# Reachability Games on Extended Vector Addition Systems with States

Tomáš Brázdil[1]⋆, Petr Jančar[2]⋆⋆, and Antonín Kučera[1]⋆⋆⋆

[1] Faculty of Informatics, Masaryk University, Botanická 68a, 60200 Brno, Czech Republic. {brazdil,kucera}@fi.muni.cz

[2] Center of Applied Cybernetics, Dept. of Computer Science, FEI, Technical University of Ostrava, 17. listopadu 15, 70833 Ostrava, Czech Republic. Petr.Jancar@vsb.cz

**Abstract.** We consider two-player turn-based games with zero-reachability and zero-safety objectives generated by extended vector addition systems with states. Although the problem of deciding the winner in such games in undecidable in general, we identify several decidable and even tractable subcases of this problem obtained by restricting the number of counters and/or the sets of target configurations.

## 1 Introduction

Vector addition systems with states (VASS) are an abstract computational model equivalent to Petri nets (see, e.g., [27, 29]) which is well suited for modelling and analysis of distributed concurrent systems. Roughly speaking, a *k-dimensional VASS*, where $k \geq 1$, is a finite-state automaton with $k$ unbounded counters which can store non-negative integers. Depending on its current control state, a VASS can choose and perform one of the available transitions. A given transition changes the control state and updates the vector of current counter values by adding a fixed vector of integers which *labels* the transition. For simplicity, we assume that transition labels can increase/decrease each counter by at most one. Since the counters cannot become negative, transitions which attempt to decrease a zero counter are disabled. Configurations of a given VASS are written as pairs $p\boldsymbol{v}$, where $p$ is a control state and $\boldsymbol{v} \in \mathbb{N}^k$ a vector of counter values. In this paper, we consider *extended VASS games* which enrich the modelling power of VASS in two orthogonal ways.

(1) Transition labels can contain *symbolic* components (denoted by $\omega$) whose intuitive meaning is "add an arbitrarily large non-negative integer to a given counter". For example, a single transition $p \to q$ labeled by $(1, \omega)$ represents an infinite number of "ordinary" transitions labeled by $(1, 0)$, $(1, 1)$, $(1, 2)$, … A natural source of motivation for introducing symbolic labels are systems with multiple resources that can be consumed and produced simultaneously by performing a transition. The $\omega$ components can then be conveniently used to model an unbounded "reloading" of resources.

⋆ Supported by the Institute for Theoretical Computer Science (ITI), project No. 1M0545.
⋆⋆ Supported by the Czech Ministry of Education, project No. 1M0567.
⋆⋆⋆ Supported by the Czech Science Foundation, grant No. P202/10/1469.

(2) To model the interaction between a system and its environment, the set of control states is split into two disjoint subsets of *controllable* and *environmental* states, which induces the corresponding partition of configurations. Transitions from the controllable and environmental configurations then correspond to the events generated by the system and its environment, respectively.

Hence, the semantics of a given extended VASS game $\mathcal{M}$ is a possibly infinitely-branching turn-based game $G_{\mathcal{M}}$ with infinitely many vertices that correspond to the configurations of $\mathcal{M}$. The game $G_{\mathcal{M}}$ is initiated by putting a token on some configuration $p\boldsymbol{v}$. The token is then moved from vertex to vertex by two players, $\Box$ and $\Diamond$, who select transitions in the controllable and environmental configurations according to some strategies. Thus, they produce an infinite sequence of configurations called a *play*. Desired properties of $\mathcal{M}$ can be formalized as *objectives*, i.e., admissible plays. The central problem is the question whether player $\Box$ (the system) has a *winning* strategy which ensures that the objective is satisfied for every strategy of player $\Diamond$ (the environment). We refer to, e.g., [32, 13, 35] for more comprehensive expositions of results related to games in formal verification. In this paper, we are mainly interested in *zero-safety* objectives consisting of plays where no counter is decreased to zero, i.e., a given system never reaches a situation when some of its resources are insufficient. Player $\Box$ always aims at satisfying a given zero-safety objective, while player $\Diamond$ aims at satisfying the dual *zero-reachability* objective.

As a simple example, consider a workshop which "consumes" wooden sticks, screws, wires, etc., and produces puppets of various kinds which are then sold at the door. From time to time, the manager may decide to issue an order for screws or other supplies, and thus increase their number by a finite but essentially unbounded amount Controllable states can be used to model the actions taken by workshop employees, and environmental states model the behaviour of unpredictable customers. We wonder whether the workshop manager has a strategy which ensures that at least one puppet of each kind is always available for sell, regardless of what the unpredictable customers do. Note that a winning strategy for the manager must also resolve the symbolic $\omega$ value used to model the order of screws by specifying a *concrete number* of screws that should be ordered.

Technically, we consider extended VASS games with *non-selective* and *selective* zero-reachability objectives, where the set of target configurations that should be reached by player $\Diamond$ and avoided by player $\Box$ is either $Z$ and $Z_C$, respectively. Here, the set $Z$ consists of all $p\boldsymbol{v}$ such that $\boldsymbol{v}_\ell = 0$ for some $\ell$ (i.e., some counter is zero); and the set $Z_C$, where $C$ is a subset of control states, consists of all $p\boldsymbol{v} \in Z$ such that $p \in C$.

**Our main results** can be summarized as follows:

(a) The problem of deciding the winner in $k$-dimensional extended VASS games (where $k \geq 2$) with $Z$-reachability objectives is in **(k-1)-EXPTIME**.
(b) A finite description of the winning region for each player (i.e., the set of all vertices where the player wins) is computable in $(k-1)$-exponential time.
(c) Winning strategies for both players are finitely and effectively representable.

We note that the classical result by Lipton [24] easily implies **EXPSPACE**-hardness (even in the case when player $\Diamond$ has no influence). These (decidability) results are complemented by noting the following straightforward undecidability:

(d) The problem of deciding the winner in 2-dimensional VASS games with “ordinary” (non-symbolic) transitions and $Z_C$-reachability objectives is undecidable. The same problem for 3-dimensional *extended* VASS games is *highly* undecidable (beyond the arithmetical hierarchy).

Further, we consider the special case of one-dimensional extended VASS games, where we provide the following (tight) complexity results:

(e) The problem of deciding the winner in one-dimensional extended VASS games with $Z$-reachability objectives is in **P**. Both players have “counterless” winning strategies constructible in polynomial time.
(f) The problem of deciding the winner in one-dimensional extended VASS games with $Z_C$-reachability objectives is **PSPACE**-complete. A finite description of the winning regions is computable in exponential time.

To the best of our knowledge, these are the first positive decidability/tractability results about a natural class of *infinitely branching* turn-based games, and some of the underlying observations are perhaps of broader interest (in particular, we obtain slight generalizations of the “classical” results about self-covering paths achieved by Rackoff [28] and elaborated by Rosier&Yen [30]).

To build some intuition behind the technical proofs of (a)–(f), we give a brief outline of these proofs and sketch some of the crucial insights. The details are available in [4].

**A proof outline for (a)–(c).** Observe that if the set of environmental states that are controlled by player $\Diamond$ is empty, then the existence of a winning strategy for player $\Box$ in $p\boldsymbol{v}$ is equivalent to the existence of a *self-covering zero-avoiding path* of the form $p\boldsymbol{v} \to^* q\boldsymbol{u} \to^+ q\boldsymbol{u}'$, where $\boldsymbol{u} \leq \boldsymbol{u}'$ and the counters stay positive along the path. The existence and the size of such paths has been studied in [28, 30] (actually, these works mainly consider the existence of an *increasing self-covering path* where $\boldsymbol{u}'$ is *strictly* larger than $\boldsymbol{u}$ in at least one component, and the counters *can* be decreased to zero in the intermediate configurations). One can easily generalize this observation to the case when the set of environmental states is non-empty and show that the existence of a winning strategy for player $\Box$ in $p\boldsymbol{v}$ is equivalent to the existence of a *self-covering zero-avoiding tree* initiated in $p\boldsymbol{v}$, which is a finite tree, rooted in $p\boldsymbol{v}$, describing a strategy for player $\Box$ where each maximal path is self-covering and zero-avoiding.

We show that the existence of a self-covering zero-avoiding tree initiated in a given configuration of a given extended VASS is decidable, and we give some complexity bounds. Let us note that this result is more subtle than it might seem; one can easily show that the existence of a self-covering (but not necessarily zero-avoiding) tree for a given configuration is already *undecidable*. Our algorithm constructs all *minimal* $p\boldsymbol{v}$ (w.r.t. component-wise ordering) where player $\Box$ has a winning strategy. Since this set is necessarily finite, and the winning region of player $\Box$ is obviously upwards-closed, we obtain a finite description of the winning region for player $\Box$. The algorithm can be viewed as a concrete (but not obvious) instance of a general approach, which is dealt with, e.g., in [33, 10, 11]. First, we compute all control states $p$ such that player $\Box$ can win in *some* configuration $p\boldsymbol{v}$. Here, a crucial step is to observe that if this is *not* the case, i.e., player $\Diamond$ can win in every $p\boldsymbol{v}$, then player $\Diamond$ has a *counterless* winning strategy which depends only on the current control state (since there are only finitely many counterless strategies, they can be tried out one by one). This computation also gives an initial bound $B$ such that for every control state $p$ we have that if player $\Box$ wins in *some* $p\boldsymbol{v}$, then he wins in all $p\boldsymbol{v}'$ where $\boldsymbol{v}'_\ell \geq B$ for all indexes (counters) $\ell \in \{1, 2, \ldots, k\}$. Then the algorithm proceeds inductively, explores the situations where at least one counter is less than $B$, computes (bigger) general bounds for the other $k-1$ counters, etc.

A finite description of a strategy for player $\Box$ which is winning in every configuration of his winning region is obtained by specifying the moves in all minimal winning configurations (observe that in a non-minimal winning configuration $p(\boldsymbol{v}+\boldsymbol{u})$ such that $p\boldsymbol{v}$ is minimal, player $\Box$ can safely make a move $p(\boldsymbol{v}+\boldsymbol{u}) \to q(\boldsymbol{v}'+\boldsymbol{u})$ where $p\boldsymbol{v} \to q\boldsymbol{v}'$ is the move associated to $p\boldsymbol{v}$). Note that this also resolves the issue with $\omega$ components in transitions performed by player $\Box$. Since the number of minimal winning configurations is finite, there is a finite and effectively computable constant $c$ such that player $\Box$ never needs to increase a counter by more than $c$ when performing a transition whose label contains a symbolic component (and we can even give a simple "recipe" which gives an optimal choice for the $\omega$ values for every configuration separately).

The winning region of player $\Diamond$ is just the complement of the winning region of player $\Box$. Computing a finite description of a winning strategy for player $\Diamond$ is somewhat trickier and relies on some observations made in the "inductive step" discussed above (note that for player $\Diamond$ it is not sufficient to stay in his winning region; he also needs to make some progress in approaching zero in some counter).

**A proof outline for (d).** The undecidability result for 2-dimensional VASS games is obtained by a straightforward reduction from the halting problem for Minsky machines with two counters initialized to zero, which is undecidable [26] (let us note that this construction is essentially the same as the one for monotonic games presented in [1]). After some minor modifications, the same construction can be also used to establish the undecidability of other natural problems for VASS and extended VASS games, such as boundedness or coverability. The high undecidability result for 3-dimensional extended VASS games is proven by reducing the problem whether a given nondeterministic Minsky machine with two counters initialized to zero has an infinite computation such that the initial instruction is executed infinitely often (this problem is known to be $\Sigma_1^1$-complete [15]). This reduction is also straightforward, but at least it demonstrates that symbolic transitions do bring some extra power (note that for "ordinary" VASS games, a winning strategy for player $\Diamond$ in a given $p\boldsymbol{v}$ can be written as a finite tree, and hence the existence of such a strategy is semidecidable).

**A proof outline for (e)–(f).** The case of one-dimensional extended VASS games with zero-reachability objectives is, of course, simpler than the general case, but our results still require some effort. In the case of $Z$-reachability objectives, we show that the winning region of player $\Diamond$ can be computed as the least fixed point of a monotonic function over a finite lattice. Although the lattice has exponentially many elements, we show that the function reaches the least fixed point only after a quadratic number of iterations. The existence and efficient constructibility of counterless winning strategies is immediate for player $\Box$, and we show that the same is achievable for player $\Diamond$. The results about $Z_C$-reachability objectives are obtained by applying known results about the emptiness problem for alternating finite automata with one letter alphabet [16] (see also [21]) and the emptiness problem for alternating two-way parity word automata [31], together with some additional observations.

**Related work.** As already mentioned, some of our results and proof techniques use and generalize the ones from [28, 30]. VASS games can be also seen as a special case of *monotonic* games considered in [1], where it is shown that the problem of deciding the winner in monotonic games with reachability objectives is undecidable (see the proof outline for (d) above). Let us note that the results presented in [1] mainly concern the so-called *downward-closed* games, which is a model different from ours. Let us also mention that (extended) VASS games are different from another recently studied model of *branching vector addition systems* [34, 6] which has different semantics and different algorithmic properties (for example, the coverability and boundedness problems for branching vector addition systems are complete for **2-EXPTIME** [6]). Generic procedures applicable to upward-closed sets of states are studied in, e.g., [3, 12, 33, 10, 11].

Note that one-dimensional VASS games are essentially one-counter automata where the counter cannot be tested for zero explicitly (that is, there are no transitions enabled only when the counter reaches zero). Such one-counter automata are also called *one-counter nets* because they correspond to Petri nets with just one unbounded place. The models of one-counter automata and one-counter nets have been intensively studied [18, 20, 22, 2, 7, 9, 19, 31, 14]. Many problems about equivalence-checking and model-checking one-counter automata are known to be decidable, but only a few of them are solvable efficiently. From this point of view, we find the polynomial-time result about one-dimensional extended VASS games with $Z$-reachability objectives encouraging.

## 2 Definitions

In this paper, the sets of all integers, positive integers, and non-negative integers are denoted by $\mathbb{Z}$, $\mathbb{N}^{>0}$, and $\mathbb{N}$, respectively. For every finite or countably infinite set $M$, the symbol $M^*$ denotes the set of all finite words (i.e., finite sequences) over $M$. The length of a given word $w$ is denoted by $|w|$, and the individual letters in $w$ are denoted by $w(0), w(1), \ldots$. The empty word is denoted by $\varepsilon$, where $|\varepsilon| = 0$. We also use $M^+$ to denote the set $M^* \smallsetminus \{\varepsilon\}$. A *path* in $\mathcal{M} = (M, \rightarrow)$, for a binary relation $\rightarrow \subseteq M \times M$, is a finite or infinite sequence $w = w(0), w(1), \ldots$ such that $w(i) \rightarrow w(i{+}1)$ for every $i$. A given $n \in M$ is *reachable* from a given $m \in M$, written $m \rightarrow^* n$, if there is a finite path from $m$ to $n$. A *run* is a maximal path (infinite, or finite which cannot be prolonged). The sets of all finite paths and all runs in $\mathcal{M}$ are denoted by $FPath(\mathcal{M})$ and $Run(\mathcal{M})$, respectively. Similarly, the sets of all finite paths and runs that start in a given $m \in M$ are denoted by $FPath(\mathcal{M}, m)$ and $Run(\mathcal{M}, m)$, respectively.

**Definition 1.** *A* game *is a tuple* $G = (V, \mapsto, (V_\Box, V_\Diamond))$ *where* $V$ *is a finite or countably infinite set of* vertices*,* $\mapsto \subseteq V \times V$ *is an* edge relation*, and* $(V_\Box, V_\Diamond)$ *is a partition of* $V$*.*

A game is played by two players, $\Box$ and $\Diamond$, who select the moves in the vertices of $V_\Box$ and $V_\Diamond$, respectively. Let $\odot \in \{\Box, \Diamond\}$. A *strategy* for player $\odot$ is a (partial) function which to each $wv \in V^* V_\odot$, where $v$ has at least one outgoing edge, assigns a vertex $v'$ such that $v \mapsto v'$. The set of all strategies for player $\Box$ and player $\Diamond$ is denoted by $\Sigma$

and $\Pi$, respectively. We say that a strategy $\tau$ is *memoryless* if $\tau(wv)$ depends just on the last vertex $v$. In the rest of this paper, we consider memoryless strategies as (partial) functions from $V_\odot$ to $V$.

A *winning objective* is a set of runs $\mathcal{W} \subseteq Run(G)$. Every pair of strategies $(\sigma, \pi) \in \Sigma \times \Pi$ and every initial vertex $v \in V$ determine a unique run $G^{(\sigma,\pi)}(v) \in Run(G, v)$ which is called a *play*. We say that a strategy $\sigma \in \Sigma$ is *$\mathcal{W}$-winning for player $\Box$* in a given $v \in V$ if for every $\pi \in \Pi$ we have that $G^{(\sigma,\pi)}(v) \in \mathcal{W}$. Similarly, a strategy $\pi \in \Pi$ is $\mathcal{W}$-winning for player $\Diamond$ if for every $\sigma \in \Sigma$ we have that $G^{(\sigma,\pi)}(v) \in \mathcal{W}$. The set of all vertices where player $\odot$ has a $\mathcal{W}$-winning strategy is called the *winning region* of player $\odot$ and denoted by $Win(\odot, \mathcal{W})$.

In this paper, we only consider *reachability* and *safety* objectives, which are specified by a subset of target vertices that should or should not be reached by a run, respectively. Formally, for a given $T \subseteq V$ we define the sets of runs $\mathcal{R}(T)$ and $\mathcal{S}(T)$, where $\mathcal{R}(T) = \{w \in Run(G) \mid w(i) \in T \text{ for some } i\}$, and $\mathcal{S}(T) = Run(G) \setminus \mathcal{R}(T)$. We note that the games with reachability and safety objectives are *determined* [25], i.e., $Win(\Box, \mathcal{S}(T)) = V \setminus Win(\Diamond, \mathcal{R}(T))$, and each player has a *memoryless* winning strategy in every vertex of his winning region.

**Definition 2.** *Let $k \in \mathbb{N}^{>0}$. A k-dimensional* vector addition system with states (VASS) *is a tuple $\mathcal{M} = (Q, T, \alpha, \beta, \delta)$ where $Q \neq \emptyset$ is a finite set of* control states*, $T \neq \emptyset$ is a finite set of* transitions*, $\alpha : T \to Q$ and $\beta : T \to Q$ are the* source *and* target *mappings, and $\delta : T \to \{-1, 0, 1\}^k$ is a* transition displacement *labeling. For technical convenience, we assume that for every $q \in Q$ there is some $t \in T$ such that $\alpha(t) = q$.*

*An* extended VASS (eVASS for short) *is a VASS where the transition displacement labeling is a function $\delta : T \to \{-1, 0, 1, \omega\}^k$.*

*A* VASS game *(or* eVASS game*) is a tuple $\mathcal{M} = (Q, (Q_\Box, Q_\Diamond), T, \alpha, \beta, \delta)$ where $(Q, T, \alpha, \beta, \delta)$ is a VASS (or eVASS) and $(Q_\Box, Q_\Diamond)$ is a partition of $Q$.*

A *configuration* of $\mathcal{M}$ is an element of $Q \times \mathbb{N}^k$. We write $p\boldsymbol{v}$ instead of $(p, \boldsymbol{v})$, and the $\ell$-th component of $\boldsymbol{v}$ is denoted by $\boldsymbol{v}_\ell$. For a given transition $t \in T$, we write $t : p \to q$ to indicate that $\alpha(t) = p$ and $\beta(t) = q$, and $p \xrightarrow{\boldsymbol{v}} q$ to indicate that $p \to q$ and $\delta(t) = \boldsymbol{v}$. A transition $t \in T$ is *enabled* in a configuration $p\boldsymbol{v}$ if $\alpha(t) = p$ and for every $1 \leq \ell \leq k$ such that $\delta(t)_\ell = -1$ we have $\boldsymbol{v}_\ell \geq 1$.

Every $k$-dimensional eVASS game $\mathcal{M} = (Q, (Q_\Box, Q_\Diamond), T, \alpha, \beta, \delta)$ induces a unique infinite-state game $G_\mathcal{M}$ where $Q \times \mathbb{N}^k$ is the set of vertices partitioned into $Q_\Box \times \mathbb{N}^k$ and $Q_\Diamond \times \mathbb{N}^k$, and $p\boldsymbol{v} \mapsto q\boldsymbol{u}$ iff the following condition holds: There is a transition $t \in T$ enabled in $p\boldsymbol{v}$ such that $\beta(t) = q$ and for every $1 \leq \ell \leq k$ we have that $\boldsymbol{u}_\ell - \boldsymbol{v}_\ell$ is either non-negative or equal to $\delta(t)_\ell$, depending on whether $\delta(t)_\ell = \omega$ or not, respectively. Note that any play can get stuck only when a counter is zero, because there is at least one enabled transition otherwise.

In this paper, we are interested in VASS and eVASS games with *non-selective* and *selective* zero-reachability objectives. Formally, for every $C \subseteq Q$ we define the set $Z_C = \{p\boldsymbol{v} \in Q \times \mathbb{N}^k \mid p \in C \text{ and } \boldsymbol{v}_i = 0 \text{ for some } 0 \leq i \leq k\}$ and we also put $Z = Z_Q$. Selective (or non-selective) zero-reachability objectives are reachability objectives where the set $T$ of target configurations is equal to $Z_C$ for some $C \subseteq Q$ (or to $Z$, respectively).

As we have already noted, our games with reachability objectives are memoryless determined and this result of course applies also to eVASS games with zero-reachability objectives. However, since eVASS games have infinitely many vertices, not all memoryless strategies are finitely representable. In this paper we will often deal with a simple form of memoryless strategies, where the decision is independent of the current counter values; such strategies are called *counterless strategies*.

**Definition 3.** *Given an eVASS* $\mathcal{M} = (Q, (Q_\Box, Q_\Diamond), T, \alpha, \beta, \delta)$*, a* strategy $\tau$ *of player* $\odot \in \{\Box, \Diamond\}$ *is* counterless *if it determines a (fixed) transition* $t_p$ *for each* $p \in Q_\odot$*, together with (fixed) values* $c_{p,\ell} \in \mathbb{N}$ *for all those* $\ell$ *for which* $\delta(t_p)_\ell = \omega$*, so that* $\tau(p\boldsymbol{v})$ *is the configuration obtained by performing* $t_p$ *in* $p\boldsymbol{v}$ *where* $\omega$*'s are instantiated with* $c_{p,\ell}$*.*

## 3 VASS and eVASS games with zero-reachability objectives

In this section, we analyze VASS and eVASS games with zero-reachability objectives (full proofs can be found in [4]). We first note that the problems of our interest are undecidable for $\mathcal{R}(Z_C)$ objectives; this can be shown by standard techniques.

**Proposition 4.** *The problem of deciding the winner in* 2*-dimensional VASS games with* $\mathcal{R}(Z_C)$ *objectives is undecidable. For* 3*-dimensional eVASS games, the same problem is* highly *undecidable (i.e., beyond the arithmetical hierarchy).*

Let us note that Proposition 4 does not hold for one-dimensional eVASS games, which are analyzed later in this section. Further, by some trivial modifications of the proof of Proposition 4 we also get the undecidability of the boundedness/coverability problems for 2-dimensional VASS games.

Now we turn our attention to $\mathcal{R}(Z)$ objectives. For the rest of this section, we fix a $k$-dimensional eVASS game $\mathcal{M} = (Q, (Q_\Box, Q_\Diamond), T, \alpha, \beta, \delta)$. Since we are interested only in $\mathcal{R}(Z)$ objectives, we may safely assume that every transition $p \xrightarrow{\boldsymbol{v}} q$ of $\mathcal{M}$ where $p \in Q_\Diamond$ satisfies $\boldsymbol{v}_\ell \neq \omega$ for every $1 \leq \ell \leq k$ (if there are some $\omega$-components in $\boldsymbol{v}$, they can be safely replaced with 0). We also use $d$ to denote the *branching degree* of $\mathcal{M}$, i.e, the least number such that every $q \in Q$ has at most $d$ outgoing transitions.

Let $\leq$ be the partial order on the set of configurations of $\mathcal{M}$ defined by $p\boldsymbol{u} \leq q\boldsymbol{v}$ iff $p = q$ and $\boldsymbol{u} \leq \boldsymbol{v}$ (componentwise). For short, we write $Win_\Diamond$ instead of $Win(\Diamond, \mathcal{R}(Z))$ and $Win_\Box$ instead of $Win(\Box, \mathcal{S}(Z))$. Obviously, if player $\Diamond$ has a winning strategy in $q\boldsymbol{v}$, then he can use "essentially the same" strategy in $q\boldsymbol{u}$ for every $\boldsymbol{u} \leq \boldsymbol{v}$ (behaving in $q'\boldsymbol{v}'$ as previously in $q'(\boldsymbol{v}' + \boldsymbol{v} - \boldsymbol{u})$, which results in reaching 0 in some counter possibly even earlier). Similarly, if $q\boldsymbol{v} \in Win_\Box$ then $q\boldsymbol{u} \in Win_\Box$ for every $\boldsymbol{u} \geq \boldsymbol{v}$. Hence, the sets $Win_\Diamond$ and $Win_\Box$ are downwards closed and upwards closed w.r.t. $\leq$, respectively. This means that the set $Win_\Box$ is finitely representable by its subset $Min_\Box$ of *minimal elements* (note that $Min_\Box$ is necessarily finite because there is no infinite subset of $\mathbb{N}^k$ with pairwise incomparable elements, as Dickson's Lemma shows). Technically, it is convenient to consider also *symbolic configurations* of $\mathcal{M}$ which are introduced in the next definition.

**Definition 5.** *A* symbolic configuration *is a pair* $q\boldsymbol{v}$ *where* $q \in Q$ *and* $\boldsymbol{v} \in (\mathbb{N} \cup \{\omega\})^k$*. We say that a given index* $\ell \in \{1, 2, \ldots, k\}$ *is* precise *in* $q\boldsymbol{v}$ *if* $\boldsymbol{v}_\ell \in \mathbb{N}$*, otherwise it is* symbolic *in* $q\boldsymbol{v}$*. The* precision *of* $q\boldsymbol{v}$*, denoted by* $P(q\boldsymbol{v})$*, is the number of indexes that are precise in* $q\boldsymbol{v}$*. We say that a configuration* $p\boldsymbol{u}$ matches *a symbolic configuration* $q\boldsymbol{v}$ *if* $p = q$ *and* $\boldsymbol{u}_\ell = \boldsymbol{v}_\ell$ *for every* $\ell$ *precise in* $q\boldsymbol{v}$*. Similarly, we say that* $p\boldsymbol{u}$ matches $q\boldsymbol{v}$ above *a given bound* $B \in \mathbb{N}$ *if* $p\boldsymbol{u}$ *matches* $q\boldsymbol{v}$ *and* $\boldsymbol{u}_\ell \geq B$ *for every* $\ell$ *symbolic in* $q\boldsymbol{v}$*.*

We extend the set $Win_\Box$ by all symbolic configurations $q\boldsymbol{v}$ such that some configuration matching $q\boldsymbol{v}$ belongs to $Win_\Box$. Similarly, the set $Win_\Diamond$ is extended by all symbolic configurations $q\boldsymbol{v}$ such that all configurations matching $q\boldsymbol{v}$ belong to $Win_\Diamond$ (note that every symbolic configuration belongs either to $Win_\Box$ or to $Win_\Diamond$). We also extend the previously fixed ordering on configurations to symbolic configurations by stipulating that $\omega \leq \omega$ and $n < \omega$ for all $n \in \mathbb{N}$. Obviously, this extension does not influence the set $Min_\Box$, and the winning region $Win_\Diamond$ can be now represented by its subset $Max_\Diamond$ of all maximal elements, which is necessarily finite.

Our ultimate goal is to compute the sets $Min_\Box$ and $Max_\Diamond$. Since our reachability games are determined, it actually suffices to compute just one of these sets. In the following we show how to compute $Min_\Box$.

We start with an important observation about winning strategies for player $\Box$, which in fact extends the "classical" observation about self-covering paths in vector addition systems presented in [28]. Let $q \in Q$ be such that $q\boldsymbol{v} \in Win_\Box$ for some $\boldsymbol{v}$, i.e., $q(\omega,\ldots,\omega) \in Win_\Box$. This means that there is a strategy of player $\Box$ that prevents unbounded decreasing of the counters; we find useful to represent the strategy by a finite *unrestricted self-covering tree for q*. The word "unrestricted" reflects the fact that we also consider configurations with negative and symbolic counter values. More precisely, an unrestricted self-covering tree for $q$ is a finite tree $\mathcal{T}$ whose nodes are labeled by the elements of $Q \times (\mathbb{Z} \cup \{\omega\})^k$ satisfying the following ($\omega$ is treated in the standard way, i.e., $\omega + \omega = \omega + c = \omega$ for every $c \in \mathbb{Z}$):

- The root of $\mathcal{T}$ is labeled by $q(0,\ldots,0)$.
- If $n$ is a non-leaf node of $\mathcal{T}$ labeled by $p\boldsymbol{u}$, then
  - if $p \in Q_\Box$, then $n$ has only one successor labeled by some $r\boldsymbol{t}$ such that $\mathcal{M}$ has a transition $p \xrightarrow{\boldsymbol{v}} r$ where $\boldsymbol{t} = \boldsymbol{u} + \boldsymbol{v}$;
  - if $p \in Q_\Diamond$, then there is a one-to-one correspondence between the successors of $n$ and transitions of $\mathcal{M}$ of the form $p \xrightarrow{\boldsymbol{v}} r$. The node which corresponds to a transition $p \xrightarrow{\boldsymbol{v}} r$ is labeled by $r\boldsymbol{t}$ where $\boldsymbol{t} = \boldsymbol{u} + \boldsymbol{v}$.
- If $n$ is a leaf of $\mathcal{T}$ labeled by $p\boldsymbol{u}$, then there is another node $m$ (where $m \neq n$) on the path from the root of $\mathcal{T}$ to $n$ which is labeled by $p\boldsymbol{t}$ for some $\boldsymbol{t} \leq \boldsymbol{u}$.

The next lemma bounds the depth of such a tree.

**Lemma 6.** *Let $q(\omega,\ldots,\omega) \in Win_\Box$ (i.e., $q\boldsymbol{v} \in Win_\Box$ for some $\boldsymbol{v}$). Then there is an unrestricted self-covering tree for q of depth at most $f(|Q|,d,k) = 2^{(d-1)\cdot|Q|} \cdot |Q|^{c\cdot k^2}$, where c is a fixed constant independent of $\mathcal{M}$, and d is the branching degree of $\mathcal{M}$.*

Lemma 6 thus implies that if $q(\omega,\ldots,\omega) \in Win_\Box$, then $q\boldsymbol{u} \in Win_\Box$ for all $\boldsymbol{u}$ with $\boldsymbol{u}_\ell \geq f(|Q|,d,k)$ for all $\ell \in \{1,2,\ldots,k\}$ (recall that each counter can be decreased at most by one in a single transition). The next lemma shows that we can compute the set of all $q \in Q$ such that $q(\omega,\ldots,\omega) \in Win_\Box$ (the lemma is formulated "dually", i.e., for player $\Diamond$).

**Lemma 7.** *The set of all $q \in Q$ such that $q(\omega,\ldots,\omega) \in Win_\Diamond$ is computable in space bounded by $g(|Q|,d,k)$, where g is a polynomial in three variables.*

An important observation, which is crucial in our proof of Lemma 7 and perhaps interesting on its own, is that if $q(\omega, \ldots, \omega) \in Win_{\Diamond}$, then player $\Diamond$ has a *counterless* strategy which is winning in every configuration matching $q(\omega, \ldots, \omega)$.

To sum up, we can compute the set of all $q(\omega, \ldots, \omega) \in Win_{\Box}$ and a bound $B$ which is "safe" for all $q(\omega, \ldots, \omega) \in Win_{\Box}$ in the sense that all configurations matching $q(\omega, \ldots, \omega)$ above $B$ belong to $Win_{\Box}$. Intuitively, the next step is to find out what happens if one of the counters, say the first one, stays bounded by $B$. Obviously, there is the *least* $j \leq B$ such that $q(j, \omega, \ldots, \omega) \in Win_{\Box}$, and there is a bound $D > B$ such that all configurations matching $q(j, \omega, \ldots, \omega)$ above $D$ belong to $Win_{\Box}$. If we manage to compute the minimal $j$ (also for the other counters, not just for the first one) and the bound $D$, we can go on and try to bound *two* counters simultaneously by $D$, find the corresponding minima, and construct a new "safe" bound. In this way, we eventually bound all counters and compute the set $Min_{\Box}$. In our next definition, we introduce some notions that are needed to formulate the above intuition precisely. (Recall that $P(q\boldsymbol{v})$ gives the number of precise, i.e. non-$\omega$, elements of $\boldsymbol{v}$.)

**Definition 8.** *For a given $0 \leq j \leq k$, let $SymMin^j_{\Box}$ be the set of all minimal $q\boldsymbol{v} \in Win_{\Box}$ such that $P(q\boldsymbol{v}) = j$. Further, let $SymMin_{\Box} = \bigcup_{i=0}^{k} SymMin^i_{\Box}$. We say that a given $B \in \mathbb{N}$ is* safe for precision *$j$, where $0 \leq j \leq k$, if for every $q\boldsymbol{v} \in \bigcup_{i=0}^{j} SymMin^i_{\Box}$ we have that $\boldsymbol{v}_\ell \leq B$ for every precise index $\ell$ in $\boldsymbol{v}$, and every configuration matching $q\boldsymbol{v}$ above $B$ belongs to $Win_{\Box}$.*

Obviously, every $SymMin^j_{\Box}$ (and hence also $SymMin_{\Box}$) is finite, and $Min_{\Box} = SymMin^k_{\Box}$. Also observe that $SymMin^0_{\Box}$ is computable in time exponential in $|Q|$ and $k$ by Lemma 7, and a bound which is safe for precision 0 is computable in polynomial time by Lemma 6. Now we design an algorithm which computes $SymMin^{j+1}_{\Box}$ and a bound safe for precision $j{+}1$, assuming that $SymMin^i_{\Box}$ for all $i \leq j$ and a bound safe for precision $j$ have already been computed.

**Lemma 9.** *Let $0 \leq j < k$, and let us assume that $\bigcup_{i=0}^{j} SymMin^i_{\Box}$ has already been computed, together with some bound $B \in \mathbb{N}$ which is safe for precision $j$. Then $SymMin^{j+1}_{\Box}$ is computable in time exponential in $|Q| \cdot B^{j+1}$, $d$, and $k{-}j{-}1$, and the bound $B + f(|Q| \cdot B^{j+1}, d, k{-}j{-}1)$ is safe for precision $j + 1$ (here $f$ is the function of Lemma 6 and $d$ is the branching degree of $\mathcal{M}$).*

Now we can easily evaluate the total complexity of computing $SymMin_{\Box}$ (and hence also $Min_{\Box}$). If we just examine the recurrence of Lemma 9, we obtain that the set $SymMin_{\Box}$ is computable in $k$-exponential time. However, we can actually decrease the height of the tower of exponentials by one when we incorporate the results presented later in this section, which imply that for one-dimensional eVASS games, the depth of an unrestricted self-covering tree can be bounded by a *polynomial* in $|Q|$ and $d$, and the set of all $q \in Q$ where $q(\omega) \in Win_{\Diamond}$ is computable in *polynomial time*. Hence, we actually need to "nest" Lemma 9 only $k{-}1$ times. Thus, we obtain the following (where 0-exponential time denotes polynomial time):

**Theorem 10.** *For a given $k$-dimensional eVASS, the set $Min_{\Box}$ is computable in $(k{-}1)$-exponential time.*

Let us note a substantial improvement in complexity would be achieved by improving the bound presented in Lemma 6. Actually, it is not so important what is the depth of an unrestricted self-covering tree, but what are the minimal numbers that allow for applying the strategy described by this tree without reaching zero (i.e., what is the maximal decrease of a counter in the tree). A more detailed complexity analysis based on the introduced parameters reveals that if the maximal counter decrease was just polynomial in the number of control states (which is our conjecture), the complexity bound of Theorem 10 would be *polynomial* for every fixed dimension $k$ (see also Section 4).

Note that after computing the set $Min_\Box$, we can easily compute a finite description of a strategy $\sigma$ for player $\Box$ which is winning in every configuration of $Win_\Box$. For every $p\boldsymbol{v} \in Min_\Box$ such that $p \in Q_\Box$, we put $\sigma(p\boldsymbol{v}) = q\boldsymbol{v}'$, where $q\boldsymbol{v}'$ is (some) configuration such that $q\boldsymbol{v}' \geq q\boldsymbol{t}$ for some $q\boldsymbol{t} \in Min_\Box$. Note that there must be at least one such $q\boldsymbol{v}'$ and it can be computed effectively. For every configuration $p\boldsymbol{u}$ such that $p\boldsymbol{u} \geq p\boldsymbol{v}$ for some $p\boldsymbol{v} \in Min_\Box$, we put $\sigma(p\boldsymbol{u}) = q(\boldsymbol{v}'+\boldsymbol{u}-\boldsymbol{v})$ where $\sigma(p\boldsymbol{v}) = q\boldsymbol{v}'$ (if there are more candidates for $p\boldsymbol{v}$, any of them can be chosen). It is easy to see that $\sigma$ is winning in every configuration of $Win_\Box$. Also observe that if we aim at constructing a winning strategy for player $\Box$ which minimizes the concrete numbers used to substitute $\omega$'s, we can use $Min_\Box$ to construct an "optimal" choice of the values which are sufficient (and necessary) to stay in the winning region of player $\Box$.

Now we present the promised results about the special case of one-dimensional VASS and eVASS games with zero-reachability objectives. Let us fix a one-dimensional eVASS game $\mathcal{M} = (Q, (Q_\Box, Q_\Diamond), T, \alpha, \beta, \delta)$ and $C \subseteq Q$. For every $i \in \mathbb{N}$, let $Win_\Diamond(C, i) = \{p \in Q \mid p(i) \in Win(\Diamond, \mathcal{R}(Z_C))\}$. It is easy to see that if $Win_\Diamond(C, i) = Win_\Diamond(C, j)$ for some $i, j \in \mathbb{N}$, then also $Win_\Diamond(C, i{+}1) = Win_\Diamond(C, j{+}1)$. Let $m_C$ be the least $i \in \mathbb{N}$ such that $Win_\Diamond(C, i) = Win_\Diamond(C, j)$ for some $j > i$, and let $n_C$ be the least $i > 0$ such that $Win_\Diamond(C, m_C) = Win_\Diamond(C, m_C{+}i)$. Obviously, $m_c + n_c \leq 2^{|Q|}$ and for every $i \geq m_c$ we have that $Win_\Diamond(C, i) = Win_\Diamond(C, m_C + ((i - m_C) \mod n_C))$. Hence, the winning regions of both players are fully characterized by all $Win_\Diamond(C, i)$, where $0 \leq i < m_C + n_C$.

The selective subcase is analyzed in the next theorem. The **PSPACE** lower bound is obtained by reducing the emptiness problem for alternating finite automata with one letter alphabet, which is known to be **PSPACE** complete [16] (see also [21] for a simpler proof). The **PSPACE** upper bound follows by employing the result of [31] which says that the emptiness problem for alternating two-way parity word automata (2PWA) is in **PSPACE** (we would like to thank Olivier Serre for providing us with relevant references). The effective constructability of the winning strategies for player $\Box$ and player $\Diamond$ follows by applying the results on non-selective termination presented below.

**Theorem 11.** *The problem whether $p(i) \in Win(\Diamond, \mathcal{R}(Z_C))$ is **PSPACE**-complete. Further, there is a strategy $\sigma$ winning for player $\Box$ in every configuration of $Win(\Box, \mathcal{S}(Z_C))$ such that for all $p \in Q_\Box$ and $i \geq m_C$ we have that $\sigma(p(i)) = \sigma(p(m_C + ((i - m_C) \mod n_C)))$. The numbers $m_C, n_C$ and the tuple of all $Win_\Diamond(C, i)$ and $\sigma(p(i))$, where $0 \leq i < m_C{+}n_C$ and $p \in Q_\Box$, are constructible in time exponential in $|\mathcal{M}|$.*

In the non-selective subcase, the situation is even better. The winning regions for both players are monotone, which means that $m_Q \leq |Q|$ and $n_Q = 1$. Further, all of the considered problems are solvable in polynomial time.

**Theorem 12.** *The problem whether $p(i) \in Win(\Diamond, \mathcal{R}(Z))$ is in* ***P****. Further, there are counterless strategies $\sigma$ and $\pi$ such that $\sigma$ is winning for player $\Box$ in every configuration of $Win(\Box, \mathcal{S}(Z))$ and $\pi$ is winning for player $\Diamond$ in every configuration of $Win(\Diamond, \mathcal{R}(Z))$. The tuple of all $Win_\Diamond(Q, i)$, $\sigma(p)$, and $\pi(q)$, where $0 \le i \le m_C$, $p \in Q_\Box$, and $q \in Q_\Diamond$, is constructible in time polynomial in $|\mathcal{M}|$.*

## 4 Conclusions, future work

Technically, the most involved result presented in this paper is Theorem 10. This decidability result is not obvious, because most of the problems related to formal verification of Petri nets (equivalence-checking, model-checking, etc.) are undecidable [8, 17, 23, 5]. Since the upper complexity bound given in Theorem 10 is complemented only by the **EXPSPACE** lower bound, which is easily derivable from [24], there is a complexity gap which constitutes an interesting challenge for future work.

## References


1. P.A. Abdulla, A. Bouajjani, and J. d'Orso. Monotonic and downward closed games. *Journal of Logic and Computation*, 18(1):153–169, 2008.
2. P.A. Abdulla and K. Čerāns. Simulation is decidable for one-counter nets. In *Proceedings of CONCUR'98*, volume 1466 of *LNCS*, pages 253–268. Springer, 1998.
3. P.A. Abdulla, K. Čerāns, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *Proceedings of LICS'96*, pages 160–170. IEEE Computer Society Press, 1996.
4. T. Brázdil, P. Jančar, and A. Kučera. Reachability games on extended vector addition systems with states. *CoRR*, abs/1002.2557, 2010.
5. O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification on infinite structures. *Handbook of Process Algebra*, pages 545–623, 2001.
6. S. Demri, M. Jurdziński, O. Lachish, and R. Lazić. The covering and boundedness problems for branching vector addition systems. In *Proceedings of FST&TCS 2009*, volume 4 of *Leibniz International Proceedings in Informatics*, pages 181–192. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2009.
7. S. Demri, R. Lazic, and A. Sangnier. Model checking freeze LTL over one-counter automata. In *Proceedings of FoSSaCS 2008*, volume 4962 of *LNCS*. Springer, 2008.
8. J. Esparza. Decidability of model checking for infinite-state concurrent systems. *Acta Informatica*, 34:85–107, 1997.
9. K. Etessami, D. Wojtczak, and M. Yannakakis. Quasi-birth-death processes, tree-like QBDs, probabilistic 1-counter automata, and pushdown systems. In *Proceedings of 5th Int. Conf. on Quantitative Evaluation of Systems (QEST'08)*. IEEE Computer Society Press, 2008.
10. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, Part I: Completions. In *Proceedings of STACS 2009*, volume 3 of *Leibniz International Proceedings in Informatics*, pages 433–444. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2009.
11. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, Part II: Complete WSTS. In *Proceedings of ICALP 2009*, volume 5556 of *LNCS*, pages 188–199. Springer, 2009.
12. A. Finkel and Ph. Schnoebelen. Well structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.
13. E. Grädel, W. Thomas, and T. Wilke. *Automata, Logics, and Infinite Games*. Number 2500 in LNCS. Springer, 2002.

14. C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. Reachability in succinct and parametric one-counter automata. In *Proceedings of CONCUR 2009*, volume 5710 of *LNCS*, pages 369–383. Springer, 2009.
15. D. Harel. Effective transformations on infinite trees with applications to high undecidability dominoes, and fairness. *Journal of the Association for Computing Machinery*, 33(1), 1986.
16. M. Holzer. On emptiness and counting for alternating finite automata. In *Developments in Language Theory II*, pages 88–97. World Scientific, 1995.
17. P. Jančar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148(2):281–301, 1995.
18. P. Jančar. Decidability of bisimilarity for one-counter processes. *Information and Computation*, 158(1):1–17, 2000.
19. P. Jančar, A. Kučera, and F. Moller. Simulation and bisimulation over one-counter processes. In *Proceedings of STACS 2000*, volume 1770 of *LNCS*, pages 334–345. Springer, 2000.
20. P. Jančar, A. Kučera, F. Moller, and Z. Sawa. DP lower bounds for equivalence-checking and model-checking of one-counter automata. *Information and Computation*, 188(1):1–19, 2004.
21. P. Jančar and Z. Sawa. A note on emptiness for alternating finite automata with a one-letter alphabet. *Information Processing Letters*, 104(5):164–167, 2007.
22. A. Kučera. The complexity of bisimilarity-checking for one-counter processes. *Theoretical Computer Science*, 304(1–3):157–183, 2003.
23. A. Kučera and P. Jančar. Equivalence-checking on infinite-state systems: Techniques and results. *Theory and Practice of Logic Programming*, 6(3):226–264, 2006.
24. R. Lipton. The reachability problem requires exponential space. Technical report 62, Yale University, 1976.
25. D.A. Martin. Borel determinacy. *Annals of Mathematics, Second Series*, 102(2):363–371, 1975.
26. M.L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.
27. J.L. Peterson. *Petri Net Theory and the Modelling of Systems*. Prentice-Hall, 1981.
28. C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6:223–231, 1978.
29. W. Reisig. *Petri Nets—An Introduction*. Springer, 1985.
30. L.E. Rosier and H.-C. Yen. A multiparameter analysis of the boundedness problem for vector addition systems. *Journal of Computer and System Sciences*, 32:105–135, 1986.
31. O. Serre. Parity games played on transition graphs of one-counter processes. In *Proceedings of FoSSaCS 2006*, volume 3921 of *LNCS*, pages 337–351. Springer, 2006.
32. W. Thomas. Infinite games and verification. In *Proceedings of CAV 2003*, volume 2725 of *LNCS*, pages 58–64. Springer, 2003.
33. R. Valk and M. Jantzen. The residue of vector sets with applications to decidability problems in Petri nets. *Acta Informatica*, 21:643–674, 1985.
34. K.N. Verma and J. Goubault-Larrecq. Karp-Miller trees for a branching extension of VASS. *Discrete Mathematics & Theoretical Computer Science*, 7(1):217–230, 2005.
35. I. Walukiewicz. A landscape with games in the background. In *Proceedings of LICS 2004*, pages 356–366. IEEE Computer Society Press, 2004.