Exhibition of a Structural Bug with Wings

Florent Avellaneda and Rémi Morin

Aix Marseille Université, CNRS, LIF UMR 7279, 13288, Marseille, France

Abstract. Checking the structural boundedness and the structural termination of vector addition systems with states boils down to detecting pathological cycles. As opposed to their non-structural variants which require exponential space, these properties need polynomial time only. The algorithm searches for a counter-example in the form of a multiset of arcs computed by means of linear programming. Yet the minimal length of a pathological cycle can be exponential in the size of the system which makes it difficult to visualize and to analyze the detected bug in details. Further minimizing the length or the number of distinct arcs in pathological paths is NP-hard.
In this paper we propose to represent pathological cycles in the form of a multiset of particular cycles called wings. We present an algorithm that builds in polynomial time a multiset of wings with a common starting point from the multiset of arcs that represents a pathological cycle. Interestingly the number of distinct wings we need is at most equal to the dimension of vectors which helps to describe in a concise way the underlying bug and to analyse it.
Next we tackle the problem of computing a pathological multiset built over wings with a bounded length. We show how to solve this problem in polynomial time by a reduction to a linear program using a separation algorithm.

1 Introduction

Consider a set of reactions that takes place among a collection of particles such that each reaction consumes a multiset of available particles and produces a linear combination of other particle types. This kind of framework can be formalized by a vector addition system [10] or, equivalently, a (pure) Petri net. In this case, particles are called *tokens* and particle types are called *places*. Consider in addition a control state that determines which reactions can occur, and such that the occurrence of a reaction leads to a possibly distinct control state. Then the model becomes formally a vector addition system with states (a VASS), a notion introduced in [8]. Checking reachability properties of these systems is equivalent to checking a Petri net using a well-known and simple simulation technique. In this paper we are interested in two *structural properties* for VASS, that is, properties that do not depend on a particular initial distribution of particles among places. In this way, we consider the initial marking as a parameter of the system. Interestingly, we give an example that shows that the usual simulation of a VASS by a Petri net does not preserve these properties in general. As a consequence, the analysis of structural properties of Petri nets by a reduction to linear programming [14, 16, 17] does not apply to the framework of VASS.

The first problem we consider asks whether the number of particles in the system remains bounded for each initial configuration. In other words only finitely many distinct configurations can be reached. Since particles often represent the consumption of

resources, such as messages in channels, this first problem asks whether there exists some amount of resources sufficient to cope with all configurations reachable from any fixed *finite* set of potential initial configurations. A second basic issue is to check that a given system terminates, i.e. whether there is no infinite execution, for each initial configuration. Thus we aim at checking that a system eventually deadlocks. Although one usually tries to avoid deadlocks in concurrent systems, termination remains in some cases a basic problem in formal verification: In particular non-termination can result from livelocks in concurrent programs when components fail to achieve their tasks.

Verifying the structural boundedness or the structural termination of a given VASS boils down to checking the costs of cycles within the system viewed as a weighted directed graph: A cycle is pathological for structural boundedness (resp. structural termination) if its arc weights sum to a positive (resp. non-negative) vector. Consequently these two problems are very close to the detection of a zero-cycle in dynamic graphs [9], which asks if there exists a cycle with a zero cost. In [11] Kosaraju and Sullivan showed how to decide the existence of such a cycle in polynomial time. Besides this problem was proved later to be equivalent to the general linear programming problem [4]. The idea is twofold. First cycles are identified with particular multisets of arcs. Second multisets of arcs with zero cost appear as solutions to some linear program. This technique adapts easily to the detection of pathological cycles for structural boundedness or structural termination. The resulting algorithm returns in polynomial time a multiset of arcs that represents a pathological cycle if such a cycle exists.

Structural properties consider systems with an arbitrary initial configuration. However, they can be checked for systems provided with an initial configuration, because a structurally bounded (resp. structurally terminating) system is bounded (resp. terminating) for any initial configuration. This abstraction approach can prove to be useful because the non-structural variants require both exponential space [2, 13]. In this direction, we give in Section 2.3 an example that shows that it can be appropriate in some cases to split the set of places into two parts: The places that are known to be bounded for the given initial marking and those that are considered to have no specific initial content. One can then unfold the system into a new system in which the former places are encoded within control states and the remaining places are checked for structural properties. When the property is not satisfied, the analysis of a computed pathological cycle is necessary to detect a false counter-example, that is to say, to verify the validity of the abstraction.

When the model of a system does not satisfy a given property, formal verification tools usually provide users with a counter-example execution in the form of a sequence of atomic steps that describes an unexpected behaviour. In this paper, we tackle the problem of providing a useful description of a pathological cycle for a structural property. The point is that the number of times an arc occurs in a pathological cycle can be exponential in the size of the given VASS, even though the time needed to compute the corresponding multiset of arcs is only polynomial. Consequently listing the sequence of arcs occurring along such a cycle is prohibitive in general. A first approach consists in providing a partial description of the detected pathological cycle as the set of all arcs occurring in this cycle —or simply the set of places interacting in the reactions per-

formed by these arcs. However, this information may not be sufficient to understand fully the detected bug.

In the particular case of a VASS with a single state —that is to say: a pure Petri net— a multiset of arcs can be regarded as a multiset of cycles with a common starting state. Moreover, due to Carathéodory's theorem [15, Cor. 7.7i], we need at most p distinct arcs to describe a structural bug if the given VASS has p places. Then each pathological cycle is decomposed into p elementary cycles of length 1 and with a common starting state. In this work, we want to extend this property to any VASS: *We aim at decomposing a given pathological cycle in the form of a multiset of particular cycles starting from a common fixed state. Moreover each component cycle should be easy to depict and the number of distinct cycles in this multiset should be at most equal to the number of places in the given VASS.*

We introduce in Section 3 a class of particular cycles, called *wings*, that are used as component cycles for the decomposition of a pathological cycle. Roughly speaking, a wing consists of a cycle provided with two paths back and forth from a fixed starting state to some particular state within the cycle. We require that the length of the three component paths of a wing is at most equal to the number of states in the given VASS. Actually we will often consider *simple wings*, that is, wings whose component paths are simple paths. Additionally, the *valuation* of a wing determines the number of iterations of its cyclic component. Indeed we can describe a wing to the user of a verification tool by listing the sequence of arcs of its three component paths and giving its valuation.

Our first main result is established in Section 4. We show how to compute in polynomial time a multiset of simple wings with a common starting state that corresponds to a given multiset of arcs that represents a pathological cycle. Moreover the number of distinct simple wings we need is at most equal to the number of places. Thus we propose to describe a structural bug to the user in the form of a small number of wings together with the number of times each wing occurs. Note that this information allows us to compute the minimal configuration required to execute the pathological cycle resulting of the iteration of each wing in some arbitrary order. This information is useful to the user when structural properties are checked instead of their non-structural variants, if the abstraction process yields a false counter-example. Then the analysis of the detected pathological cycle can lead to a refined model with a reduced set of non-initialized places.

Finding shortest counter-examples is often desirable in automated verification, because they are easier to analyse, see e.g. [3, 12]. Unfortunately, searching for a pathological cycle built over a minimal number of arcs, or a minimal number of interacting places, is NP-hard (Prop. 18 and 19). Yet we show in Section 5 that we can minimize in polynomial time the length of the component paths in wings used to describe a pathological path. To do so, we fix a starting state q and a natural number ℓ and we focus on wings starting from q whose component paths have a length at most ℓ. By means of an encoding in linear programming and a separation algorithm, we show how to decide whether there exists a pathological multiset of such wings, and if so, to compute one (Theorem 23). In the rest of this paper, we focus on structural termination for simplicity's sake. However, all results adapt easily to structural boundedness.

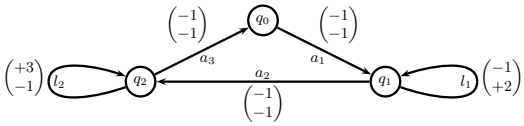

Fig. 1. A vector addition system with states

2 Background

Let p be a fixed non-zero natural number. A vector addition system with states is simply a directed graph whose arcs are labeled by vectors from $\mathbb{Z}^p$.

Definition 1. *[8] A* vector addition system with states *(for short, a VASS) is a pair $\mathcal{S} = (Q, A)$ where Q is a finite set of states, and $A \subseteq Q \times \mathbb{Z}^p \times Q$ is a finite set of arcs labeled by vectors from $\mathbb{Z}^p$.*

Throughout the paper we let $\mathcal{S} = (Q, A)$ be a VASS. We let $|Q|$ and $|A|$ denote the cardinalities of Q and A respectively. The source and the target of a labeled arc $a \in A$ are denoted by $\mathrm{dom}(a)$ and $\mathrm{cod}(a)$ respectively. We let $\mathrm{cost}(a) \in \mathbb{Z}^p$ denote the column vector labeling each arc $a \in A$. The size of a VASS $\mathcal{S} = (Q, A)$ is $\mathrm{size}(\mathcal{S}) = |A| \times (2 \times \lceil \log_2(|Q| + 1) \rceil + p \times (1 + \lceil \log_2(1 + v_{\max}) \rceil))$ where $v_{\max}$ is the maximal absolute value of coefficients of vectors labeling arcs in $\mathcal{S}$.

2.1 Basics and Notations

Let $\mathcal{S} = (Q, A)$ be a VASS. A path is a sequence of arcs $\gamma = a_1 ... a_n \in A^\star$ such that we have $\mathrm{dom}(a_{i+1}) = \mathrm{cod}(a_i)$ for each $i \in [1..n-1]$. A path $\gamma = a_1 ... a_n \in A^\star$ is closed if $n \geqslant 1$ and $\mathrm{dom}(a_1) = \mathrm{cod}(a_n)$. A closed path is called a *cycle*. A path $\gamma = a_1 ... a_n \in A^\star$ is *simple* if $\mathrm{dom}(a_i) \neq \mathrm{dom}(a_j)$ for all distinct i, j. A *circuit* is a simple and closed path. The cost of a path $\gamma = a_1 ... a_n$ is the vector $\mathrm{cost}(\gamma) = \sum_{i=1}^{i=n} \mathrm{cost}(a_i)$. Further the cost of a multiset of arcs $x \in \mathbb{N}^A$ is $\mathrm{cost}(x) = \sum_{a \in A} x[a] \cdot \mathrm{cost}(a)$ and the cost of a finite multiset of paths $\mathcal{F}$ is $\mathrm{cost}(\mathcal{F}) = \sum_{\gamma \in A^\star} \mathcal{F}[\gamma] \cdot \mathrm{cost}(\gamma)$. Let v and v' be two integral vectors with n coordinates: $v = (v[1], ..., v[n])$ and $v' = (v'[1], ..., v'[n])$. We put as usual $v \geqslant v'$ if $v[i] \geqslant v'[i]$ for each i; $v > v'$ if $v[i] > v'[i]$ for each i; and $v \gneq v'$ if $v \geqslant v'$ and $v \neq v'$.

A configuration is a pair $(q, r) \in Q \times \mathbb{N}^p$ consisting of a control state q and a multiset of available particles r. A labeled arc $a \in A$ is enabled at the configuration (q, r) and leads to the configuration (q', r') if $\mathrm{dom}(a) = q$, $\mathrm{cod}(a) = q'$, and $r + \mathrm{cost}(a) = r'$. An execution of $\mathcal{S}$ from an initial configuration $(q_{\mathrm{in}}, r_{\mathrm{in}})$ is a sequence of labeled arcs $a_1 ... a_n \in A^\star$ such that there are configurations (q_0, r_0), ..., (q_n, r_n) for which $(q_0, r_0) = (q_{\mathrm{in}}, r_{\mathrm{in}})$ and for each $i \in [1..n]$, the labeled arc a_i is enabled at (q_{i-1}, r_{i-1}) and leads to (q_i, r_i). Then the configuration (q_n, r_n) is reachable from $(q_{\mathrm{in}}, r_{\mathrm{in}})$.

In this paper we are mainly interested in checking the structural termination of a given VASS: We want to verify that for each initial configuration $(q_{\mathrm{in}}, r_{\mathrm{in}})$ the length of executions from $(q_{\mathrm{in}}, r_{\mathrm{in}})$ is bounded. It is easy to observe with the help of Dickson's lemma [10, Lemma 4.1] that this property is equivalent to the condition that there exists no cycle γ with $\mathrm{cost}(\gamma) \geqslant 0$. Thus we aim at detecting pathological cycles in $\mathcal{S}$.

Definition 2. *A cycle γ in a VASS $\mathcal{S}$ is* pathological *if cost$(\gamma) \geqslant 0$.*

Example 3. Along this paper, we shall use as a running example the 2-dimensional VASS depicted in Figure 1 with three states q_0, q_1, and q_2 and five weighted arcs a_1, a_2, a_3, l_1, and l_2. The cost of the cycle $\gamma = a_1.l_1^5.a_2.l_2^3.a_3$ is cost$(\gamma) = (1, 4)^\top$. So this cycle is pathological.

2.2 Multisets of Arcs vs. Cycles

We shall represent cycles of a VASS $\mathcal{S}$ as particular multisets of arcs. Let $x \in \mathbb{N}^A$ be a multiset of arcs. We denote by $\|x\| = |\{a \in A \mid x[a] \geqslant 1\}|$ the number of distinct arcs in x and by A_x the *support of* x, that is to say the set of arcs $a \in A$ such that $x[a] \geqslant 1$. Thus $\|x\| = |A_x|$. The *underlying graph* G_x of x is the (undirected) graph $G_x = (Q_x, E_x)$ where the set of vertices $Q_x = \{\mathrm{dom}(a) \mid a \in A_x\} \cup \{\mathrm{cod}(a) \mid a \in A_x\}$ collects the source and the target of all arcs in x and the set of edges $E_x = \{\{\mathrm{dom}(a), \mathrm{cod}(a)\} \mid a \in A_x \text{ and } \mathrm{dom}(a) \neq \mathrm{cod}(a)\}$ keeps track of all connections induced by arcs in x.

A multiset of arcs $x \in \mathbb{N}^A$ is called *connected* if G_x is a connected graph. Let $x \in \mathbb{N}^A$ and $C_1, ..., C_n \subseteq Q_x$ be the connected components of G_x. For each $1 \leqslant i \leqslant n$ and each $a \in A$, we put $x_i[a] = x[a]$ if $\mathrm{dom}(a) \in C_i$ and $x_i[a] = 0$ otherwise. Then $x = x_1 + ... + x_n$ and the multisets $x_i \in \mathbb{N}^A$ are called the *connected components of* x. A multiset of arcs x is called *Eulerian* if for each state $q \in Q$ the number of arcs incident from q equals the number of arcs incident to q, i.e. $\sum_{\mathrm{dom}(a)=q} x[a] = \sum_{\mathrm{cod}(a)=q} x[a]$. A connected and Eulerian multiset of arcs is called a *circulation*. Note that if x and y are Eulerian, then $x + y$ is Eulerian. If moreover $x \leqslant y$ then $y - x$ is Eulerian, too. The *multiplicity* of a non-zero multiset $x \in \mathbb{N}^A \setminus \{\mathbf{0}\}$ within a multiset $y \in \mathbb{N}^A$ is the greatest natural number k such that $k \cdot x \leqslant y$.

Each cycle $\gamma = a_1...a_n$ of $\mathcal{S}$ is represented by the multiset of arcs $x_\gamma = \sum_{i=1}^{i=n} a_i$, i.e. $x_\gamma[a]$ is the number of occurrences of a in γ. Since γ is a cycle, the multiset of arcs x_γ is non-empty, Eulerian and connected. For instance, continuing Example 3, the multiset of arcs $a_1 + a_2 + a_3 + 5 \cdot l_1 + 3 \cdot l_2$ is the circulation corresponding to the cycle $\gamma = a_1.l_1^5.a_2.l_2^3.a_3$. Conversely, each non-empty circulation corresponds to a cycle of $\mathcal{S}$: This is an immediate variant of Euler's theorem [5, Th. 1.8.1].

Proposition 4. *Let $x \in \mathbb{N}^A$ be a non-empty circulation. Then there exists a cycle γ such that $x_\gamma = x$.*

In [11], Kosaraju and Sullivan showed how to detect a cycle with a zero cost in polynomial time. Basically their algorithm searches for a non-empty circulation with a zero cost recursively by alternatively solving homogeneous linear programs and computing strongly connected components. It is straightforward to adapt this technique to the detection of pathological cycles. In fact it is sufficient to replace a vector equality $x = \mathbf{0}$ by $x \geqslant \mathbf{0}$ in part of the linear programs considered. Moreover we can require that the resulting algorithm returns a circulation that represents a pathological cycle if such a cycle exists. Note here that this algorithm remains polynomial although it does not boil down to solving a linear program as in the particular case of a Petri net [17].

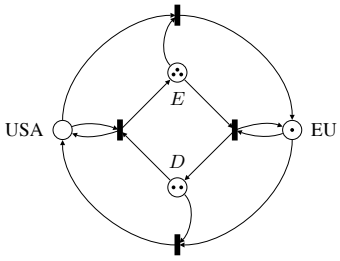

Fig. 2. A terminating Petri net

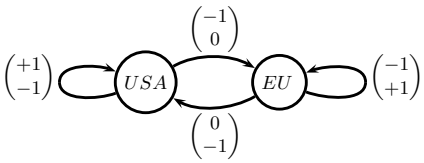

Fig. 3. A structurally terminating VASS

2.3 Semi-Structural Properties of Petri Nets

When modeling a message-passing system as a Petri net, one often distinguishes two types of places:

- *control places* whose bounded marking describes the current global state;
- *container places* whose tokens represent pending messages.

It may be then interesting to check termination for a fixed initial marking of control places but an arbitrary initial marking of container places. In this way, semi-structural termination generalises both termination and structural termination by specifying a subset of places with an arbitrary initial marking.

A simple approach allows us to check semi-structural termination. First we erase the container places and check that the resulting Petri net is bounded. Next we build the corresponding finite marking graph viewed as a VASS and re-incorporate the constraints of container places. If the resulting VASS is structurally terminating, then the original Petri net is semi-structurally terminating, i.e. it terminates for any initial marking of its container places. Recall that checking termination of a Petri net requires exponential space [17] whereas we can check structural termination of a VASS in polynomial time. Thus, considering semi-structural termination of a Petri net and hence structural termination of a VASS can turn out to be efficient to check that a Petri net terminates.

Example 5. Consider the currency change Petri net depicted in Fig. 2. The container places E and D collect euros and dollars respectively. An additional token walks around between the two control places EU and USA. When the control token is in EU then euros can be changed into dollars, and conversely if the control token is in USA then dollars can be changed into euros. Moving from EU to USA (resp. from USA to EU) requires to pay a tax in dollars (resp. in euros). This Petri net is not structurally terminating because currency can circulate between euros and dollars provided that there is a token in both control places EU and USA. However, the resulting unfolded VASS, depicted in Fig. 3, consists of two states and is obviously structurally terminating. Thus the currency change Petri net from Fig. 2 terminates for any initial amount. Note that the usual Petri net associated with the VASS from Fig. 3 is precisely the Petri net from Fig. 2. Therefore the classical simulation of a VASS by a Petri net does not preserve structural termination.

3 Representation of a Circulation by a Multiset of Cycles

3.1 Exponential Length of Minimal Pathological Cycles

The algorithm to detect pathological paths can provide us with a circulation that corresponds to a pathological cycle. Moreover the *size* of the natural coefficients of such a circulation is polynomial. In order to help the understanding of a structural bug detected in the form of a circulation, it is useful to represent this counter-example as a pathological cycle. Then the length of this pathological cycle equals the sum of the circulation coefficients. Consequently the minimal length of the resulting cycle can be exponential in the size of the VASS as illustrated by the next example.

Example 6. Consider the VASS with a single state and six arcs labeled by the six following 6-dimensional vectors:

$$t_1 = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} ; t_2 = \begin{pmatrix} -1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} ; t_3 = \begin{pmatrix} 0 \\ -1 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} ; t_4 = \begin{pmatrix} 0 \\ 0 \\ -2 \\ 1 \\ 0 \\ 0 \end{pmatrix} ; t_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -2 \\ 1 \\ 0 \end{pmatrix} ; t_6 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2 \\ 1 \end{pmatrix}$$

It is easy to see that each pathological cycle needs all arcs because of their pairwise dependencies. Moreover a pathological cycle that contains one occurrence of t_6 needs 2 occurrences of t_5, 4 occurrences of t_4 and hence 4 occurrences of t_3, 2 occurrences of t_2 and one occurrence of t_1. Therefore the pathological cycle $\gamma = t_1 + 2 \cdot t_2 + 4 \cdot t_3 + 4 \cdot t_4 + 2 \cdot t_5 + t_6$ has a minimal length. We can easily generalize this example to a VASS made of $2 \times m$ arcs whose pathological cycles have a length greater than $2 \times (2^m - 1)$.

Thus listing the sequence of arcs occurring along a pathological cycle is prohibitive. For that reason we need to design a *compact representation* of pathological cycles.

3.2 Looking for a Format

It is clear that a pathological cycle γ (or a circulation) can be decomposed into a multiset $\mathcal{F}$ of circuits with $\text{cost}(\mathcal{F}) = \text{cost}(\gamma)$. Then Caratheodory's theorem [15, Cor. 7.7i] allows us to compute a multiset $\mathcal{F}'$ over at most p circuits (where p stands for the dimension of vectors) such that $\text{cost}(\mathcal{F}') = m \cdot \text{cost}(\gamma)$ for some $m \in \mathbb{N} \setminus \{0\}$. However, the connectedness of the underlying set of arcs may be lost at this point, that is, $\mathcal{F}'$ does not represent a pathological cycle any longer.

A natural idea is to use an additional connecting cycle on which the component circuits would hang. In other words it would be nice to find

- a sequence of circuits $\sigma_0, \ldots, \sigma_{k-1}$, with $k \leqslant p$,
- a sequence of fixed connection states $q_0, \ldots, q_{k-1}$ with $q_i \in Q_{\sigma_i}$
- a connecting cycle $w_0 \ldots w_{k-1}$, where w_i is a simple path from q_i to $q_{i+1 \pmod k}$,
- and a sequence $n_0, \ldots, n_{k-1}$ of natural numbers

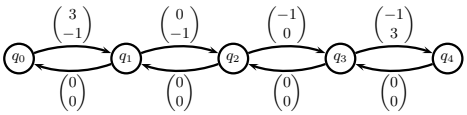

Fig. 4. Counter-example

such that the cycle $\gamma' = \sigma_0^{n_0} w_0 \sigma_1^{n_1} \dots \sigma_{k-1}^{n_{k-1}} w_{k-1}$ satisfies $\text{cost}(\gamma') = m \cdot \text{cost}(\gamma)$ for some $m \in \mathbb{N} \setminus \{0\}$. Example 3 shows that in some cases pathological circulations can effectively be decomposed in this way. However, till now, it remains open whether it exists such a pathological cycle for every non structurally terminating VASS. For that reason, we consider in the sequel of this paper another kind of representation for pathological circulations. Before that, we would like to stress that we cannot require additionally that the connecting cycle $w_0.w_1 \dots w_k$ is simple, as the next example shows.

Example 7. Consider the 2-dimensional VASS with 5 states from Fig. 4. Each pathological cycle in this VASS makes use of each arc. Such cycles cannot be decomposed in the above considered form with a *simple* connecting cycle.

3.3 From Multisets of Arcs to Multisets of Wings

At present we propose to describe pathological cycles of a VASS in the form of a multiset of particular cycles called wings. Roughly speaking, a wing with valuation k is a cycle which consists of k iterations of a circuit plus a path back and forth from one state of the circuit to some fixed starting state. This shared starting state will ensure that a multiset of wings remains connected.

Definition 8. *Let $q, q' \in Q$ be two states of $\mathcal{S}$. Let γ_0 be a cycle of $\mathcal{S}$ starting from q'. Let γ_1 be a path from q to q' and γ_2 be a path from q' to q. Let $k \in \mathbb{N}$. We assume that the length of each path γ_0, γ_1 and γ_2 is at most equal to the number of states $|Q|$. Let $W = \gamma_1.\gamma_0^k.\gamma_2$ be the cycle which starts from q and which consists of γ_1, followed by k iterations of the cycle γ_0, followed by γ_2. Then W is called a* wing *of $\mathcal{S}$ with* valuation *k. A wing is said to be* simple *if its three component paths γ_0, γ_1, and γ_2 are simple.*

A simple wing is often represented by a multiset of arcs $W = D + k \cdot C$ where C is the set of arcs occurring in the cycle γ_0 while D is the multiset of arcs occurring in γ_1 and γ_2. Then the multiset W is connected and Eulerian. Note that the path $\gamma_1.\gamma_2$ from q to q in a simple wing need not be simple (nor non-empty). However, each arc occurs *at most twice* in $\gamma_1.\gamma_2$.

Example 9. We continue Example 3 with $p = 2$. We have observed that the cost of the cycle γ is $\text{cost}(\gamma) = (1, 4)^\top$. Consider the two simple wings $W_1 = a_1.l_1^{10}.a_2.a_3$ with valuation 10 and $W_2 = a_1.a_2.l_2^6.a_3$ with valuation 6. Noteworthy $2 \cdot \text{cost}(\gamma) = \text{cost}(W_1) + \text{cost}(W_2)$. This equality illustrates precisely how simple wings can represent a cycle up to a scalar multiplication factor of its cost.

Our first result asserts that there exists such a representation by wings with a shared starting state for any pathological circulation.

Theorem 10. *Let $\hat{H}$ be a non-empty circulation and $\hat{q} \in Q_{\hat{H}}$. There exists a non-empty multiset $\mathcal{F}$ of simple wings starting from $\hat{q}$ such that cost$(\mathcal{F}) = m \cdot$ cost$(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$; moreover $\mathcal{F}$ is built over at most p distinct wings.*

The next section is devoted to the proof of Theorem 10. The factor m is necessary to make sure that the simple wings obtained share the common starting state $\hat{q}$ and hence to get an obvious cycle made of this multiset of wings. This factor m is not a drawback of this approach because we search for pathological cycles and moreover the actual length of the resulting pathological cycle is not relevant. It allows us also to ensure additionally that $\mathcal{F}$ is built over of at most p distinct wings.

4 Construction of Representative Wings from a Circulation

In this section we fix a non-empty circulation $\hat{H} \in \mathbb{N}^A$ and a state $\hat{q} \in Q_{\hat{H}}$. We show how to compute in polynomial time a non-empty multiset $\mathcal{F}$ of simple wings starting from $\hat{q}$ such that cost$(\mathcal{F}) = m \cdot$ cost$(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$.

The construction of $\mathcal{F}$ proceeds inductively over the size of $A_{\hat{H}}$. At each step, a wing $W = D + k \cdot C \leqslant \hat{H}$ with valuation k is added to $\mathcal{F}$ and removed from $\hat{H}$ until $\hat{H}$ is empty. This wing should satisfy the three following properties:

1. Some arc in the cyclic component C has multiplicity k within $\hat{H}$; in this way, at least one arc is removed from the support of $\hat{H}$ at each step: $\|\hat{H} - W\| < \|\hat{H}\|$.
2. The Eulerian multiset of remaining arcs $\hat{H} - W$ is connected; this ensures that we can proceed recursively.
3. The fixed state $\hat{q}$ belongs to the new circulation $\hat{H} - W$, so that all wings share this common starting state —except of course if $\hat{H} - W$ is already empty.

The first idea for the search of such a wing W within $\hat{H}$ is that it is sufficient to find a circuit C satisfying these conditions. This leads us to the following central notion of an *adequate* circuit.

Definition 11. *Let $H \in \mathbb{N}^A$ be a non-empty circulation and $q_0 \in Q_H$. A circuit C with multiplicity $k \geqslant 1$ in H is* adequate *for H and q_0 if it satisfies the two next conditions:*

- *the multiset of arcs $H - k \cdot C$ is connected;*
- *if $H - k \cdot C$ is not empty then $Q_{H-k\cdot C}$ contains q_0.*

Example 12. Continuing Example 3, we consider the circulation $H = a_1 + a_2 + a_3 + 5 \cdot l_1 + 3 \cdot l_2$ for the VASS depicted in Figure 1. Then the two circuits l_1 and l_2 are adequate for H and q_0 whereas the circuit $a_1.a_2.a_3$ is not.

Note that $\|H - k \cdot C\| < \|H\|$ for any circuit C with multiplicity k in H. The construction of $\mathcal{F}$ relies on two independent algorithms presented in the two next subsections. The first algorithm shows how to find an adequate circuit for any non-empty circulation $H \in \mathbb{N}^A$ and any state $\hat{q} \in Q_H$. The second one is much easier. It explains how to build the expected multiset $\mathcal{F}$ of wings with the help of adequate circuits as inputs.

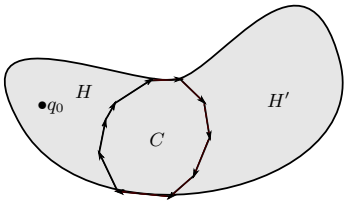

Fig. 5. Searching for an adequate circuit

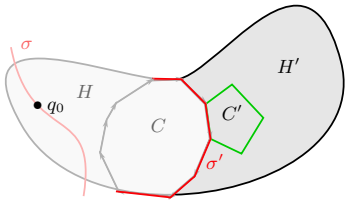

Fig. 6. Induction step

4.1 Finding an Adequate Circuit in a Circulation for a Fixed State

The search for a circuit C adequate for H and q_0 proceeds non-deterministically and inductively over the number of arcs in A_H. Each step distinguishes two main cases. The simpler case assumes that all circuits within H contain q_0. Then each circuit is adequate for H and q_0. The reason is that any connected component of the Eulerian multiset $H - k \cdot C$ contains a circuit, and hence contains q_0.

The more interesting case considers that there exists a circuit $C \leqslant H$ that does not contain q_0. Let k be the multiplicity of C within H. Then $q_0 \in Q_{H-k\cdot C}$ because q_0 does not occur in C. Hence $H - k \cdot C$ is not empty. Then the circuit C is adequate if $H - k \cdot C$ is connected. In this case, the search is terminated. Otherwise we consider a connected component H' of $H - k \cdot C$ that does not contain q_0, as illustrated in Fig. 5. We will show how to find in H' a circuit C', with multiplicity k' in H', such that

1. at least one arc $a \in A_{C'} \setminus A_C$ satisfies $H'[a] = k'$. Then $H'[a] = H[a]$ and k' is also the multiplicity of a in H; hence $\|H - k' \cdot C'\| < \|H\|$.
2. each connected component of $H' - k' \cdot C'$ contains a state from C. Then $H - k' \cdot C'$ is connected; moreover $q_0 \in Q_{H-k'\cdot C'}$ because q_0 does not occur in H'.

It follows that C' is adequate for H and q_0.

The search for an appropriate circuit C' within H' is regarded as a generalisation of the search for an adequate circuit C within H where the connectivity of $H - k \cdot C$ is replaced by the connectivity of $H' - k' \cdot C'$ if one incorporates the circuit C. Actually, for simplicity's sake, we will consider at this point a simple path σ made of all but one arcs from C. Intuitively, σ will play the role of C. However we shall also consider a special case where σ is the empty path to deal with adequate circuits.

Definition 13. *Let $H \in \mathbb{N}^A$ be a non-empty circulation, $q_0 \in Q_H$, and $\sigma \in A^\star$ be a simple path. A circuit C with multiplicity $k \geqslant 1$ in H is* appropriate *for H and (q_0, σ) if it satisfies the two next conditions:*

1. *there exists an arc $a \in A_C \setminus A_\sigma$ such that $H[a] = k$;*
2. *each connected component of $H - k \cdot C$ contains a state from $Q_\sigma \cup \{q_0\}$.*

Observe that a circuit C is appropriate for H and (q_0, ϵ) where ϵ denotes the empty path (Def. 13) if, and only if, it is adequate for H and q_0 (Def. 11). For that reason, the search for an adequate circuit will simply ask for an appropriate circuit w.r.t. the empty path ϵ in Algorithm 2 below.

We present now in Algorithm 1 a way to compute circuits appropriate for H and (q_0, σ), provided that σ is not a circuit and $q_0 \in Q_\sigma$ if σ is not empty.

Algorithm 1 AppropriateCircuit(H, q_0, σ)

Require: $H \in \mathbb{N}^A$ is a non-empty circulation.
Require: σ is a simple path consisting of arcs from A and such that σ is not a circuit.
Require: $q_0 \in Q_H$ and $q_0 \in Q_\sigma$ if the path σ is non-empty.
  if all circuits $C \leqslant H$ satisfy $Q_C \cap (Q_\sigma \cup \{q_0\}) \neq \emptyset$ **then**
    Let $b \in A_H \setminus A_\sigma$
    $\beta \leftarrow b$ # Initially β is a path of length 1
    while β contains no circuit **do**
      if there exists some arc $b' \in A_H \setminus A_\sigma$ with $\mathrm{dom}(b') = \mathrm{cod}(b)$ **then**
        Choose some $b' \in A_H \setminus A_\sigma$ with $\mathrm{dom}(b') = \mathrm{cod}(b)$
      else
        Choose some $b' \in A_H \cap A_\sigma$ such that $\mathrm{dom}(b') = \mathrm{cod}(b)$
      end if
      $b \leftarrow b'$
      Add the arc b to the end of the path β
    end while
    return a circuit C within β
  else
    Let $C \leqslant H$ be such a circuit such that $Q_C \cap (Q_\sigma \cup \{q_0\}) = \emptyset$
    Let k be the multiplicity of C in H
    if each connected component of $H - k \cdot C$ contains a state from $Q_\sigma \cup \{q_0\}$ **then**
      return C # In particular if $H = k \cdot C$.
    else
      Let H' be a connected component of $H - k \cdot C$ with $Q_{H'} \cap (Q_\sigma \cup \{q_0\}) = \emptyset$.
      Let q_0' be a state from $Q_{H'} \cap Q_C$ and a be an arc from A_C with $H[a] = k$.
      Let σ' be the path made of all arcs from $A_C \setminus \{a\}$
      return AppropriateCircuit(H', q_0', σ') # Then $\|H'\| < \|H\|$
    end if
  end if

Proposition 14. *Let $H \in \mathbb{N}^A$ be a circulation. Let $q_0 \in Q_H$ and $\sigma \in A^\star$ be a simple path such that $q_0 \in Q_\sigma$ if σ is not empty. Provided that σ is not a circuit, Algorithm 1 returns a circuit that is appropriate for H and (q_0, σ).*

Assume that $H \in \mathbb{N}^A$ is a non-empty circulation and $\sigma = a_1...a_n$ is a simple path consisting of arcs from A such that σ is not a circuit. Let $q_0 \in Q_H$ be a state of H such that $q_0 \in Q_\sigma$ if σ is non-empty. Searching for an appropriate circuit C for H and (q_0, σ) is slightly more involved than searching for an adequate one. However, Algorithm 1 proceeds similarly to the above discussion and distinguishes two main cases.

We need first to determine whether all circuits in H contain a state from $Q_\sigma \cup \{q_0\}$. To do so, one considers the subset $A' \subseteq A$ consisting of all arcs from A_H whose source and target do not belong to $Q_\sigma \cup \{q_0\}$. Let A'_1,..., A'_n be the strongly connected components of A'. Then there exists a circuit C in H with $Q_C \cap (Q_\sigma \cup \{q_0\}) = \emptyset$ if, and only if, A' contains a self-loop arc or one of the strongly connected components A'_i has two states. Depending on whether this condition is satisfied, we investigate one of the following two cases:

1. We assume first that all circuits in H contain a state from $Q_\sigma \cup \{q_0\}$. Algorithm 1 builds a circuit $C = a_0 a_1 ... a_{n-1}$ in H using preferably arcs that do not appear in σ. Since σ is not a circuit and H is a non-empty circulation, we can choose an arbitrary arc $b \in A_H \setminus A_\sigma$ and consider first the path $\beta = b$. This path is extended iteratively by adding arcs from A_H to the end of β until β contains a circuit C. At each iteration, there are potential candidates to complete β because H is Eulerian. However, we require that arcs from $A_H \setminus A_\sigma$ are preferred to the others in this extension process. Clearly this loop terminates after at most $|Q_H|$ iterations. At this point, we claim that C is appropriate for H and (q_0, σ).
   Proof. Let $k \geqslant 1$ be the multiplicity of C in H. Since H is Eulerian, $H - k \cdot C$ is Eulerian. Let H' be a connected component of $H - k \cdot C$. Since $H - k \cdot C$ is Eulerian, H' is Eulerian. Therefore there is some circuit in H' and hence H' contains a state from $Q_\sigma \cup \{q_0\}$. Thus, all connected components of $H - k \cdot C$ contain a state from $Q_\sigma \cup \{q_0\}$.
   Since the simple path σ is not closed, the circuit C within β cannot be made of arcs from σ only. In other words, C contains at least one arc that does not belong to A_σ. Assume that there is an arc $a_i \in A_\sigma \cap A_C$. Due to the priority of arcs adopted, the arc a_i is the single arc with $\mathrm{dom}(a_i) = \mathrm{cod}(a_{i-1 \pmod n})$. Since H is Eulerian, we have $H[a_{i-1 \pmod n}] \leqslant H[a_i]$. Since C contains at least one arc that does not belong to A_σ, there exists an arc $a \in A_C \setminus A_\sigma$ such that $H[a] \leqslant H[a_i]$. It follows that there exists $a \in A_C \setminus A_\sigma$ such that $H[a]$ is equal to the multiplicity C in H. ■
2. We assume now that there exists some circuit C in H with $Q_C \cap (Q_\sigma \cup \{q_0\}) = \emptyset$. Let $k \geqslant 1$ be the multiplicity of C in H. If each connected component of $H - k \cdot C$ contains at least one state from $Q_\sigma \cup \{q_0\}$ then C is appropriate for H and (q_0, σ). Therefore we assume now that $H - k \cdot C$ is non-empty and admits some connected component H' of $H - k \cdot C$ that contains no state from $Q_\sigma \cup \{q_0\}$. Let $a \in A_C$ be such that $H[a] = k$. Then $H'[a] = 0$ and hence $\|H'\| < \|H\|$. Moreover $Q_{H'} \cap Q_C \neq \emptyset$, otherwise there would be no path from $Q_{H'}$ to Q_C in the circulation H. We fix some state $q_0' \in Q_{H'} \cap Q_C$. We let also σ' denote the simple path made of all arcs from $A_C \setminus \{a\}$. Then σ' contains all arcs from $A_C \cap A_{H'}$. Moreover σ' is not a circuit and $q_0' \in Q_{\sigma'}$ as soon as σ' is not empty. At this point we claim that any circuit C' appropriate for H' and (q_0', σ') is also appropriate for H and (q_0, σ).
   Proof. The situation is illustrated in Fig. 6. Let $k' \geqslant 1$ be the multiplicity of C' in H'. Then,
   - Each connected component of $H' - k' \cdot C'$ contains a state from $Q_{\sigma'} \cup \{q_0'\}$.
   - There exists an arc $a' \in A_{C'} \setminus A_{\sigma'}$ such that $H'[a'] = k'$.

   Since σ' contains all arcs from C that occur in H', we have $a' \notin A_C$. Therefore $H[a'] = (H - k \cdot C)[a'] = H'[a'] = k'$. It follows that k' is also the multiplicity of C' in H. Since H' contains no state from $Q_\sigma \cup \{q_0\}$, C' contains no state from $Q_\sigma \cup \{q_0\}$ either. Further, we have $a' \in A_{C'} \setminus A_\sigma$. Since $q_0 \in H$ and $q_0 \notin H'$, q_0 appears in $H - k' \cdot C'$. To conclude the proof, we show simply that $H - k' \cdot C'$ is connected.
   Since $H - k \cdot C \geqslant k' \cdot C'$, we have $H - k' \cdot C' \geqslant k \cdot C \geqslant C$. Thus all states of Q_C are strongly connected to each other in $H - k' \cdot C'$. Let $q'' \in Q_{H - k' \cdot C'}$. It remains to show that there exists a path from q'' to a state from C made of arcs

from $H - k' \cdot C'$. The claim is trivial if $q'' \in Q_C$. If $q'' \notin Q_C$ then q'' belongs to one of the connected components of $H - k \cdot C$. We distinguish two cases:

- $q'' \in Q_{H'}$. Since $q'' \in Q_{H-k' \cdot C'}$, there exists some arc $a'' \in H - k' \cdot C'$ such that $q'' = \mathrm{dom}(a'')$ or $q'' = \mathrm{cod}(a'')$. Since $q'' \notin Q_C$, we have $a'' \notin C$ and hence $H[a''] = H'[a'']$. Then $H'[a''] - k' \cdot C'[a''] = H[a''] - k' \cdot C'[a''] \geqslant 1$. It follows that $q'' \in Q_{H'-k' \cdot C'}$. Since each connected component of $Q_{H'-k' \cdot C'}$ contains a state from $Q_{\sigma'} \cup \{q'_0\}$ and $Q_{\sigma'} \cup \{q'_0\} \subseteq Q_C$, there exists a path from q'' to C in $H' - k' \cdot C'$ and hence in $H - k' \cdot C'$.
- $q'' \in Q_{H''}$ where H'' is a connected component of $H - k \cdot C$ different from H'. Then $Q_{H''} \cap Q_C \neq \emptyset$ otherwise there would be no path from the set of states $Q_{H''}$ to the set of states Q_C in H. Therefore there exists a path from q'' to C in H'' and hence in $H - k' \cdot C'$.

Thus $H - k' \cdot C'$ is connected and the circuit C' is appropriate for H and (q_0, σ). ■

4.2 Building a Multiset of Simple Wings from a Pathological Circulation

The construction of a representative multiset $\mathcal{F}$ of simple wings from the multiset $\hat{H}$ of arcs is described in Algorithm 2. Initially $\mathcal{F}$ is empty and we put $H = \hat{H}$. Hence $\mathrm{cost}(\mathcal{F}) + \mathrm{cost}(H) = m \cdot \mathrm{cost}(\hat{H})$ with $m = 1$. This equality will act as a loop invariant of the main iterating process. First, a circuit C adequate for $\hat{H}$ and $\hat{q}$ is found with the help of Algorithm 1. Recall here that a circuit C is appropriate for H and $(\hat{q}, \epsilon)$ (where ϵ denotes the empty path) if, and only if, it is adequate for H and $\hat{q}$. Let k be the multiplicity of C in H. Then the Eulerian multiset $H - k \cdot C$ is connected and $\hat{q} \in Q_{H-k \cdot C}$ provided that $H - k \cdot C$ is not empty. Moreover $\|H - k \cdot C\| < \|H\|$.

We build from C a wing W starting from $\hat{q}$ with C as its cyclic component. If $\hat{q}$ appears in C then $W = k \cdot C$ is a simple wing starting from $\hat{q}$. Assume that $\hat{q} \notin Q_C$. Then $\hat{q} \in Q_{H-k \cdot C}$. Since H is connected, there is a state $q \in Q_C \cap Q_{H-k \cdot C}$. Since $H - k \cdot C$ is connected, there are a simple path γ_1 from $\hat{q}$ to q and a simple path γ_2 from q to $\hat{q}$ made of arcs from $A_{H-k \cdot C}$. We let D denote the multiset of arcs that corresponds to the cycle $\gamma_1.\gamma_2$. Then the multiset $W = D + k \cdot C$ represents a simple wing which starts from $\hat{q}$. Moreover $D[a] \leqslant 2$ for each $a \in A$ because γ_1 and γ_2 are simple paths, hence $W \leqslant 3 \cdot H$, because $k \cdot C \leqslant H$. Furthermore, each arc $a \in A_C$ with multiplicity k in H does not occur in $\gamma_1.\gamma_2$, since it does not occur in $H - k \cdot C$. We distinguish then three cases:

1. If $W = H$ then the simple wing W is added to $\mathcal{F}$ and removed from H leading to the empty multiset $H' = \mathbf{0}$.
2. If $W \leqslant H$, $H - W$ is connected and $\hat{q} \in Q_{H-W}$ then the simple wing W is added to $\mathcal{F}$ and removed from H leading to the new circulation $H' = H - W$ such that $\hat{q} \in Q_{H'}$. Since k is the multiplicity of C in H, we get $\|H'\| < \|H\|$.
3. Otherwise the multiset of wings $\mathcal{F}$ is multiplied by 3. Then we have $\mathrm{cost}(\mathcal{F}) + \mathrm{cost}(3 \cdot H) = m \cdot \mathrm{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$. Let a be an arc from C such that $H[a] = k$. Then $3 \cdot H[a] - D[a] = 3k$ because a does not occur in $\gamma_1.\gamma_2$. On the other hand, for each arc a' from C with $H[a'] \geqslant k+1$, we have $3 \cdot H[a'] - D[a'] \geqslant 3k + 1$ because $D[a'] \leqslant 2$. It follows that $3k$ is the multiplicity of C in $3 \cdot H - D$. We consider the new wing $W' = D + 3k \cdot C$. The wing W' is added to $\mathcal{F}$ and

Algorithm 2 Computing a multiset of simple wings

Require: A non-empty circulation $\hat{H}$ and a state $\hat{q} \in Q_{\hat{H}}$
  $\mathcal{F} \leftarrow \mathbf{0}$ # Initially $\mathcal{F}$ is the empty multiset of simple wings
  $H \leftarrow \hat{H}$ # Initially $\text{cost}(\mathcal{F}) + \text{cost}(H) = m \cdot \text{cost}(\hat{H})$ with $m = 1$
  while $H \neq \mathbf{0}$ **do**
    $C \leftarrow \text{AppropriateCircuit}(H, \hat{q}, \epsilon)$ # C is adequate for H and $\hat{q}$.
    Let k be the multiplicity of C in H # $k \cdot C \leqslant H$ and $H - k \cdot C$ is connected
    if $\hat{q} \in Q_C$ **then**
      $D \leftarrow \mathbf{0}$ # $D \in \mathbb{N}^A$ is the empty multiset of arcs
      $W \leftarrow k \cdot C$ # The multiset W represents a simple wing such that $W \leqslant H$
    else
      Let q be some state in $Q_C \cap Q_{H-k\cdot C}$.
      Let γ_1 be a simple path from $\hat{q}$ to q made of arcs from $A_{H-k\cdot C}$.
      Let γ_2 be a simple path from q to $\hat{q}$ made of arcs from $A_{H-k\cdot C}$.
      Let D be the multiset of arcs that corresponds to the cycle $\gamma_1.\gamma_2$. # Then $D \leqslant 2 \cdot H$
      $W \leftarrow D + k \cdot C$ # The multiset W represents a simple wing such that $W \leqslant 3 \cdot H$
    end if
    if $(H = W)$ or $(W \leqslant H$ and $H - W$ is connected and $\hat{q} \in Q_{H-W})$ **then**
      Add the simple wing W to $\mathcal{F}$.
      $H \leftarrow H - W$ # $\text{cost}(\mathcal{F}) + \text{cost}(H) = m \cdot \text{cost}(\hat{H})$ for some $m \geqslant 1$
    else
      $W' \leftarrow D + 3k \cdot C$ # We have $A_{H-k\cdot C} = A_{3\cdot H - W'}$
      $\mathcal{F} \leftarrow 3 \cdot \mathcal{F}$ # $\text{cost}(\mathcal{F}) + \text{cost}(3 \cdot H) = m \cdot \text{cost}(\hat{H})$ for some $m \geqslant 1$
      Add the simple wing W' to $\mathcal{F}$.
      $H \leftarrow 3 \cdot H - W'$ # $\text{cost}(\mathcal{F}) + \text{cost}(H) = m \cdot \text{cost}(\hat{H})$ for some $m \geqslant 1$
    end if
  end while
  return $\mathcal{F}$

removed from $3 \cdot H$ leading to the new Eulerian multiset of arcs $H' = 3 \cdot H - W'$. For each $a \in A$, we have $3(H - k \cdot C)[a] \geqslant H'[a] \geqslant 3(H - k \cdot C)[a] - 2$, because $D[a] \leqslant 2$. Hence $A_{H'} = A_{H-k\cdot C}$. Consequently, H' is connected, $\|H'\| < \|H\|$, and $\hat{q} \in Q_{H'}$ if $H' \neq \mathbf{0}$.

Thus, in all cases we get that H' is Eulerian and connected. Moreover $\hat{q} \in Q_{H'}$ provided that H' is not empty and hence the next iteration of the algorithm can proceed analogously. Furthermore we have $\|H'\| < \|H\|$ henceforth Alg. 2 terminates after at most $|A|$ iterations.

Example 15. We continue Examples 3 and 12 to illustrate an execution of Alg. 2 with the VASS depicted in Figure 1, the circulation $\hat{H} = a_1 + a_2 + a_3 + 5 \cdot l_1 + 3 \cdot l_2$, and the base state $\hat{q} = q_0$. First, the adequate circuit l_1 with multiplicity 5 can be chosen which leads to the wing $W_1 = a_1 + a_2 + a_3 + 5 \cdot l_1$. Since $\hat{H} - W_1$ does not contain $\hat{q}$, we put $W_1' = a_1 + a_2 + a_3 + 15 \cdot l_1$ and get $\mathcal{F} = \{W_1'\}$ and $H = 3 \cdot \hat{H} - W_1' = 2 \cdot a_1 + 2 \cdot a_2 + 2 \cdot a_3 + 9 \cdot l_2$ at the end of the first iteration.

In the second iteration, l_2 is the unique adequate circuit for H and $\hat{q}$. Therefore we put $W_2 = a_1 + a_2 + a_3 + 9 \cdot l_2$ and get $\mathcal{F} = \{W_1', W_2\}$ and $H' = H - W_2 = a_1 + a_2 + a_3$

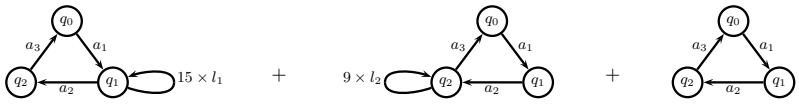

Fig. 7. Multiset of wings computed in Example 15

because this Eulerian multiset of arcs is connected and contains $\hat{q}$. The third and last iteration selects the adequate circuit $W_3 = a_1 + a_2 + a_3$ which yields the multiset of wings $\mathcal{F} = \{W_1', W_2, W_3\}$ depicted in Fig. 7. Observe here that $\text{cost}(\mathcal{F}) = (3, 12)^\top = 3 \cdot \text{cost}(\hat{H})$.

It is clear that the property that $\text{cost}(\mathcal{F}) + \text{cost}(H) = m \cdot \text{cost}(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$ is a loop invariant of Algorithm 2. Consequently,

Theorem 16. *Let $\hat{H}$ be a non-empty circulation and $\hat{q} \in Q_{\hat{H}}$. Algorithm 2 returns a non-empty multiset $\mathcal{F}$ of simple wings starting from $\hat{q}$ such that cost$(\mathcal{F}) = m \cdot$ cost$(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$.*

Clearly $\mathcal{F}$ is made of at most $|A|$ wings. Moreover the valuation of each wing in $\mathcal{F}$ is at most $3^{|A|} \times \max_{a \in A} \hat{H}[a]$. Since $\hat{H}$ is obtained from our variant of Kosaraju and Sullivan's algorithm, the size of $\hat{H}$ is polynomial in the size of $\mathcal{S}$. Thus, the size of the valuation of each wing in $\mathcal{F}$ is also polynomial in the size of $\mathcal{S}$.

4.3 An Upper Bound for the Number of Distinct Simple Wings

Since Algorithm 2 terminates in less than $|A|$ iterations, it provides us with a multiset $\mathcal{F}$ of simple wings starting from the arbitrarily fixed state $\hat{q}$ with at most $|A|$ distinct wings. We can make sure that the representative multiset $\mathcal{F}$ contains at most p distinct wings.

This results essentially from Carathéodory's theorem [15, Cor. 7.7i] which states that for each set $X \subseteq \mathbb{Q}^p$ of p-dimensional rational vectors, any rational vector $v \in \mathbb{Q}^p$ that lies in $\text{Cone}(X) = \{\lambda_1 \cdot x_1 + ... + \lambda_n \cdot x_n \mid n \geqslant 1; x_1, ..., x_n \in X; \lambda_1, ..., \lambda_n \in \mathbb{Q}^+\}$ lies in $\text{Cone}(X')$ for some $X' \subseteq X$ with $|X'| \leqslant p$, i.e. $v = \lambda_1 \cdot x_1 + ... + \lambda_n \cdot x_n$ with $p \geqslant n \geqslant 1$, $x_1, ..., x_n \in X$ and $\lambda_1, ..., \lambda_n \in \mathbb{Q}^+$.

Consider a multiset of wings $\mathcal{F} = k_1 \cdot W_1 + ... + k_n \cdot W_n$ with $\text{cost}(\mathcal{F}) \geqslant \mathbf{0}$. Carathéodory's theorem ensures that there are rational numbers $\lambda_1, ..., \lambda_n \in \mathbb{Q}^+$ such that $\text{cost}(\mathcal{F}) = \lambda_1 \cdot \text{cost}(W_1) + ... + \lambda_n \cdot \text{cost}(W_n)$ and $\lambda_i \neq 0$ for at most p values of i. Actually these rational numbers λ_i can be found using linear programming. Further Euclid's algorithm enables us to compute the least common multiple m of the denominators of all λ_i. Then we get $m \cdot \text{cost}(\mathcal{F}) = k_1' \cdot \text{cost}(W_1) + ... + k_n' \cdot \text{cost}(W_n) \geqslant \mathbf{0}$ with $k_i' \in \mathbb{N}$ and $k_i' \neq 0$ for at most p values of i. Hence,

Corollary 17. *Let $\hat{H}$ be a non-empty circulation and $\hat{q} \in Q_{\hat{H}}$. We can compute in polynomial time a multiset $\mathcal{F}$ built over* at most *p distinct simple wings starting from $\hat{q}$ such that cost$(\mathcal{F}) = m \cdot$ cost$(\hat{H})$ for some $m \in \mathbb{N} \setminus \{0\}$.*

Since our algorithm is polynomial, the size of the valuation of these wings and the size of the number of occurrences of these wings are polynomial in the size of $\mathcal{S}$.

5 Searching for Minimal Counter-Examples

Shortest counter-examples are usually more valuable in the debugging phase, because they focus on the actual causes of the bug and hence they are easier to understand [3, 12]. That is why many verification tools offer to search for an erroneous path with a minimal length, see e.g. with Spin [7]. Several directions can be followed to describe a structural bug of a VASS in a minimal way. Pathological cycles with a minimal length are not that interesting in general because their length can be exponential in the size of the system (Example 6). The first natural approach we consider consists in searching for pathological cycles with a minimal number of distinct arcs. However, with no surprise,

Proposition 18. *Computing a pathological cycle of a VASS with a minimal number of distinct arcs is NP-hard.*

Since multisets of wings with a common starting state are a particular case of cycles and each pathological cycle can be represented by a pathological multiset of wings over the same set of arcs, Prop. 18 applies to the particular case of multisets of wings with a common starting state.

A coordinate $i \in [1..p]$ is said to be *involved* in an arc a if $\text{cost}(a)[i] \neq 0$. The set of interacting coordinates in a cycle collects all coordinates involved in its arcs. A second natural approach aims at minimizing the number of interacting coordinates in a pathological cycle. Again, with no surprise,

Proposition 19. *Computing a pathological cycle of a VASS with a minimal number of interacting coordinates is NP-hard.*

Similarly to Prop. 18, this result applies to pathological multisets of wings with a common starting state. Thus searching for minimal multisets of wings appears to be hard in general.

In this section, we consider the problem of finding a pathological multiset of wings whose component paths have a minimal length. We show how to solve this problem in polynomial time using a separation algorithm. To do so, we fix a starting state $\hat{q}$ and a natural number ℓ and we focus on wings starting from $\hat{q}$ whose component paths have length at most ℓ. We show how to decide whether there exists a pathological multiset made of these wings, and if so, to compute one in polynomial time. In this way, we can minimize the length of the component paths used in a pathological multiset of wings.

5.1 An Upper Bound for the Valuations of Wings

Let $\mathcal{S} = (Q, A)$ be a VASS, $\hat{q} \in Q$ be a fixed state of $\mathcal{S}$ and $\ell \in \mathbb{N}$. For simplicity's sake, *we call length of a wing the maximal length of its component paths*. However, the results presented here can be adapted to the case where the length of a wing is the sum of the lengths of its component paths. We want to determine whether there exists a multiset $\mathcal{F}$ made of wings starting from $\hat{q}$ with length at most ℓ such that $\text{cost}(\mathcal{F}) \geqslant \mathbf{0}$. We observe first that we can restrict the search to wings with a valuation at most equal to 2^{Φ} where Φ is polynomial in the size of $\mathcal{S}$.

Lemma 20. *Let $\mathcal{F}$ be a non-empty multiset of wings starting from $\hat{q}$ with length at most ℓ such that cost$(\mathcal{F}) \geqslant \mathbf{0}$. Let $\Phi = 96 \times p^4 \times$ size$(\mathcal{S})$. Then there exists a non-empty finite multiset $\mathcal{F}'$ of wings starting from $\hat{q}$ with length at most ℓ and valuation at most 2^{Φ} such that cost$(\mathcal{F}') \geqslant \mathbf{0}$.*

Proof. By Cor. 17, there are a positive natural number $n \leqslant p$ and n wings $W_1, ..., W_n$ such that the system (Sys1) of $p + n$ inequalities

$$\begin{aligned} \textstyle\sum_{i=1}^{n} k_i \cdot \text{cost}(W_i) &\geqslant \mathbf{0} \\ k_i &> 0 \text{ for each } i \in [1..n] \end{aligned}$$

has an integral solution. We put $W_i = D_{2i} + k'_i \cdot C_{2i+1}$ where k'_i is the valuation of the wing W_i. We consider now the new system (Sys2) of $p + 2n$ inequalities

$$\begin{aligned} \textstyle\sum_{i=1}^{n} k_{2i} \cdot \text{cost}(D_{2i}) + k_{2i+1} \cdot \text{cost}(C_{2i+1}) &\geqslant \mathbf{0} \\ k_{2i} &> 0 \text{ for each } i \in [1..n] \\ k_{2i+1} &\geqslant 0 \text{ for each } i \in [1..n] \end{aligned}$$

Since (Sys1) has an integral solution, (Sys2) has an integral solution. Any integral solution to (Sys2) corresponds to some multiset $\mathcal{F}$ of wings starting from $\hat{q}$ such that cost$(\mathcal{F}) \geqslant \mathbf{0}$ and for each i, the wing $D_{2i} + k_{2i+1} \cdot C_{2i+1}$ appears once and the wing D_{2i} with valuation 0 appears $k_{2i+1} - 1$ times if $k_{2i+1} \geqslant 1$.

Recall that solving a system of linear Diophantine inequalities is NP-complete. Moreover some integral solution of such a system use polynomial space, only. The matrix from (Sys2) has $p + 2 \times n$ rows and $2 \times n$ columns. The absolute value of each component of this matrix is at most $2 \times |Q| \times v_{\max}$ where $v_{\max}$ is the maximal absolute value of components in vectors carried by arcs in $\mathcal{S}$. We can assume of course that $|A| \geqslant 1$, $|Q| \geqslant 1$ and $p \geqslant 1$. Then size$(\mathcal{S}) \geqslant \lceil \log_2(2 \times |Q| \times v_{\max} + 1) \rceil$. The size of each row is $2 \times n \times \lceil \log_2(2 \times |Q| \times v_{\max} + 1) \rceil \leqslant 2 \times p \times$ size$(\mathcal{S})$. By [15, Cor.17.1b], there exists some integral solution to (Sys2) whose size is at most $6 \times (2 \times p)^3 \times \varphi$, where the facet complexity φ is smaller than $2 \times p \times$ size$(\mathcal{S})$. Thus there is a solution to (Sys2) whose size is at most $96 \times p^4 \times$ size$(\mathcal{S}) = \Phi$. Consequently there exists some integral solution of (Sys2) where each variable k_i satisfies $k_i \leqslant 2^{\Phi}$. ∎

Note here that the number N of wings starting from $\hat{q}$ with length at most ℓ and valuation at most 2^{Φ} is exponential in the size of $\mathcal{S}$. Let W_1,..., W_N be an enumeration of these wings. Then the linear program $\sum_{i=1}^{i=N} x[i] \cdot \text{cost}(W_i) \geqslant \mathbf{0}$ with $x \in \mathbb{Q}^N$ and $x \gneq \mathbf{0}$ has a solution if and only if there exists a non-empty multiset $\mathcal{F}$ of wings starting from $\hat{q}$ with length at most ℓ (and valuation at most 2^{Φ}) such that cost$(\mathcal{F}) \geqslant \mathbf{0}$.

We consider actually a kind of dual problem. We define the linear program $\text{LP}_{\mathcal{S},\hat{q},\ell}$ for a vector $w \in \mathbb{Q}^p$ of p unknown which consists of the following two sets of constraints:

- $w[i] > 0$, for each $i \in [1..p]$;
- $-\text{cost}(W)^{\top} w > 0$, for each wing W starting from $\hat{q}$ with length at most ℓ and valuation at most 2^{Φ}.

By Gordan Theorem [15, p. 95], the linear program $\text{LP}_{\mathcal{S},\hat{q},\ell}$ has no solution if and only if there exists some non-negative non-zero linear combination of its row vectors that

Algorithm 3 (Separation algorithm)

Require: $\mathcal{S} = (Q, A)$ is a VASS, $w \in \mathbb{Q}^p$, $\hat{q} \in Q$.
Ensure: returns `true` if w is a solution to $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ and some violated inequality otherwise
  if $w \not\succ \mathbf{0}$ **then**
    return some $i \in [1..p]$ such that $w[i] \leqslant 0$.
  end if
  for $q, q' \in Q$ **do**
    Compute $\mathrm{blmw}_{q,q'}(w) \in \mathbb{Q}$ and a path $\gamma_{q,q'} \in A^\star$ in polynomial time
  end for
  for $q \in Q$ **do**
    if (*) $\mathrm{blmw}_{\hat{q},q}(w) + 2^\Phi \times \mathrm{blmw}_{q,q}(w) + \mathrm{blmw}_{q,\hat{q}}(w) \geqslant 0$ **then**
      return the row vector $\mathrm{cost}(\gamma_{\hat{q},q}) + 2^\Phi \cdot \mathrm{cost}(\gamma_{q,q}) + \mathrm{cost}(\gamma_{q,\hat{q}})$
    end if
  end for
  return `true`

sum to a non-negative vector, i.e. there exists a non-empty multiset $\mathcal{F}$ of these wings with $\mathrm{cost}(\mathcal{F}) \geqslant \mathbf{0}$.

Corollary 21. *The linear program $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ has no solution iff there exists a non-empty multiset $\mathcal{F}$ of wings starting from $\hat{q}$ with length at most ℓ such that cost$(\mathcal{F}) \geqslant \mathbf{0}$.*

5.2 Separation of Solutions

The linear program $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ consists of exponentially many inequalities. So we shall not build the whole set of its inequalities. However, we show here how to decide in polynomial time whether a given vector $w \in \mathbb{Q}^p$ is a solution to $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ or not, and, in the latter case, to compute an inequality of $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ for which w fails.

If some component $w[i]$ of w is non-positive, then the constraint $w[i] > 0$ is not satisfied. Thus we may assume that $w > \mathbf{0}$. We denote by $\mathcal{S}/w = (Q, A/w)$ the directed graph obtained from the VASS $\mathcal{S}$ by replacing the label $\mathrm{cost}(a) \in \mathbb{Z}^p$ of each arc $a \in A$ by $\mathrm{cost}(a)^\top w$. For any two states $q, q' \in Q$, we compute the maximal weight $\mathrm{blmw}_{q,q'}(w) \in \mathbb{Q}$ of the paths from q to q' in $\mathcal{S}/w$ with length at most ℓ. We compute also a path $\gamma_{q,q'} \in A^\star$ from q to q' with length at most ℓ and such that its weights sum to $\mathrm{blmw}_{q,q'}(w)$ if it is regarded as a path in $\mathcal{S}/w$, i.e. $\mathrm{cost}(\gamma_{q,q'})^\top w = \mathrm{blmw}_{q,q'}(w)$. Note that $\mathrm{blmw}_{q,q}(w) \geqslant 0$ for each $q \in Q$. Let $q \in Q$ be some state of $\mathcal{S}$. If $\mathrm{blmw}_{\hat{q},q}(w) + 2^\Phi \times \mathrm{blmw}_{q,q}(w) + \mathrm{blmw}_{q,\hat{q}}(w) \geqslant 0$ then the wing W built with the path $\gamma_{\hat{q},q}$, followed by 2^Φ iterations of the cycle $\gamma_{q,q}$ and the path $\gamma_{q,\hat{q}}$ satisfies $\mathrm{cost}(W)^\top w \geqslant 0$. Otherwise w is a solution to $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$.

Proposition 22. *Let $w \in \mathbb{Q}^p$. We can decide in polynomial time whether w is a solution to $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ or not, and, in the latter case, return an inequality of $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ for which w fails.*

5.3 Computing a Pathological Multiset of Wings with Length at most ℓ

Although the linear program $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ consists of exponentially many inequalities, the fundamental result due to Grötschel, Lovász and Schrijver [15, Th. 14.1] asserts that

it is sufficient to design a separation oracle in order to solve this linear program in polynomial time. Given a vector $w > \mathbf{0}$, the separation oracle must decide whether w is a solution to $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ or not, and, in the latter case, compute an inequality of $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ for which w fails; in other words the separation oracle must compute a wing W with length at most ℓ and valuation at most 2^{Φ} for which $\mathrm{cost}(W)^{\top} w \geqslant 0$ whenever w is not a solution to $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$. We have shown in Subsection 5.2 above how to design such an oracle. As a consequence, we get our second main result:

Theorem 23. *Let $\mathcal{S} = (Q, A)$ be a VASS, $\hat{q} \in Q$ be a particular state and ℓ be a natural number. We can decide in polynomial time whether there exists a non-empty multiset $\mathcal{F}$ of wings starting from $\hat{q}$ with length at most ℓ such that cost$(\mathcal{F}) \geqslant \mathbf{0}$.*

With no surprise, the algorithm designed by Grötschel, Lovász and Schrijver to prove [15, Th. 14.1] can provide us with a certificate that $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ has no solution in the form of polynomially many constraints from $\mathrm{LP}_{\mathcal{S},\hat{q},\ell}$ that have no solution. By Gordan Theorem again, we can derive from this certificate a multiset $\mathcal{F}$ of wings with $\mathrm{cost}(\mathcal{F}) \geqslant \mathbf{0}$. Consequently we can find in polynomial time a multiset of wings with a minimal size that describes a pathological cycle for structural termination. Further, we can guarantee that this multiset consists of at most p distinct wings.

6 Conclusion and Future Work

In this paper we tackle the problem of illustrating a structural bug detected in the form of a pathological circulation in a concise way. We propose to represent pathological cycles for structural termination as a set of wings that share a common starting state. Our main result shows how to compute a pathological multiset of wings in polynomial time (Th. 16) from any pathological circulation. Further we need only p distinct wings in such a multiset due to Carathéodory's theorem.

In practice it is interesting to search for pathological cycles (or pathological multisets of wings) with a minimal number of arcs or a minimal number of interacting places. Yet, both problems are NP-hard. Our second result is more theoretical: We have applied the separation technique from [15, Th. 14.1] to prove that one can search for wings whose component paths have a minimal length in polynomial time, too. Interestingly all results presented in this paper apply —or can be easily adapted— to structural boundedness: A VASS is said to be structurally bounded if for each initial configuration the number of reachable configurations is finite. This property corresponds to the non-existence of cycles with a non-negative non-zero cost.

Message Sequence Graphs (MSGs) are a popular formalism to describe communication protocols by means of partial orders of events called Message Sequence Charts [6]. As discussed in [1], MSGs can be regarded as a special case of VASSs when the latter are provided with a partial-order semantics. In this way, new features can be stirred into message sequence graphs such as message loss, message duplication, dynamic process creation, bounded counters or timers, etc. For that reason we found it useful to develop a prototype that implements the model-checking and the reachability tech-

niques from [1]. In the near future our verification tool will benefit from the description of structural bugs by wings presented in this paper.

Acknowledgements We would like to thank the anonymous reviewer who detected a mistake in the previous version of this paper and whose observations helped us to improve Algorithm 2 and to simplify its proof.

References

1. F. Avellaneda and R. Morin. Checking partial-order properties of vector addition systems with states. In *International Conference on Application of Concurrency to System Design*, pages 100–109, 2013.
2. H. Carstensen. Decidability questions for fairness in Petri nets. In Franz-Josef Brandenburg, Guy Vidal-Naquet, and Martin Wirsing, editors, *STACS*, volume 247 of *Lecture Notes in Computer Science*, pages 396–407. Springer, 1987.
3. E. M. Clarke, O. Grumberg, K. L. McMillan, and X. Zhao. Efficient generation of counterexamples and witnesses in symbolic model checking. In *DAC*, pages 427–432, 1995.
4. E. Cohen and N. Megiddo. Strongly polynomial-time and NC algorithms for detecting cycles in dynamic graphs (preliminary version). In David S. Johnson, editor, *STOC*, pages 523–534. ACM, 1989.
5. R. Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, 2010.
6. J. G. Henriksen, M. Mukund, K. Narayan Kumar, M. A. Sohoni, and P. S. Thiagarajan. A theory of regular MSC languages. *Information and Computation*, 202(1):1–38, 2005.
7. G. Holzmann. *The Spin model checker: primer and reference manual*. Addison-Wesley Professional, first edition, 2003.
8. J.E. Hopcroft and J-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.
9. K. Iwano and K. Steiglitz. Testing for cycles in infinite graphs with periodic structure (extended abstract). In Alfred V. Aho, editor, *STOC*, pages 46–55. ACM, 1987.
10. R.M. Karp and R.E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.
11. S. R. Kosaraju and G. F. Sullivan. Detecting cycles in dynamic graphs in polynomial time (preliminary version). In Janos Simon, editor, *STOC*, pages 398–406. ACM, 1988.
12. O. Kupferman and S. Sheinvald-Faragy. Finding shortest witnesses to the nonemptiness of automata on infinite words. In Christel Baier and Holger Hermanns, editors, *CONCUR*, volume 4137 of *Lecture Notes in Computer Science*, pages 492–508. Springer, 2006.
13. R.J. Lipton. The reachability problem requires exponential space. Technical Report 63, Yale University, 1976.
14. G. Memmi and G. Roucairol. Linear algebra in net theory. In Wilfried Brauer, editor, *Advanced Course: Net Theory and Applications*, volume 84 of *Lecture Notes in Computer Science*, pages 213–223. Springer, 1980.
15. A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
16. J. Sifakis. Structural properties of Petri nets. In Józef Winkowski, editor, *MFCS*, volume 64 of *Lecture Notes in Computer Science*, pages 474–483. Springer, 1978.
17. D.D. Sleator. Data structures and terminating Petri nets. In Imre Simon, editor, *LATIN*, volume 583 of *Lecture Notes in Computer Science*, pages 488–497. Springer, 1992.