	siphons size min-max (avg.)	total time (ms)		
				dedicated algorithm	MiniSAT	GNU Prolog
BioModels	345	0-64 (4.21)	1-413 (3.10)	19734	611	195
Petriweb	80	0-11 (2.85)	0-7 (2.03)	2325	156	6

Table 1: Computation time in milliseconds on the BioModels and Petriweb benchmarks.

Quite surprisingly, on all these practical instances, both MiniSAT and GNU-Prolog solve the minimal siphon enumeration problem, in less than one millisecond in average, with a slightly better average performance for the CLP(B) program over the SAT solver. Furthermore, MiniSAT and GNU-Prolog outperform the dedicated algorithm by one or two orders of magnitude.

However, one particular model, number 175 in BioModels, has very high computation time and was excluded from Table 1. Table 2 presents the performance figures obtained on this model and on three other hardest instances for which we also provide the number of places and transitions. Even if the model is quite large, e.g. for Kohn's map of the cell cycle control with 509 species and 775 reactions, the computation time for enumerating its 81 minimal siphons is astonishingly short: one millisecond only. On these hard instances, the SAT solver is faster than the CLP(B) program by one or two orders of magnitude, and is the only algorithm to solve the problem for model 175, in 137 seconds.

model	# siphons	# places	# transitions	dedicated algorithm	MiniSAT	GNU Prolog
Kohn's map	81	509	775	28	1	221
BIOMD000000175	3042	118	194	∞	137000	∞
BIOMD000000205	32	194	313	21	1	34
BIOMD000000239	64	51	72	2980	1	22

Table 2: Computation time in milliseconds on the hardest instances of biochemical networks.

Model 175 represents a quantitative model that relates the EGF and HRG stimulations of the ErbB receptors to the activation of ERK and AKt in MCF-7 breast cancer cells [2]. This is the first model to take into account both the ERK and AKt pathways, four ErbB receptors, and their simultaneous activation by two ligands. Previous models of ErbB (e.g. the model developed in [43]) were limited to a single ErbB receptor because of combinatorial complexity. As a result, the ErbB signaling network is highly connected and indeed the underlying Petri net contains the highest number of arcs, and of organizations as remarked in [18, 25], of the BioModels repository.

6.4 Clause Density Analysis

The enumeration of the set of all minimal siphons is a problem of enumeration of all the solutions of an NP-complete problem, so the question is: *why are the CSP-based algorithms for enumerating siphons so efficient on the existing benchmarks, even on large graphs from systems biology?*

One possible explanation could be obtained by considering the well-known phase transition phenomenon in 3-SAT. The probability that a random 3-SAT problem is satisfiable has been shown to undergo a sharp phase transition as the ratio of the number of clauses over the number of variables crosses the critical value of about 4.26 [34,14], going from satisfiability to unsatisfiability with probability one, when the number of variables grows to infinity. It is in this region of the density that the SAT instances are difficult to decide, while before and after that density the instances are usually easy.

This density of SAT instances associated to the enumeration of all minimal siphons, grows during enumeration since clauses are added for each solution found. We can thus check whether the initial density of the 3-SAT instances associated to BioModels instances are greater than the critical value of 4.26. If the initial density is above the critical value, it will remain above, and the instances are thus easy because there will be a small number of solutions. On the other hand, if the initial density is below the threshold value of 4.26, the computation time may be long because the threshold value were the 3-SAT hardest instances are may be traversed, and the clauses may be satisfiable with an exponential number of solutions.

The density of a SAT instance is:

$$\text{density} = \frac{\#clauses}{\#variables}$$

Considering our problem of enumerating minimal siphons of a general Petri net $PN = (P, T, W)$ on our $\mid P \mid$ Boolean variable, initially we have $\sum_{t \in T} \mid t^\bullet \mid$ clauses plus one clause of non-empty siphon:

$$\text{density} = \frac{\sum_{t \in T} \mid t^\bullet \mid + 1}{\mid P \mid}$$

To transform a general SAT instance to a 3-SAT instance, we add μ clauses and μ variables:

$$\text{density}_{\text{3-SAT}} = \frac{\sum_{t \in T} \mid t^\bullet \mid + 1 + \mu}{\mid P \mid + \mu}$$

where

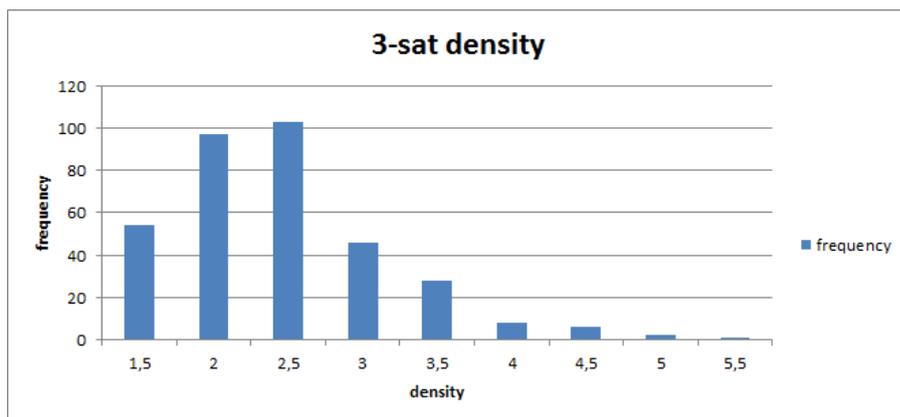$$\mu = \sum_{t \in T} \max(0, \mid {}^\bullet t \mid - 2)$$

Fig. 11: Distribution of the initial density of the initial 3-SAT problems considered to enumerate all minimal siphons of the Petri net from the BioModels benchmark.

The initial density distribution of all BioModels instances are illustrated in the histogram of Figure 11.

This histogram shows that the initial density is in fact below the critical value for the majority of models. The initial 3-SAT density of our hardest model number 175 equals 2.39. Since the density grows during enumeration by adding the clauses for minimality, the possibility to traverse the critical region of density exists. Density considerations thus do not suffice to explain why we are so efficient in enumerating all the solutions of an NP-complete problem in large classes of practical Petri nets.

6.5 Tree-Width Analysis

There is also a rich literature about the polynomial-time complexity of CSPs when the constraint hypergraph is bounded relatively to a variety of graph measures, including cutwidth and tree-width [21]. Since we have shown that both Q-MINIMALSIPHON and k-SIPHON are expressible as CSPs, the existence of a polynomial algorithm for deciding these properties for Petri net graphs of bounded tree-width follows by Prop. 5. However, since these problems are also expressible in Monadic Second Order (MSO) logic, they are in fact of linear time complexity in this case, as shown by Theorems 4 and 6.

It is thus interesting to measure the tree-widths of the Petri nets of our benchmark. QuickBB [8] is a program for computing the tree-width of a graph. When given enough time, this algorithm yields the exact tree-width of the

---

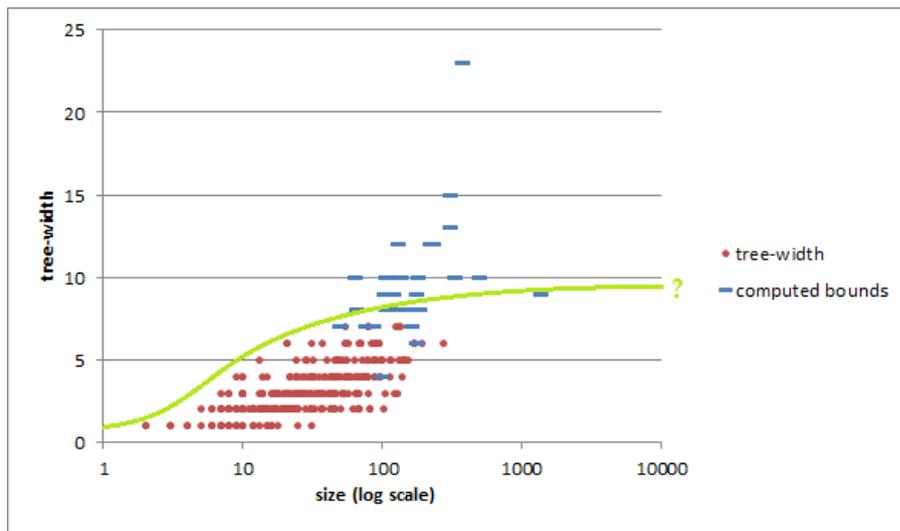[8] http://graphmod.ics.uci.edu/group/quickbb/

24

Fig. 12: Variation of the computed tree-widths as a function of size (places and transitions) of the Petri nets associated to BioModels. The computed exact tree-widths are depicted by red circles. When not known, the computed upper bounds are depicted by blue bars.

graph. When stopped before termination, it yields an upper bound of the tree-width.

We have applied QuickBB on the 432 curated models of our BioModels benchmark. For 31 models, the exact tree-width could not be computed in a time-out of one hour, but the tree-width was bounded by 23. For the remaining 342 models, the exact tree-width was computed and was always less than 10 as shown in Figure 12. The tree-width of our three hardest instances are given in Table 3, and have been determined to be less than 10 and 15. These tree-width values are relatively small values for graphs of hundreds of species and reactions.

model	# places	# transitions	tree-width
BIOMD000000175	118	194	≤ 15
BIOMD000000205	194	313	≤ 10
BIOMD000000239	51	72	≤ 10

Table 3: Tree-width of the hardest instances of BioModels database.

The BioModels benchmark thus seems to indicate that biochemical reaction networks have a bounded tree-width less than ten. This is in agreement with the idea that even very large biochemical processes are not fully intercon-

25

nected as in a grid, but composed through interface molecular species. These considerations suggest that biochemical networks are of bounded tree-width, in which case Theorems 4, 6 and 8 show that the minimal siphon decision problems are indeed tractable, and in fact of linear time theoretical complexity.

7 Conclusion

Siphons and traps in Petri nets are meaningful pools of places that display a specific behaviour in the Petri net dynamics, and that guarantee some persistence properties in the simulation of a system of biochemical reactions, independently of the kinetics.

We have described a Boolean model for the problem of enumerating all minimal siphons in a Petri net and have compared two Boolean methods to the state-of-the-art algorithm from the Petri net community [11]. On the benchmark of 345 biological models from the curated part of the BioModels repository, the Boolean method for enumerating all minimal siphons using MiniSAT is very efficient. It also scales very well in the size of the net. The CLP(B) program also solves all but one instances of the benchmark, with a better performance than MiniSAT in average, but does not scale-up as well on the largest size Petri nets, such as for instance on Kohn's map with 509 species and 775 reactions. The MiniSAT solver and the CLP(B) program both outperform the dedicated algorithms of the Petri net wommunity by one or two orders of magnitude and solve instances out of reach of these other algorithms.

The efficiency of the MiniSAT and CLP(B) methods for enumerating all solutions of an NP-complete problem for all, including large, instances of our practical benchmarks was quite surprising and lead us to study the theoretical complexity of these problems. Besides the proofs of NP-completeness of the existence of a siphon of a given cardinality, and of co-NP-completeness of the siphon-trap property, we have shown that these decision problems are tractable in Petri nets of bounded tree-width. Then we have shown that the BioModels benchmark of large biochemical networks have indeed a relatively small tree-widths.

These various results militate for the analysis of biochemical networks with Petri net concepts and Constraint Programming tools.

References

1. R. Backofen, S. Will, and E. Bornberg-Bauer. Application of constraint programming techniques for structure prediction of lattice proteins with extended alphabets. *Bioinformatics*, 15(3):234–242, 1999.

2. M. R. Birtwistle, M. Hatakeyama, N. Yumoto, B. A. Ogunnaike, J. B. Hoek, and B. N. Kholodenko. Ligand-dependent responses of the ErbB signaling network: experimental and modeling analysis. *Molecular Systems Biology*, 3(144), Sept. 2007.

3. A. Bockmayr and A. Courtois. Using hybrid concurrent constraint programming to model dynamic biological systems. In *Proceedings of ICLP'02, International Conference on Logic Programming*, volume 2401 of *Lecture Notes in Computer Science*, pages 85–99, Copenhagen, 2002. Springer-Verlag.

4. L. Calzone, A. Gelay, A. Zinovyev, F. Radvanyi, and E. Barillot. A comprehensive modular map of molecular interactions in RB/E2F pathway. *Molecular Systems Biology*, 4(173), 2008.

5. N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biochemical interaction networks. *Theoretical Computer Science*, 325(1):25–44, Sept. 2004.

6. F. Chu and X.-L. Xie. Deadlock analysis of petri nets using siphons and mathematical programming. *IEEE Transactions on Robotics and Automation*, 13(6):793–804, 1997.

7. F. Commoner. Deadlocks in petri nets. *Applied Data Research Inc., Wakefield, MA*, 1972.

8. F. Corblin, S. Tripodi, E. Fanchon, D. Ropers, and L. Trilling. A declarative constraint-based method for analyzing discrete genetic regulatory networks. *Biosystems*, 98(2):91–104, 2009.

9. R. Cordone, L. Ferrarini, and L. Piroddi. Characterization of minimal and basis siphons with predicate logic and binary programming. In *Proceedings of IEEE International Symposium on Computer-Aided Control System Design*, pages 193–198, 2002.

10. R. Cordone, L. Ferrarini, and L. Piroddi. Some results on the computation of minimal siphons in petri nets. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii USA, dec 2003.

11. R. Cordone, L. Ferrarini, and L. Piroddi. Enumeration algorithms for minimal siphons in petri nets based on place constraints. *IEEE transactions on systems, man and cybernetics. Part A, Systems and humans*, 35(6):844–854, 2005.

12. B. Courcelle. The monadic second-order logic of graphs i. recognizable sets of finite graphs. *Information and Computation*, pages 12–75, 1990.

13. B. Courcelle and I. Durand. Automata for the verification of monadic second-order graph properties. *Journal of Applied Logic*, pages 368–409, 2012.

14. J. M. Crawford and L. D. Auton. Experimental results on the crossover point in satisfiability problems. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 21–27. AAAI press, 1993.

15. L. F. de Figueiredo, S. Schuster, C. Kaleta, and D. A. Fell. Can sugars be produced from fatty acids? a test case for pathway analysis tools. *Bioinformatics*, 25(1):152–158, 2009.

16. V. Devloo, P. Hansen, and M. Labbe. Identification of all steady states in large biological systems by logical analysis. *Bulletin of Mathematical Biology*, 65:1025–1051, 2003.

17. D. Diaz and P. Codognet. Design and implementation of the GNU Prolog system. *Journal of Functional and Logic Programming*, 6, Oct. 2001.

18. P. Dittrich and P. di Fenizio. Chemical organisation theory. *Bulletin of Mathematical Biology*, 69(4):1199–1231, Apr. 2007.

19. F. Fages, S. Soliman, and R. Coolen. CLPGUI: a generic graphical user interface for constraint logic programming. *Journal of Constraints, Special Issue on User-Interaction in Constraint Satisfaction*, 9(4):241–262, Oct. 2004.

20. E. Fanchon, F. Corblin, L. Trilling, B. Hermant, , and D. Gulino. Modeling the molecular network controlling adhesion between human endothelial cells: Inference and simulation using constraint logic programming. In *CMSB'04: Proceedings of the 20 international conference on Computational Methods in Systems Biology*, pages 104–118. Springer-Verlag, 2004.

21. G. Gottlob, N. Leone, and F. Scarcello. A comparison of structural CSP decomposition methods. *Artificial Intelligence*, 124:2000, 2000.

22. R. Goud, K. van Hee, R. Post, and J. van der Werf. Petriweb: A repository for petri nets. In S. Donatelli and P. Thiagarajan, editors, *Petri Nets and Other Models of Concurrency - ICATPN 2006*, volume 4024 of *Lecture Notes in Computer Science*, pages 411–420. Springer-Verlag, 2006.

23. M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *8th Int. School on Formal Methods for the Design of Computer, Communication and Software Systems: Computational Systems Biology SFM'08*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264, Bertinoro, Italy, Feb. 2008. Springer-Verlag.

24. S. Helfert, A. Estevez, B. Bakker, P. Michels, and C. Clayton. Roles of triosephosphate isomerase and aerobic metabolism in trypanosoma brucei. *Biochem. J.*, 357:117–125, 2001.

25. C. Kaleta, S. Richter, and P. Dittrich. Using chemical organization theory for model checking. *Bioinformatics*, 25(15):1915–1922, 2009.

26. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Proceedings of a Symposium on the Complexity of Computer Computations*, pages 85–103, New York, NY, 1972. IBM Research Symposia Series, Plenum Press.

27. M. Kinuyama and T. Murata. Generating siphons and traps by petri net representation of logic equations. In *Proceedings of 2th Conference of the Net Theory SIG-IECE*, pages 93–100, 1986.

28. J. Kneis, A. Langer, and P. Rossmanith. Courcelle's theorem - a game-theoretic approach. *Discrete Optimization*, 8(4):568–594, 2011.

29. K. W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*, 10(8):2703–2734, Aug. 1999.

30. A. Larhlimi and A. Bockmayr. A new constraint-based description of the steady-state flux cone of metabolic networks. *Discrete Applied Mathematics*, 157(10):2257–2266, 2009. Networks in Computational Biology.

31. K. Lautenbach. Linear algebraic calculation of deadlocks and traps. In G. Voss and Rozenberg, editors, *Concurrency and Nets Advances in Petri Nets*, pages 315–336, New York, 1987. Springer-Verlag.

32. N. le Novère, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J. L. Snoep, and M. Hucka. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acid Research*, 1(34):D689–D691, Jan. 2006.

33. M. Minoux and K. Barkaoui. Deadlocks and traps in petri nets as horn-satisfiability solutions and some related polynomially solvable problems. *Discrete Applied Mathematics*, 29:195–210, 1990.

34. D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of sat problems. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 459–465. AAAI press, 1992.

35. T. Murata. Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–579, Apr. 1989.

36. F. Nabli. Finding minimal siphons as a CSP. In *CP'11: The Seventeenth International Conference on Principles and Practice of Constraint Programming, Doctoral Program*, pages 67–72, Sept. 2011.

37. F. Nabli, F. Fages, T. Martinez, and S. Soliman. A boolean model for enumerating minimal siphons and traps in petri-nets. In *Proceedings of CP'2012, 18th International Conference on Principles and Practice of Constraint Programming*, volume 7514 of *Lecture Notes in Computer Science*, pages 798–814. Springer-Verlag, Oct. 2012.

38. F. Nabli and S. Soliman. Steady-state solution of biochemical systems, beyond S-Systems via T-invariants. In P. Quaglia, editor, *CMSB'10: Proceedings of the 8th International Conference on Computational Methods in Systems Biology*, pages 14–22. CoSBi, ACM, Oct. 2010.

39. O. Oanea, H. Wimmel, and K. Wolf. New algorithms for deciding the siphon-trap property. In *PETRI NETS'10 Proceedings of the 31st international conference on Applications and Theory of Petri Nets*, pages 267–286. Springer-Verlag, 2010.

40. J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall, New Jersey, 1981.

41. V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman. Petri net representations in metabolic pathways. In L. Hunter, D. B. Searls, and J. W. Shavlik, editors, *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 328–336. AAAI Press, 1993.

42. N. Robertson and P. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

43. B. Schoeberl, C. Eichler-Jonsson, E. D. Gilles, and G. Muller. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors. *Nature Biotechnology*, 20(4):370–375, 2002.

44. S. Soliman. Invariants and other structural properties of biochemical models as a constraint satisfaction problem. *Algorithms for Molecular Biology*, 7(15), May 2012.

45. L. Stryer. *Biochemistry*. Freeman, New York, 1995.

46. S. Tanimoto, M. Yamauchi, and T. Watanabe. Finding minimal siphons in general petri nets. *IEICE Trans. on Fundamentals in Electronics, Communications and Computer Science*, pages 1817–1824, 1996.

47. A. von Kamp and S. Schuster. Metatool 5.0: fast and flexible elementary modes analysis. *Bioinformatics*, 22(15):1930–1931, 2006.

48. M. Yamauchi and T. Watanabe. Time complexity analysis of the minimal siphon extraction problem of petri nets. *EICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, pages 2558–2565, 1999.

49. I. Zevedei-Oancea and S. Schuster. Topological analysis of metabolic networks based on petri net theory. *In Silico Biology*, 3(29), 2003.