Long-Run Average Behavior of Vector Addition Systems with States

Krishnendu Chatterjee Thomas A. Henzinger Jan Otop May 15, 2019

Abstract

A vector addition system with states (VASS) consists of a finite set of states and counters. A configuration is a state and a value for each counter; a transition changes the state and each counter is incremented, decremented, or left unchanged. While qualitative properties such as state and configuration reachability have been studied for VASS, we consider the long-run average cost of infinite computations of VASS. The cost of a configuration is for each state, a linear combination of the counter values. In the special case of uniform cost functions, the linear combination is the same for all states. The (regular) long-run emptiness problem is, given a VASS, a cost function, and a threshold value, if there is a (lasso-shaped) computation such that the long-run average value of the cost function does not exceed the threshold. For uniform cost functions, we show that the regular long-run emptiness problem is (a) decidable in polynomial time for integer-valued VASS, and (b) decidable but nonelementarily hard for natural-valued VASS (i.e., nonnegative counters). For general cost functions, we show that the problem is (c) NP-complete for integer-valued VASS, and (d) undecidable for natural-valued VASS. Our most interesting result is for (c) integer-valued VASS with general cost functions, where we establish a connection between the regular long-run emptiness problem and quadratic Diophantine inequalities. The general (nonregular) long-run emptiness problem is equally hard as the regular problem in all cases except (c), where it remains open.

1 Introduction

Vector Addition System with States (VASS). Vector Addition Systems (VASs) [22] are equivalent to Petri Nets, and they provide a fundamental framework for formal analysis of parallel processes [16]. The extension of VASs with a finite-state transition structure gives Vector Addition Systems with States (VASS). Informally, a VASS consists of a finite set of control states and transitions between them, and a set of k counters, where at every transition between the control states each counter is either incremented, decremented, or remains unchanged. A configuration consists of a control state and a valuation of each counter, and the transitions of the VASS describe the transitions between the configurations. Thus a VASS represents a finite description of an infinite-state transition system between the configurations. If the counters can hold all possible integer values, then we call them integer-valued VASS; and if the counters can hold only non-negative values, then we call them natural-valued VASS.

Modelling power. VASS are a fundamental model for concurrent processes [16], and often used in performance analysis of concurrent processes [14, 18, 20, 21]. Moreover, VASS have been used (a) in analysis of parametrized systems [2], (b) as abstract models for programs for bounds and amortized analysis [31], (c) in interactions between components of an API in component-based synthesis [17]. Thus they provide a rich framework for a variety of problems in verification as well as program analysis.

Previous results for VASS. A computation (or a run) in a VASS is a sequence of configurations. The well-studied problems for VASS are as follows: (a) control-state reachability where given a set of target control states a computation is successful if one of the target states is reached; (b) configuration reachability where given a set of target configurations a computation is successful if one of the target configurations is reached.

For natural-valued VASS, (a) the control-state reachability problem is ExpSpace-complete: the ExpSpace-hardness is shown in [27, 15] and the upper bound follows from [30]; and (b) the configuration reachability problem is decidable [28, 23, 24, 25], and a recent breakthrough result shows that the problem is non-elementary hard [12]. For integer-valued VASS, (a) the control-state reachability problem is NLogSpace-complete: the counters can be abstracted away and we have to solve the reachability problem for graphs; and (b) the configuration reachability problem is NP-complete: this is a folklore result obtained via reduction to linear Diophantine inequalities.

Long-run average property. The classical problems for VASS are qualitative (or Boolean) properties where each computation is either successful or not. In this work we consider long-run average property that assigns a real value to each computation. A cost is associated to each configuration and the value of a computation is the long-run average of the costs of the configurations of the computation. For cost assignment to configuration we consider linear combination with natural coefficients of the values of the counters. In general the linear combination for the cost depends on the control state of the configuration, and in the special case of uniform cost functions the linear combination is the same across all states.

Motivating examples. We present some motivating examples for the problems we consider. First, consider a VASS where the counters represent different queue lengths, and each queue consumes resource (e.g., energy) proportional to its length, however, for different queues the constant of the proportionality might differ. The computation of the long-run average resource consumption is modeled as a uniform cost function, where the linear combination for the cost function is obtained from the constants of proportionality. Second, consider a system that uses two different batteries, and the counters represent the charge levels. At different states, different batteries are used, and we are interested in the long-run average charge of the used battery. This is modeled as a general cost function, where depending on the control state the cost function is the value of the counter representing the battery used in the state.

Our contributions. We consider the following decision problem: given a VASS and a cost function decide whether there is a regular (or periodic) computation such that the long-run average value is at most a given threshold. Our main contributions are as follows:

- 1. For uniform cost functions, we show that the problem is (a) decidable in polynomial time for integer-valued VASS, and (b) decidable but non-elementary hard for natural-valued VASS. In (b) we assume that the cost function depends on all counters.
- 2. For general cost functions, we show that the problem is (a) NP-complete for integer-valued VASS, and (b) undecidable for natural-valued VASS.

Our most interesting result is for general cost functions and integer-valued VASS, where we establish an interesting connection between the problem we consider and quadratic Diophantine inequalities. Finally, instead of regular computations, if we consider existence of an arbitrary computation, then all the above results hold, other than the NP-completeness result for general cost functions and integer-valued VASS (which remains open).

Related works. Long-run average behavior have been considered for probabilistic VASS [6], and other infinite-state models such as pushdown automata and games [10, 11, 1]. In these works the costs are associated with transitions of the underlying finite-state representation. In contrast, in our work the costs depend on the counter values and thus on the configurations. Costs based on configurations, specifically the content of the stack in pushdown automata, have been considered in [29]. Quantitative asymptotic bounds for polynomial-time termination in VASS have also been studied [4, 26], however, these works do not consider long-run average property. Finally, a related model of automata with monitor counters with long-run average property have been considered in [9, 8]. However, there are some crucial differences: in automata with monitor counters, the cost always depends on one counter, and counters are reset once the value is used. Moreover, the complexity results for automata with monitor counters are quite different from the results we establish

The full proofs of presented theorems and lemmas have been relegated to the appendix.

2 Preliminaries

For a sequence w, we define w[i] as the (i+1)-th element of w (we start with 0) and w[i,j] as the subsequence $w[i]w[i+1]\dots w[j]$. We allow j to be ∞ for infinite sequences. For a finite sequence w, we denote by |w| its length; and for an infinite sequence the length is ∞ .

We use the same notation for vectors. For a vector $\vec{x} \in \mathbb{R}^k$ (resp., \mathbb{Q}^k , \mathbb{Z}^k or \mathbb{N}^k), we define x[i] as the i-th component of \vec{x} . We define the support of \vec{x} , denoted by $supp(\vec{x})$ as the set of components of \vec{x} with non-zero values. For vectors \vec{x}, \vec{y} of equal dimension, we denote by $\vec{x} \cdot \vec{y}$, the dot-product of \vec{x} and \vec{y} .

2.1 Vector addition systems with states (VASS)

A k-dimensional vector addition system with states (VASS) over \mathbb{Z} (resp., over \mathbb{N}), referred to as VASS(\mathbb{Z} , k) (resp., VASS(\mathbb{N} , k)), is a tuple $\mathcal{A} = \langle Q, Q_0, \delta \rangle$, where (1) Q is a finite set of states, (2) $Q_0 \subseteq Q$ is a set of initial states, and (3) $\delta \subseteq Q \times Q \times \mathbb{Z}^k$. We denote by VASS(\mathbb{Z} , k) (resp., VASS(\mathbb{N} , k)) the class of k-dimensional VASS over \mathbb{Z} (resp., \mathbb{N}). We often omit the dimension in VASS and write VASS(\mathbb{Z}), VASS(\mathbb{N}), VASS(\mathbb{N}), VASS(\mathbb{Z}) if a definition or an argument is uniform w.r.t. the dimension. Configurations and computations. A configuration of a VASS(\mathbb{Z} , k) A is a pair from $Q \times \mathbb{Z}^k$, which

Configurations and computations. A configuration of a VASS(\mathbb{Z}, k) \mathcal{A} is a pair from $Q \times \mathbb{Z}^k$, which consists of a state and a valuation of the counters. A computation of \mathcal{A} is an infinite sequence π of configurations such that (a) $\pi[0] \in Q_0 \times \{\vec{0}\}^1$, and (b) for every $i \geq 0$, there exists $(q, q', \vec{y}) \in \delta$ such that $\pi[i] = (q, \vec{x})$ and $\pi[i+1] = (q', \vec{x} + \vec{y})$. A computation of a VASS(\mathbb{N}, k) \mathcal{A} is a computation π of \mathcal{A} considered as a VASS(\mathbb{Z}, k) such that the values of all counters are natural, i.e., for all i > 0 we have $\pi[i] \in Q \times \mathbb{N}^k$.

Paths and cycles. A path $\rho = (q_0, q'_0, \vec{y}_0), (q_1, q'_1, \vec{y}_1), \ldots$ in a VASS(\mathbb{Z}) (resp., VASS(\mathbb{N})) \mathcal{A} is a (finite or infinite) sequence of transitions (from δ) such that for all $0 \le i \le |\rho|$ we have $q'_i = q_{i+1}$. A finite path ρ is a cycle if $\rho = (q_0, q'_0, \vec{y}_0), \ldots, (q_m, q'_m, \vec{y}_m)$ and $q_0 = q'_m$. Every computation in a VASS(\mathbb{Z}) (resp., VASS(\mathbb{N})) defines a unique infinite path. Conversely, every infinite path in a VASS(\mathbb{Z}) \mathcal{A} starting with $q_0 \in Q_0$ defines a computation in \mathcal{A} . However, if \mathcal{A} is a VASS(\mathbb{N}), some paths do not have corresponding computations due to non-negativity restriction posed on the counters.

Regular computations. We say that a computation π of a VASS(\mathbb{Z}) (resp., VASS(\mathbb{N})) is regular if it corresponds to a path which is ultimately periodic, i.e., it is of the form $\alpha\beta^{\omega}$ where α, β are finite paths.

2.2 Decision problems

We present the decision problems that we study in the paper.

Cost functions. Consider a VASS(\mathbb{Z}, k) (resp., VASS(\mathbb{N}, k)) $\mathcal{A} = (Q, Q_0, \delta)$. A cost function f for \mathcal{A} is a function $f: Q \times \mathbb{Z}^k \to \mathbb{Z}$, which is linear with natural coefficients, i.e., for every q there exists $\vec{a} \in \mathbb{N}^k$ such that $f(q, \vec{z}) = \vec{a} \cdot \vec{z}$ for all $\vec{z} \in \mathbb{Z}^k$. We extend cost functions to computations as follows. For a computation π of \mathcal{A} , we define $f(\pi)$ as the sequence $f(\pi[0]), f(\pi[1]), \ldots$ Every cost function f is given by a labeling $l: Q \to \mathbb{N}^k$ by $f(q, \vec{z}) = l(q) \cdot \vec{z}$. We define the size of f, denoted by |f|, as the size of binary representation of l considered as a sequence of natural numbers of the length $|Q| \cdot k$.

Uniform cost functions. We say that a cost function f is uniform, if it is given by a constant function l, i.e., for all states q, q' and $\vec{z} \in \mathbb{Z}^k$ we have $f(q, \vec{z}) = f(q', \vec{z})$. Uniform cost functions are given by a single vector $\vec{a} \in \mathbb{N}^k$.

The long-run average. We are interested in the long-run average of the values returned by the cost function, which is formalized as follows. Consider an infinite sequence of real numbers w. We define

$$LimAvg(w) = \liminf_{k \to \infty} \frac{1}{k+1} \sum_{i=0}^{k} w[i].$$

Definition 1 (The average-value problems). Given a VASS(\mathbb{Z}, k) (resp., VASS(\mathbb{N}, k)) \mathcal{A} , a cost function f for \mathcal{A} , and a threshold $\lambda \in \mathbb{Q}$,

 $^{^{1}}$ Without loss of generality, we assume that the initial counter valuation is $\vec{0}$. We can encode any initial configuration in the VASS itself.

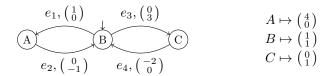


Figure 1: The VASS \mathcal{A}_e and its labeling

- the average-value problem (resp., the regular average-value problem) asks whether A has a computation (resp., a regular computation) π such that LIMAVG $(f(\pi)) \leq \lambda$, and
- the finite-value problem (resp., the regular finite-value problem) asks whether A has a computation (resp., a regular computation) π such that LimAvG($f(\pi)$) < ∞ .

The following example illustrates the regular average-value problem for $VASS(\mathbb{Z})$.

Example 2. Consider the VASS($\mathbb{Z}, 2$) $A_e = \langle \{A, B, C\}, \{B\}, \{e_1, e_2, e_3, e_4\} \rangle$ depicted in Figure 1 and a (non-uniform) cost function f is given by the labeling from Figure 1. $A \mapsto \begin{pmatrix} 4 \\ 0 \end{pmatrix}$,

Consider an infinite path $(e_1e_2e_3e_4)^{\omega}$ that defines the regular computation π_0 . This computation can be divided into blocks of 4 consecutive configurations. The (i+1)-th block has the following form

$$\dots (B, \begin{pmatrix} -i \\ 2i \end{pmatrix})(A, \begin{pmatrix} -i+1 \\ 2i \end{pmatrix})(B, \begin{pmatrix} -i+1 \\ 2i-1 \end{pmatrix})(C, \begin{pmatrix} -i+1 \\ 2i+2 \end{pmatrix})\dots$$

and the corresponding values of $f(\pi)$ are:

$$\dots i -4i+4 \qquad i \qquad 2i+2 \quad \dots$$

Therefore, the sum of values over each block is 6 and hence LIMAVG $(f(\pi_0)) = \frac{3}{2}$. The answer to the (regular) average-value problem with any threshold is YES. Consider $j \in \mathbb{N}$ and a path $(e_1e_2)^j(e_1e_2e_3e_4)^\omega$. Observe that it defines a regular computation π_i , which after the block $(e_1e_2)^j$ coincides with π_0 with all counter values shifted by $\binom{-2j}{3j}$. Note that in each block $e_1e_2e_3e_4$, the value f on the vector $\binom{-2j}{3j}$ are j, -8j, j, 3j. Therefore, $\operatorname{LIMAVG}(f(\pi_j)) = \operatorname{LIMAVG}(f(\pi_0)) + \frac{-3j}{4} = \frac{-3j-6}{4}$. It follows that for every threshold $\lambda \in \mathbb{Q}$, there exists a regular computation π_j with $\operatorname{LIMAVG}(f(\pi_0)) \leq \lambda$.

Organization. In this paper, we study the (regular) average-value problem for $VASS(\mathbb{Z},k)$ and $VASS(\mathbb{N}, k)$. First, we study the average-value problem for uniform cost functions (Section 3). Next, we consider the non-uniform case, where we focus on $VASS(\mathbb{Z})$ (Section 4). We start with solving the (regular) finite-value problem, and then we move to the regular average-value problem. We show that both problems are NP-complete. Next, we discuss the non-uniform case for VASS(N) and we show that the regular finite-value problem is decidable (and non-elementary), while the (regular) average-value problem is undecidable.

3 Uniform cost functions

In this section we study the average-value problem for $VASS(\mathbb{Z})$ and $VASS(\mathbb{N})$ for the uniform cost functions.

Integer-valued VASS: $VASS(\mathbb{Z})$ 3.1

Consider a VASS(\mathbb{Z}, k) \mathcal{A} and a uniform cost function f for \mathcal{A} . This function is defined by a single vector of coefficients $\vec{a} \in \mathbb{N}^k$. First, observe that we can reduce the number of the counters to one, which tracks the current cost. This counter c stores the value of $f(\pi[i]) = \vec{a} \cdot \vec{z}$, where \vec{z} are the values of counters. Initially, the counter c is $0 = \vec{a} \cdot \vec{0}$. Next, in each step i > 0, counters \vec{z} are updated with some values \vec{y} and we update c with $\vec{a} \cdot \vec{y}$. Note that the value of $f(\pi[i+1]) = \vec{a} \cdot (\vec{z} + \vec{y})$ equals $f(\pi[i]) + \vec{a} \cdot \vec{y}$.

Second, observe that the average-value problem for $VASS(\mathbb{Z}, 1)$ and uniform cost functions is equivalent to single-player average energy games (with no bounds on energy levels), which are solvable in polynomial time [3]. Moreover, average-energy games (with no bounds on energy levels) admit memoryless winning strategies and hence the average-value and the regular average-value problems coincide. In consequence we have:

Theorem 3. The average-value and the regular average-value problems for $VASS(\mathbb{Z})$ and uniform cost functions are decidable in polynomial time.

3.2 Natural-valued VASS: VASS(N)

For natural-valued VASS, we cannot reduce the number of counters to one; we need to track all counters to make sure that all of them have non-negative values. Moreover, we show that the average-value problem for (single counter) $VASS(\mathbb{N}, 1)$ is in PSPACE while the average-value problem for $VASS(\mathbb{N})$ is nonelementary.

First, the average-value problem for (single counter) $VASS(\mathbb{N}, 1)$ in the uniform case is equivalent to single-player average energy games with non-negativity constraint on the energy values. The latter problem is in PSPACE and it is NP-hard [3].

In the multi-counter case, we show that the average-value problem is mutually reducible to the configuration reachability problem for VASS, which has recently been shown nonelementary hard [12]. We additionally assume that the cost function depends on all its arguments, i.e., all coefficients are non-negative. This assumption allows us show that if there is a computation of the average value below some λ , then there is one, which is lasso-shaped and the cycle in the lasso has exponential size in the size of the VASS and λ . Therefore, we can non-deterministically pick such a cycle and check whether it is reachable from the initial configuration.

Theorem 4. The average-value and the regular average-value problems for $VASS(\mathbb{N})$ and uniform cost functions with non-zero coefficients are decidable and mutually reducible to the configuration reachability problem for $VASS(\mathbb{N})$.

Proof. We present both directions of the reduction.

Hardness. Consider a $A_0 \in \mathbf{VASS}(\mathbb{N}, k)$ and two configurations $(q, \vec{v}), (q', \vec{v}')$. Without loss of generality, we assume that $\vec{v} = \vec{v}' = \vec{0}$. We construct a $VASS(\mathbb{N}, k+1)$ A_1 build as an extension of A_0 with two additional states q_S, q_F and an additional counter c_{k+1} . The state q_S is the only initial state with a single outgoing transition, which goes to q, where only the counter c_{k+1} is incremented, and other counters are unchanged. Then, the counter c_{k+1} is not changed except for the state q_F . The state q_F is a sink with an incoming transition $(q', q_F, \vec{0})$ and two loops over q_F . One loop does not change the counters and the other that decrements c_{k+1} (and does not change the other counters). The cost function returns the sum of counters. Observe that every computation of the average-value 0 has to reach q_F with first k counters being 0. Such a computation exists if and only if $(q', \vec{0})$ is reachable in A_0 from $(q, \vec{0})$.

Decidability. We can solve the average-value problem for $VASS(\mathbb{N})$ and uniform cost functions with non-zero coefficients in non-deterministic exponential time with a single query to the VASS-reachability oracle.

Consider a VASS(N, k) \mathcal{A} and assume it has a computation π of the average-value not exceeding λ . Then, for every $\epsilon > 0$, the computation π contains a cyclic computation (a cycle w.r.t. the state and the valuation of counters) of the average value $M = \lambda + \epsilon$. Let π^C be a cyclic computation of the minimal length among computations of the average value M. Consider three sets of positions in π^C :

- (a) N_0 , where the value of f is less than M,
- (b) N_1 , where the value of f belongs to [M, 2M], and
- (c) N_2 , where the value of f exceeds 2M.

Note that $|N_2| \leq |N_0|$, as otherwise the average value exceeds M. Since π^C has the minimal-length, it does not contain a subcycle computation. A subcycle computation has either the average less-or-equal M, and hence it is shorter than π^C , or it has the average greater than M, and hence it can be removed. Therefore, all positions in π^C are distinct. Finally, since all coefficients are non-zero, the value of each counter is bounded by M, and hence $|N_0|, |N_1| \leq |Q| \cdot (2M)^k$. In consequence, the length of π^C is bounded by $3|Q| \cdot (2M)^k$.

This has two consequences. First, there are finitely many such cyclic computations and hence there exists a cyclic computation of the value at most λ and length $3|Q| \cdot (2\lambda)^k$. Second, to decide the average-value problem for an VASS(N, k), we can non-deterministically generate a cyclic computation of length bounded by $3|Q| \cdot (2\lambda)^k$, verify that (a) its average does not exceed λ , and (b) it is reachable from some initial configuration.

Observe that the above procedure implies that the average-value and the regular average-value problems coincide for $VASS(\mathbb{N})$ under uniform cost functions with non-zero coefficients.

We have used in the proof the fact that f depends on all counters. We conjecture that this assumption can be lifted:

Open question 5. Is the average-value problem for $VASS(\mathbb{N})$ and uniform cost functions decidable?

4 General cost functions and $VASS(\mathbb{Z})$

First, we consider (an extension of) the regular finite-value problem for $VASS(\mathbb{Z})$ (Section 4.1). We show that one can decide in non-deterministic polynomial time whether a given VASS has a regular computation (a) of the value $-\infty$, or (b) of some finite value. To achieve this, we introduce path summarizations, which allow us to state conditions that entail (a) and respectively (b). We show that these conditions can be checked in NP.

We apply the results from Section 4.1 to solve the regular average-value problem (Section 4.2). We consider only VASS that have some computation of a finite value, and no computation of the value $-\infty$, i.e., the answer to (a) is NO and the answer to (b) is YES. In other cases we can easily answer to the regular average-value problem. If the answer to (a) is YES, then for any threshold we answer YES. If the answers to (a) and (b) are NO, then for any threshold we answer NO. Finally, we show NP-hardness of the regular finite-value and the regular average-value problems (Section 4.3).

The main result of this section is the following theorem:

Theorem 6. The regular finite-value and the regular average-value problems for $VASS(\mathbb{Z})$ with (general) cost functions are NP-complete.

We fix a VASS(\mathbb{Z}, k) $\mathcal{A} = \langle Q, Q_0, \delta \rangle$, with the set of states Q and the set of transitions δ , and a cost function f, which we refer to throughout this section. We exclude the complexity statements, where the asymptotic behavior applies to all \mathcal{A} and f.

4.1 The finite-value problem

For a path ρ , we define characteristics $GAIN(\rho)$, $VALS(\rho)$, which summarize the impact of ρ on the values of counters (GAIN) and the value of the (partial) average of costs (VALS).

Let ρ be of the form $(q_1, q_2, \vec{y}_1) \dots (q_m, q_{m+1}, \vec{y}_m)$. We define $GAIN(\rho)$ as the sum of updates along ρ , i.e., $GAIN(\rho) = \sum_{i=1}^{m} \vec{y}_i$. The vector $GAIN(\rho)$ is the update of counters upon the whole path ρ . Observe that

$$GAIN(\rho_1 \rho_2) = GAIN(\rho_1) + GAIN(\rho_2).$$

Let $l: Q \to \mathbb{N}^k$ be the function representing f, i.e., for every $\vec{z} \in \mathbb{Z}^k$ we have $f(q, \vec{z}) = l(q) \cdot \vec{z}$. We define $VALS(\rho)$ as the sum of vectors $l(q_i)$ along ρ , i.e., $VALS(\rho) = \sum_{i=1}^m l(q_i)$. Note that we exclude the last state q_{m+1} , and hence we have

$$VALS(\rho_1 \rho_2) = VALS(\rho_1) + VALS(\rho_2).$$

The vector $Vals(\rho)$ describes the coefficients with which each counter contributes to the average along the path ρ .

Consider a regular computation π and a path $\rho_1(\rho_2)^{\omega}$ that corresponds to it. Let π' be the (regular) computation obtained from π by contracting $|\rho_2|$ to a single transition $(q_f, q_f, \text{GAIN}(\rho_2))$, which is a loop over a fresh state q_f . The counters over this transition are updated by $\text{GAIN}(\rho_2)$ and the cost function in q_f is defined as $f(q_f, \vec{z}) = \frac{1}{|\rho_2|} \text{VALS}(\rho_2) \cdot \vec{z}$.

The values of π and π' may be different, but they differ only by some finite value. Indeed, the difference over a single iteration of ρ_2 updates and computation of the partial averages are interleaved along ρ_2 , while in $(q_f, q_f, \text{GAIN}(\rho_2))$ we first compute the partial average and then update the counters. Therefore, that difference is a finite value N that does not depend on the initial values of counters. It follows that the difference between the average values of (the computations corresponding to) ρ_2 and $(q_f, q_f, \text{GAIN}(\rho_2))$ is bounded by N.

Finally, observe that the value of π' can be easily estimated. Observe that the partial average of the first n+1 values of f in π' equals

$$\frac{1}{n+1} \left(\vec{0} \cdot \text{VALS}(\rho_2) + \text{GAIN}(\rho_2) \cdot \text{VALS}(\rho_2) + \ldots + n \text{GAIN}(\rho_2) \cdot \text{VALS}(\rho_2) \right) = \frac{n}{2} \text{GAIN}(\rho_2) \cdot \text{VALS}(\rho_2).$$

In consequence, we have the following:

Lemma 7. Let π be a regular computation corresponding to a path $\rho_1(\rho_2)^{\omega}$. Then, one of the following holds:

- 1. $GAIN(\rho_2) \cdot VALS(\rho_2) < 0$ and $LIMAVG(f(\pi)) = -\infty$, or
- 2. $GAIN(\rho_2) \cdot VALS(\rho_2) = 0$ and $LIMAVG(f(\pi))$ is finite, or
- 3. $GAIN(\rho_2) \cdot VALS(\rho_2) > 0$ and $LIMAVG(f(\pi)) = \infty$.

Example 8. Consider the VASS A_e and the cost function f from Example 2. We have shown that the computation defined by the path $(e_1e_2e_3e_4)^{\omega}$ has finite average value. This can be algorithmically computed using Lemma 7. We compute

GAIN
$$(e_1e_2e_3e_4) = {1 \choose 0} + {0 \choose -1} + {0 \choose 3} + {-2 \choose 0} = {-1 \choose 2}$$

VALS $(e_1e_2e_3e_4) = {1 \choose 1} + {0 \choose 4} + {1 \choose 1} + {0 \choose 3} = {6 \choose 3}$

Therefore, $GAIN(e_1e_2e_3e_4) \cdot VALS(e_1e_2e_3e_4) = 0$.

Reduction to integer quadratic programming. Lemma 7 reduces the regular finite-value problem to finding a cycle with $GAIN(\rho) \cdot VALS(\rho) < 0$ (resp., $GAIN(\rho) \cdot VALS(\rho) = 0$). We show that existence of such a cycle can be stated as a polynomial-size instance of integer quadratic programming, which can be decided in NP [13]. An instance of integer quadratic programming consists of a symmetric matrix $\mathbf{A} \in \mathbb{Q}^{n \times n}, \vec{a} \in \mathbb{Q}^{n}, d \in \mathbb{Q}, \mathbf{B} \in \mathbb{Q}^{n \times m}, \vec{c} \in \mathbb{Q}^{m}$ and it asks whether there exists a vector $\vec{x} \in \mathbb{Z}^{n}$ satisfying the following system:

$$\vec{x}^T \mathbf{A} \vec{x} + \vec{a} \vec{x} + d \le 0$$
$$\mathbf{B} \vec{x} \le \vec{c}$$

Let ρ be a cycle. Observe that both $GAIN(\rho)$, $VALS(\rho)$ depend only on the multiplicity of transitions occurring in ρ ; the order of transitions is irrelevant. Consider a vector $\vec{x} = (x[1], \dots, x[m])$ of multiplicities of transitions e_1, \dots, e_m in ρ . Then,

$$Gain(\rho) \cdot Vals(\rho) = \sum_{1 \le i, j \le m} x[i]x[j]Gain(e_i) \cdot Vals(e_j)$$

Therefore, we have

$$2 \cdot GAIN(\rho) \cdot VALS(\rho) = \vec{x}^T \mathbf{A} \vec{x}$$
 (1)

for a symmetric matrix $\mathbf{A} \in \mathbb{Z}^{m \times m}$ defined for all i, j as

$$\mathbf{A}[i,j] = \text{GAIN}(e_i) \cdot \text{VALS}(e_i) + \text{GAIN}(e_i) \cdot \text{VALS}(e_i).$$

The left hand side of (1) is multiplied by 2 to avoid division by 2. It follows that if $GAIN(\rho) \cdot VALS(\rho) < 0$ (resp., $GAIN(\rho) \cdot VALS(\rho) = 0$), then the inequality $\vec{x}^T \mathbf{A} \vec{x} - 1 \le 0$ (resp., $\vec{x}^T \mathbf{A} \vec{x} \le 0$) has a solution.

Note that the above inequality can have a solution, which does not correspond to a cycle. We can encode with a system of linear inequalities that \vec{x} corresponds to a cycle. Consider $S \subseteq Q$, which corresponds to all states visited by ρ . It suffices to ensure that (a) for all $s \in S$, the number of incoming transitions to s, which is the sum of multiplicities of transitions leading to s, equals the number of transitions outgoing from s, (b) for every $s \in S$, the number of outgoing transitions is greater or equal to 1, and (c) for $s \in Q \setminus S$, the number of incoming and outgoing transitions is 0. Finally, we state that all multiplicities are non-negative. We can encode such equations and inequalities as $\mathbf{B}_S \vec{x} \leq \vec{c}_S$. Observe that every vector of multiplicities $\vec{x} \in \mathbb{Z}$ satisfies $\vec{x}^T \mathbf{A} \vec{x} - 1 \leq 0$ (resp., $\vec{x}^T \mathbf{A} \vec{x} \leq 0$) and $\mathbf{B}_S \vec{x} \leq \vec{c}_S$ defines a cycle ρ with $\mathrm{GAIN}(\rho)$, $\mathrm{VALS}(\rho) < 0$ (resp., $\mathrm{GAIN}(\rho)$, $\mathrm{VALS}(\rho) = 0$). The matrices \mathbf{A} , \mathbf{B}_S and the vector \vec{c}_S are polynomial in $|\mathcal{A}|$. The set S can be picked non-deterministically. In consequence, we have the following:

Lemma 9. The problem: given a VASS(\mathbb{Z}, k) \mathcal{A} and a cost function f, decide whether \mathcal{A} has a regular computation of the value $-\infty$ (resp., less than $+\infty$) is in NP.

Example 10. Consider the VASS A_e and the cost function f from Example 2. We consider some cycle ρ starting in B over all states of A_e . Let $\vec{x} \in \mathbb{N}^4$ be the multiplicities of transitions e_1, e_2, e_3, e_4 respectively in ρ Then, we have

$$\mathbf{A} = \begin{bmatrix} 2 & 3 & 4 & -2 \\ 3 & 0 & -1 & -9 \\ 4 & -1 & 6 & 1 \\ -2 & -9 & 1 & 0 \end{bmatrix}$$

Since ρ is a cycle starting from B, we have $x_1 = x_2$ and $x_3 = x_4$. Therefore, we can eliminate x_2, x_4 and the matrix A can be simplified to $A' \in \mathbb{Z}^{2 \times 2}$ defined as

$$\mathbf{A}' = \begin{bmatrix} 8 & -8 \\ -8 & 8 \end{bmatrix}$$

where x_1 (resp., x_3) is the multiplicity of the cycle e_1e_2 (resp., e_3e_4). Let $\vec{x}' = (x_1, x_3)$. Note that

$$(\vec{x}')^T \mathbf{A}' \vec{x} = (x_1 - x_3)^2$$

It follows that a cycle ρ defines the computation of some finite value if and only if it contains the multiplicities of transitions e_1, e_2, e_3, e_4 are equal.

4.2 The regular average-value problem

In this section, we study the regular average-value problem for $\mathbf{VASS}(\mathbb{Z})$. We assume that \mathcal{A} has a regular computation of finite value and does not have a regular computation of the value $-\infty$. Finally, we consider the case of the threshold $\lambda = 0$. The case of an arbitrary threshold $\lambda \in \mathbb{Q}$ can be easily reduced to the case $\lambda = 0$ (see [9, 8] for intuitions and techniques in mean-payoff games a.k.a. limit-average games).

Consider a regular computation π and let $\rho_1(\rho_2)^{\omega}$ be a path corresponding to π . We derive the necessary and sufficient conditions on ρ_1, ρ_2 to have LIMAVG $(f(\pi)) \leq 0$.

Assume that LIMAVG $(f(\pi)) \le 0$. Due to Lemma 7, we have $GAIN(\rho_2) \cdot VALS(\rho_2) = 0$, which implies that every iteration of ρ_2 has the same average. Therefore, LIMAVG $(f(\pi))$ is the average value of ρ_2 (starting

with counter values defined by ρ_1). The latter value is at most 0 if and only if the sum of values along ρ_2 is at most 0. We define this sum below.

The sum of values $\operatorname{Sum}_{\vec{g}}(\rho)$. Let l be the labeling defining the cost function f. Consider a path ρ of length m such that $\rho = (q_1, q_2, \vec{y}_1) \dots (q_m, q_{m+1}, \vec{y}_m)$. Let π be the computation corresponding to ρ with the initial counter values being \vec{g} . Then, the value of counters at the position i in π is $\vec{g} + \sum_{j=1}^{i-1} \vec{y}_j$. Therefore, the sum of values over ρ starting with counter values $\vec{g} \in \mathbb{Z}^k$, denoted by $\operatorname{Sum}_{\vec{q}}(\rho)$, is given by the following formula:

$$\mathsf{Sum}_{\vec{g}}(\rho) = \sum_{i=1}^m \left(\vec{g} + \sum_{j=1}^{i-1} \vec{y}_j \right) \cdot l(q_i)$$

We have the following:

Lemma 11. There exists a regular computation π with LimAvG $(f(\pi)) \leq 0$ if and only if there exist paths ρ_1, ρ_2 such that

- (C1) $\rho_1(\rho_2)^{\omega}$ is an infinite path and ρ_2 is a cycle,
- (C2) GAIN(ρ_2) · VALS(ρ_2) = 0, and
- (C3) $\operatorname{\mathsf{Sum}}_{\operatorname{GAIN}(\rho_1)}(\rho_2) \leq 0.$

The proof of Lemma 11 has been relegated to the appendix.

Due to results from the previous section, we can check in NP the existence of ρ_1 , ρ_2 satisfying (C1) and (C2). We call a cycle ρ_2 balanced if $GAIN(\rho_2) \cdot VALS(\rho_2) = 0$. In the remaining part we focus on condition (C3), while keeping in mind (C1) and (C2).

The plan of the proof. The proof has three key ingredients which are as follows: factorizations, quadratic factor elimination, and the linear case.

• Factorizations. We show that we can consider only paths ρ_2 of the form

$$\alpha_0 \beta_1^{n[1]} \alpha_1 \beta_2^{n[2]} \dots \beta_p^{n[p]} \alpha_p,$$

where $|\alpha_i|, |\beta_i| \leq |\mathcal{A}|$ and each β_i is a distinct simple cycle (Lemma 13). It follows that p is exponentially bounded in $|\mathcal{A}|$. Next, we show that for ρ_2 in such a form we have

$$\mathsf{Sum}_{\mathrm{GAIN}(\rho_1)}(\rho_2) = \vec{n}^T \mathbf{B} \vec{n} + \vec{c} \cdot \vec{n} + e$$

where $\mathbf{B} \in \mathbb{Z}^{p \times p}$, $\vec{c} \in \mathbb{Z}^p$ (Lemma 14). Therefore, $\mathsf{Sum}_{\mathsf{GAIN}(\rho_1)}(\rho_2) \leq 0$ can be presented as an instance of integer quadratic programming, where the variables correspond to multiplicities of simple cycles.

However, there are two problems to overcome. First, ρ_2 has to be balanced, i.e., it has to satisfy GAIN(ρ_2) · VALS(ρ_2) = 0, which introduces another quadratic equation. Solving a system of two quadratic equations over integers is considerably more difficult (see [13] for references). Second, p is (bounded by) the number of distinct simple cycles and hence it can be exponential in $|\mathcal{A}|$. Therefore, **B** and \vec{c} may have exponential size (of the binary representation) in $|\mathcal{A}|$.

- Quadratic factor elimination. To solve these problems, we fix a sequence TPL = $(\alpha_0, \beta_1, \alpha_1, \dots, \beta_p, \alpha_p)$ and consider \mathbf{B}_{TPL} , \vec{c}_{TPL} , e_{TPL} for TPL. We show that one of the following holds (Lemma 15 and Lemma 17):
 - $\ \vec{n}^T \mathbf{B}_{\text{TPL}} \vec{n} + \vec{c}_{\text{TPL}} \cdot \vec{n} + e_{\text{TPL}} \leq 0$ has a simple solution, or
 - there exist $\vec{d}_{\text{TPL}} \in \mathbb{Z}^p, h_{\text{TPL}} \in \mathbb{Z}$ such that for all vectors \vec{n} , if the cycle $\rho_2 = \alpha_0 \beta_1^{n[1]} \dots \beta_p^{n[p]} \alpha_p$ is balanced (C2), then

$$\vec{n}^T \mathbf{B}_{\mathrm{TPL}} \vec{n} = \vec{d}_{\mathrm{TPL}} \cdot \vec{n} + h_{\mathrm{TPL}}.$$

We can decide in non-deterministic polynomial time whether the first condition holds (Lemma 22).

• The linear case. Assuming that the second condition holds, we reduce the problem of solving the quadratic inequality $\vec{n}^T \mathbf{B}_{\text{TPL}} \vec{n} + \vec{c}_{\text{TPL}} \cdot \vec{n} + e_{\text{TPL}} \leq 0$ to solving the linear inequality

$$(\vec{c}_{\text{TPL}} + \vec{d}_{\text{TPL}}) \cdot \vec{n} + (e_{\text{TPL}} + h_{\text{TPL}}) \le 0.$$
 (lin)

Moreover, we can compute \vec{d}_{TPL} , h_{TPL} from the sequence TPL. At this point, we can solve the problem in non-deterministic exponential time. Next, we argue that we do not have to compute the whole system.

We show that if $(\vec{c}_{\text{TPL}} + \vec{d}_{\text{TPL}}) \cdot \vec{n} + e_{\text{TPL}} + h_{\text{TPL}} \leq 0$ has a solution, then it has a solution for a vector \vec{n}_0 with $m = O(|\mathcal{A}|)$ non-zero components. Therefore, we can remove cycles corresponding to 0 coefficients of \vec{n}_0 . Still, $\sum_{i=0}^p |\alpha_i|$ can be exponential in $|\mathcal{A}|$, but this operation shortens the size of the template, i.e., the value $\sum_{i=0}^p |\alpha_i| + \sum_{i=1}^p |\beta_i|$, and hence by iterating it we get a polynomial size template, which yields a polynomial-size system of inequalities. These inequalities can be solved in NP.

We now present the details of each ingredient.

4.2.1 Factorizations

The regular finite-value problem has been solved via reduction to solving (quadratic and linear) inequalities. In this section, we show a reduction of the regular average-value problem to linear and quadratic inequalities as well. First, we establish that we can consider only cycles ρ , which have a compact representation using templates parametrized by multiplicities of cycles. The value $\operatorname{Sum}_{\vec{g}}(\rho)$ for a cycle represented by a template is given by a quadratic function in the multiplicities of cycles.

Templates and multiplicities. A template TPL is a sequence of paths $(\alpha_0, \beta_1, \alpha_1, \dots, \beta_p, \alpha_p)$ such that all β_1, \dots, β_p are cycles and $\alpha_0 \beta_1 \alpha_1 \dots \beta_p \alpha_p$ is a cycle. A template is minimal if for all $i \in \{0, \dots, p\}$ we have $|\alpha_i| < |Q|$ and all β_i are pairwise distinct simple cycles. For every vector $\vec{n} \in \mathbb{N}^p$, called multiplicities, we define $\text{TPL}(\vec{n})$ as a cycle

$$\alpha_0 \beta_1^{n[1]} \alpha_1 \beta_2^{n[2]} \dots \beta_p^{n[p]} \alpha_p. \tag{*}$$

A cycle ρ has a (minimal) factorization if there exists a (minimal) template and multiplicities \vec{n} such that $\rho = \text{TpL}(\vec{n})$.

Observe that every cycle ρ has a factorization such that for all i we have $|\alpha_i| < |Q|$ and each β_i is a simple cycle. However, the sequence β_i 's can have repetitions. The following lemma states that if a cycle β occurs twice in ρ , then we can group them together.

Lemma 12 (Cycle grouping). Consider $\vec{g} \in \mathbb{Z}^k$ and a cycle $\alpha_0 \beta \alpha_1 \beta \alpha_2$. Then, one of the following holds:

$$\operatorname{Sum}_{\vec{g}}(\alpha_0 \beta^2 \alpha_1 \alpha_2) \leq \operatorname{Sum}_{\vec{g}}(\alpha_0 \beta \alpha_1 \beta \alpha_2)$$

$$\operatorname{Sum}_{\vec{g}}(\alpha_0 \alpha_1 \beta^2 \alpha_2) \leq \operatorname{Sum}_{\vec{g}}(\alpha_0 \beta \alpha_1 \beta \alpha_2)$$

Proof sketch. To show the lemma observe that

$$\begin{split} \mathsf{Sum}_{\vec{g}}(\alpha_0\beta\alpha_1\beta\alpha_2) &= \frac{1}{2} \big(\mathsf{Sum}_{\vec{g}}(\alpha_0\beta^2\alpha_1\alpha_2) + \\ &\quad \quad \\ \mathsf{Sum}_{\vec{g}}(\alpha_0\alpha_1\beta^2\alpha_2) \big) \end{split}$$

Careful repeated application of Lemma 12 implies that we can look for a cycle ρ satisfying (C1), (C2) and (C3) among cycles that have a minimal factorization:

Lemma 13. For every cycle ρ and $\vec{g} \in \mathbb{Z}^k$, there exists a cycle ρ' that has a minimal factorization (*) such that $GAIN(\rho) = GAIN(\rho')$, $VALS(\rho) = VALS(\rho')$ and $Sum_{\vec{q}}(\rho) \ge Sum_{\vec{q}}(\rho')$.

Proof sketch. Note that repeated application of Lemma 12 allows us to group simple cycles together but it may create connecting paths of length greater than |Q|. In such a case, we extract simple cycles from long connecting paths and group them together again using Lemma 12. Such process terminates as in each iteration it either decreases the sum of lengths' of connecting paths or does not change this sum but decreases the number of (non-grouped) cycles. Note that the initial cycle ρ and the resulting cycle ρ' have the same multisets of transitions and hence $GAIN(\rho) = GAIN(\rho')$ and $VALS(\rho) = VALS(\rho')$. It follows that this process preserves condition (C2).

Consider a template $TPL = (\alpha_0, \beta_1, \alpha_1, \dots, \beta_p, \alpha_p)$. We present $Sum_{\vec{g}}(TPL(\vec{n}))$ as a function from multiplicities of simple cycles $\vec{n} \in \mathbb{N}$ into \mathbb{Z} . First, observe that

$$\mathsf{Sum}_{\vec{q}}(\mathrm{TPL}(\vec{n})) = \mathsf{Sum}_{\vec{0}}(\mathrm{TPL}(\vec{n})) + \vec{g} \cdot \mathrm{VALS}(\mathrm{TPL}(\vec{n})).$$

The expression $VALS(TPL(\vec{n}))$ is a linear expression in \vec{n} with natural coefficients, and the expression $Sum_{\vec{0}}(TPL(\vec{n}))$ is a quadratic function in each of its arguments \vec{n} .

Lemma 14. Given a template TPL we can compute in polynomial time in $|\text{TPL}| + |\mathcal{A}| + |f|$, a symmetric matrix $\mathbf{B}_{\text{TPL}} \in \mathbb{Z}^{p \times p}$, $\vec{c}_{\text{TPL}} \in \mathbb{Z}^p$ and $e_{\text{TPL}} \in \mathbb{Z}$ such that the following holds:

$$2 \cdot \mathsf{Sum}_{\vec{0}}(\mathsf{TPL}(\vec{n})) = \vec{n}^T \mathbf{B}_{\mathsf{TPL}} \vec{n} + \vec{c}_{\mathsf{TPL}} \vec{n} + e_{\mathsf{TPL}}$$
 (2)

Moreover, for all $i, j \in \{1, \dots, p\}$ we have

$$\mathbf{B}_{\text{TPL}}[i,j] = \text{GAIN}(\beta_{\min(i,j)}) \cdot \text{VALS}(\beta_{\max(i,j)}). \tag{3}$$

The proof of Lemma 14 has been relegated to the appendix.

Observe that \mathbf{B}_{TPL} is similar to the matrix \mathbf{A} from (1). We exploit this similarity in the following section to eliminate the term $\vec{n}^T \mathbf{B}_{\text{TPL}} \vec{n}$ if possible.

4.2.2 Elimination of the quadratic factor

We show how to simplify the expression (2) of Lemma 14 for $Sum_{\vec{0}}(TPL(\vec{n}))$. We show that either the inequality

$$Sum_{\vec{o}}(TPL(\vec{n})) + \vec{q} \cdot VALS(TPL(\vec{n})) < 0$$

has a simple solution for every \vec{g} , or the quadratic term in (2) of Lemma 14 can be substituted with a linear term.

Negative and linear templates. Consider a template TPL. A template TPL is *positive* (resp., *negative*) if there exist multiplicities $\vec{n}_1, \vec{n}_2 \in \mathbb{N}^p$ such that

- 1. $\vec{n}_1^T \mathbf{B}_{\text{TPL}} \vec{n}_1 > 0 \text{ (resp., } \vec{n}_1^T \mathbf{B}_{\text{TPL}} \vec{n}_1 < 0), \text{ and}$
- 2. for every $t \in \mathbb{N}^+$, we have

$$Gain(Tpl(t\vec{n}_1 + \vec{n}_2)) \cdot Vals(Tpl(t\vec{n}_1 + \vec{n}_2)) = 0.$$

A template TPL is linear if there exist $\vec{d}_{\text{TPL}} \in \mathbb{Z}^p$ and $h_{\text{TPL}} \in \mathbb{Z}$ such that for all \vec{n} , if GAIN(TPL(\vec{n})) · VALS(TPL(\vec{n})) = 0, then $\vec{n}^T \mathbf{B}_{\text{TPL}} \vec{n} = \vec{d}_{\text{TPL}} \cdot \vec{n} + h_{\text{TPL}}$.

We observe that the existence of a negative cycle TPL implies that $\mathsf{Sum}_{\vec{g}}(\mathsf{TPL}(\vec{n})) \leq 0$ has a solution for every $\vec{g} \in \mathbb{N}^k$, which in turn implies that the answer to the average-value problem is YES. Basically, for ρ^t defined as $\mathsf{TPL}(t\vec{n}_1 + \vec{n}_2)$ and t big enough we can make $\mathsf{Sum}_{\vec{g}}(\rho^t)$ arbitrarily small.

Lemma 15. If there exist a negative template, whose any state is reachable from some initial state, then the answer to the regular average-value problem with threshold 0 is YES.

Proof sketch. Assume that TPL is negative and consider ρ^t defined as TPL $(t\vec{n}_1+\vec{n}_2)$. Note that $2 \cdot \mathsf{Sum}_{\vec{0}}(\rho^t) = (t\vec{n}_1+\vec{n}_2)^T \mathbf{B}_{\mathsf{TPL}}(t\vec{n}_1+\vec{n}_2) + \vec{c}_{\mathsf{TPL}}(t\vec{n}_1+\vec{n}_2) + e_{\mathsf{TPL}}$ is a quadratic polynomial in t with the leading coefficient $\vec{n}_1^T \mathbf{B}_{\rho_1} \vec{n}_1$ negative, whereas $\vec{g} \cdot \mathsf{VALS}(\rho^t)$ is linear in t. Therefore, for t big enough (which depends on \vec{g}) $\mathsf{Sum}_{\vec{g}}(\rho^t) = \mathsf{Sum}_{\vec{0}}(\rho^t) + \vec{g} \cdot \mathsf{VALS}(\rho^t)$ is negative and ρ^t is balanced. It follows that ρ_1 leading from an initial state to the first state of TPL and $\rho_2 = \rho^t$ for t big enough satisfy conditions (C1), (C2), (C3) of Lemma 11 and hence the answer to the average-value problem is YES.

We show that either there is a template, which is negative or all templates are linear (Lemma 17). Next, we show that we can check in non-deterministic polynomial time whether there exists a negative template (Lemma 22). We start with an example presenting one of the main ideas.

Example 16. Consider the VASS A_e and the cost function f from Example 2. We consider balanced cycles ρ starting in B. We know from Example 10 that in balanced cycles all transitions have the same number of occurrences. Therefore, ρ is factorized into one of two minimal templates: $\text{TPL}_1 = (\epsilon, e_1e_2, \epsilon, e_3e_4, \epsilon)$ or $\text{TPL}_2 = (\epsilon, e_3e_4, \epsilon, e_1e_2, \epsilon)$.

$$\mathbf{B}_{\mathrm{TPL}_1} = egin{bmatrix} 4 & -1 \ -1 & 4 \end{bmatrix} \quad \mathbf{B}_{\mathrm{TPL}_2} = egin{bmatrix} 4 & -7 \ -7 & 4 \end{bmatrix}$$

Since transitions have the same number of occurrences, the multiplicities of e_1e_2 and e_3e_4 are equal and hence we consider \vec{n} such that n[1] = n[2]. For such vectors \vec{n} we have $\vec{n}\mathbf{B}_{\mathrm{TPL}_1}\vec{n} = 6(n[1])^2$ and $\vec{n}\mathbf{B}_{\mathrm{TPL}_2}\vec{n} = -6(n[1])^2$. Therefore, TPL_1 is positive and TPL_2 is negative.

Interestingly, $\vec{n}^T \mathbf{B}_{\mathrm{TPL}_1} \vec{n} = -\vec{n}^T \mathbf{B}_{\mathrm{TPL}_2} \vec{n}$. This is not a coincidence. Recall the matrix \mathbf{A}' from Example 10 and observe that $\mathbf{A}' = \mathbf{B}_{\mathrm{TPL}_1} + \mathbf{B}_{\mathrm{TPL}_2}$. As we consider \vec{v} such that $\vec{n}^T \mathbf{A}' \vec{n} = 0$ and hence the above equality follows.

Lemma 17. (1) There exists a negative template or all templates are linear. (2) If there exists a positive template TPL, then there exists a negative one of the size bounded by $|\text{TPL}|^2$. (3) If a template TPL is linear, we can compute \vec{d}_{TPL} , h_{TPL} in polynomial time in $|\text{TPL}| + |\mathcal{A}| + |f|$.

Proof ideas. Consider a template TPL with all connecting paths being empty, i.e., TPL = $(\epsilon, \beta_1, \dots, \beta_p, \epsilon)$. Let \vec{n} be a vector of multiplicities such that a cycle TPL (\vec{n}) is balanced, i.e., GAIN $(\text{TPL}(\vec{n})) \cdot \text{VALS}(\text{TPL}(\vec{n})) = 0$. Then, TPL $(t\vec{n})$ is balanced for all $t \in \mathbb{N}$. We consider three cases:

The case $\vec{n}_1^T \mathbf{B}_{\mathrm{TPL}} \vec{n} < 0$. Then, TPL is negative. Vectors $\vec{n}_1 = \vec{n}$ and $\vec{n}_2 = \vec{0}$ witness negativity.

The case $\vec{n}^T \mathbf{B}_{\text{TPL}} \vec{n} > 0$. Then, TPL is positive and we show that the template with reversed cycles $\text{TPL}^R = (\epsilon, \beta_p, \dots, \beta_1, \epsilon)$ is negative.

First, observe that $GAIN(TPL(\vec{n})) \cdot VALS(TPL(\vec{n})) = 0$ can be stated as a matrix equation $\vec{n}^T \mathbf{A} \vec{n} = 0$, where for all i, j we have

$$\mathbf{A}[i, j] = \text{GAIN}(\beta_i) \cdot \text{VALS}(\beta_i) + \text{GAIN}(\beta_i) \cdot \text{VALS}(\beta_i).$$

This can be derived in virtually the same way as in Section 4.1. Next, recall from Lemma 14 that \mathbf{B}_{TPL} is similar to \mathbf{A} . For all i, j we have

$$\mathbf{B}_{\text{TPL}}[i,j] = \text{GAIN}(\beta_{min(i,j)}) \cdot \text{VALS}(\beta_{max(i,j)})$$

A similar identity holds for TPL^R , but i, j are substituted by respectively p - i + 1 and p - j + 1 and min is swapped with max. In consequence, we have

$$\mathbf{B}_{\text{Tpl},R}[p+1-i,p+1-j] = \text{GAIN}(\beta_{max(i,j)}) \cdot \text{VALS}(\beta_{min(i,j)})$$

Therefore, for all i, j we have

$$A[i,j] = B_{TPL}[i,j] + B_{TPLR}[p-i+1,p-j+1]$$

It follows that

$$0 = \vec{n}^T \mathbf{A} \vec{n} = \vec{n}^T \mathbf{B}_{\text{Tpl}} \vec{n} + \vec{n_R}^T \mathbf{B}_{\text{Tpl}} \vec{n_R}$$

where \vec{n}_R is the reversed vector \vec{n} . Note that $\text{TPL}^R(\vec{n}_R)$ has the same multiset of transitions as $\text{TPL}^R(\vec{n}_R)$, and hence $\text{TPL}^R(\vec{n}_R)$ is balanced, and $\text{TPL}^R(t\vec{n}_R)$ is balanced for all $t \in \mathbb{N}$. It follows that TPL^R is negative.

The above two cases fail. Then for all \vec{n} , if GAIN(TPL(\vec{n}))·VALS(TPL(\vec{n})) = 0, then $\vec{n}^T \mathbf{B}_{\text{TPL}} \vec{n} = 0$. Therefore, TPL is linear with \vec{d}_{TPL} , h_{TPL} being $\vec{0}$ and 0 respectively. In the general case, if the connecting paths are non-empty, then \vec{d}_{TPL} , h_{TPL} need not be 0. Intuitively, GAIN(TPL(\vec{n}))·VALS(TPL(\vec{n})) depends on the whole template whereas \mathbf{B}_{TPL} depends only on the cycles of TPL. The expression $\vec{d}_{\text{TPL}} \cdot \vec{n} + h_{\text{TPL}}$ accounts for that difference.

Essentially the same line of reasoning can be applied if the connecting paths in TPL are non-empty. However, the argument becomes more technical. We present it in full detail in the appendix.

Now, we discuss how to check whether there exists a negative template. One of the conditions of negativity is that $GAIN(TPL(t\vec{n}_1 + \vec{n}_2)) \cdot VALS(TPL(t\vec{n}_1 + \vec{n}_2)) = 0$. Recall that GAIN and VALS depend on the multiset of transitions, but not on the order of transitions. Therefore, for a given template TPL, we define $TRANS_{TPL}(\vec{n}) \in \mathbb{N}^m$, where $m = |\delta|$, as the vector of multiplicities of transitions in $TPL(\vec{n})$. We will write $TRANS(\vec{n})$ if TPL is clear from the context.

Lemma 18. Let TPL be a template and let $\vec{n} \in \mathbb{N}^p$ be a vector of multiplicities. There exist $r_1, \ldots, r_\ell \in \mathbb{Q}^+$ and $\vec{z}_1, \ldots, \vec{z}_\ell \in \mathbb{N}^p$ such that

- (1) $supp(\vec{z_i}) \leq m$ (the number of transitions of $|\mathcal{A}|$),
- (2) there exists $t \in \mathbb{N}^+$ such that $TRANS(\vec{z_i}) = t \cdot TRANS(\vec{n})$, and
- (3) $\vec{n} = \sum_{i=1}^{\ell} r_i \vec{z}_i$.

Remark. The condition $\text{Trans}(\vec{z}_i) = t \cdot \text{Trans}(\vec{n})$ implies that if the cycle $\text{Tpl}(\vec{n})$ is balanced, then all the cycles $\text{Tpl}(\vec{z}_1), \ldots, \text{Tpl}(\vec{z}_\ell)$ are balanced as well.

Proof ideas. First, observe that TRANS is a linear function transforming vectors from \mathbb{N}^p into vectors from \mathbb{N}^m . The value p can be exponential w.r.t. $m = |\delta|$ and hence we show that each vector from \mathbb{N}^p can be presented as a linear combination over \mathbb{Q}^+ of vectors with polynomially-bounded supports.

The full proof has been relegated to the appendix.

Next, we show that if there exists a negative template, then there exists one of polynomial size.

Lemma 19. If there exists a negative template, then there exists one of polynomial size in |TPL| + |A| + |f|.

Proof ideas. Assume that a template TPL is positive (resp., negative). Let $\vec{n}_1, \vec{n}_2 \in \mathbb{N}^p$ be the vectors witnessing positivity (resp., negativity). Then, using Lemma 18, we show that there exist \vec{n}_1^0, \vec{n}_2^0 that witness template TPL being positive or negative and the support of both \vec{n}_1^0, \vec{n}_2^0 is bounded by $|\delta|$. We remove from TPL cycles corresponding to coefficient 0 in both \vec{n}_1^0 and \vec{n}_2^0 . We get a template with polynomially many cycles. If all connecting paths are still bounded by |Q| we terminate with TPL of polynomial size. However, as we remove some cycles, some connecting path are concatenated and in the result we get connecting paths longer than |Q|. For such connecting path, we extract simple cycles and group them (as in Lemma 13). We get another minimal template TPL' shorter than TPL, which is positive or negative. By iterating this process, we get a polynomial-size template TPL $^{\neq 0}$, which is positive or negative. Then, by Lemma 17 there exists a negative template of polynomial size.

The full proof has been relegated to the appendix.

Still, to check whether there exists a negative template we have to solve a system consisting of a quadratic inequality $\vec{n}_1^T \mathbf{B}_{\text{TPL}} \vec{n}_1 < 0$ and a quadratic equation, which corresponds to GAIN(TPL($t\vec{n}_1 + \vec{n}_2$)) · VALS(TPL($t\vec{n}_1 + \vec{n}_2$)) = 0. We show that this quadratic equation can be transformed into a system of linear inequalities.

Lemma 20. Let TPL be a template. There exist systems of linear equations and inequalities S_1, \ldots, S_l such that (1) each S_i has polynomial size in $|\text{TPL}| + |\mathcal{A}| + |f|$, (2) for all $\vec{n} \in \mathbb{N}^p$ we have $\text{GAIN}(\text{TPL}(\vec{n})) \cdot \text{VALS}(\text{TPL}(\vec{n})) = 0$ if and only if for some i the vector \vec{n} satisfies S_i .

Proof ideas. We construct a matrix $\mathbf{A} \in \mathbb{Z}^{(p+1)\times(p+1)}$ such that for every $\vec{n} \in \mathbb{N}^p$ we have $\operatorname{GAIN}(\operatorname{Tpl}(\vec{n})) \cdot \operatorname{VALS}(\operatorname{Tpl}(\vec{n})) = 0$ if and only if for $\vec{n}_1 = (n[1], \dots, n[p], 1)$ we have $\vec{n}_1^T \mathbf{A} \vec{n}_2 = 0$. Moreover, we know that for all $\vec{n}_1 \in \mathbb{N}^{p+1}$ we have $\vec{n}_1^T \mathbf{A} \vec{n}_1 \geq 0$. We carry out standard elimination of quadratic terms using the quadratic formula on the successive variables in $\vec{n}_1^T \mathbf{A} \vec{n}_1$. Such a procedure involves computing squares in each step $(\Delta = b^2 - 4ac)$ and hence it can lead to coefficients of double-exponential order in p+1 (the number of dimensions). However, due to condition $\vec{n}_1^T \mathbf{A} \vec{n}_1 \geq 0$ for all $\vec{n}_1 \in \mathbb{N}^{p+1}$, we can show that for every i either x[i] = 0 or x[i] has to be a double root $(\Delta = 0)$ and hence it can be described with a simple equation, which is of polynomial size.

The full proof has been relegated to the appendix.

Finally, Lemma 19 and Lemma 20 imply that existence of a negative template can be solved in NP via reduction to integer quadratic programming.

Lemma 21. We can verify in NP whether a given $VASS(\mathbb{Z}, k)$ \mathcal{A} has a negative template.

Proof sketch. We non-deterministically pick a template TPL of polynomial size (Lemma 19). Then, we non-deterministically pick a system S_i for template (Lemma 20). We make two copies of S_i : S_i^1 and S_i^2 ; in one we substitute \vec{n} with \vec{n}_2 and in the other with $\vec{n}_1 + \vec{n}_2$. It follows that for all $t \in \mathbb{R}$ the vector $t\vec{n}_1 + \vec{n}_2$ satisfies S_i (substituted for \vec{n}). Finally, we solve an instance of integer quadratic programming consisting of $\vec{n}_1 \mathbf{B}_{\text{TPL}} \vec{n}_1 - 1 \leq 0$ and linear equations and inequalities S_i^1 and S_i^2 , which is an instance of integer quadratic programming and hence can be solved in NP [13].

4.2.3 The linear case

We consider the final case, where all templates are linear. The decision procedure described in the following lemma answers YES (in at least one of non-deterministic computations) whenever the answer to the regular average-value problem with threshold 0 is YES and all templates are linear. Thus, it is complete.

Our main algorithm assumes that all templates are linear if it fails to find a negative template. The failure can be due to a wrong non-deterministic pick. Having that in mind, we make sure that the decision procedure from the following lemma is sound regardless of the linearity of templates, i.e., if it answers YES, then the answer to the regular average-value problem with threshold 0 is YES.

Lemma 22. Assume that all templates are linear. Then, we can solve the regular average-value problem with threshold 0 in non-deterministic polynomial time. Moreover, the procedure is sound irrespectively of the linearity assumption.

Proof sketch. First, we show that using Lemma 18, if there is a cycle ρ with (a) $GAIN(\rho) \cdot VALS(\rho) = 0$ and (b) $Sum_{\vec{g}}(\rho) \leq 0$, then there is a cycle ρ' defined by a template of polynomial size satisfying both conditions (a) and (b).

Assume that for some $\vec{g} \in \mathbb{Z}^k$ we have

- 1. $\operatorname{\mathsf{Sum}}_{\vec{a}}(\rho) < 0$ and
- 2. $GAIN(\rho) \cdot VALS(\rho) = 0$.

Then, there exists a minimal template TPL and a vector of multiplicities \vec{n} such that TPL(\vec{n}) = ρ (Lemma 13). As TPL is linear, then we can compute in polynomial time $\vec{d}_{\text{TPL}} \in \mathbb{Z}^p$, $h_{\text{TPL}} \in \mathbb{Z}$, which are polynomial in $|\text{TPL}| + |\mathcal{A}| + |f|$ such that

$$2 \cdot \mathsf{Sum}_{\vec{q}}(\mathrm{TPL}(\vec{n})) = (\vec{c}_{\mathrm{TPL}} + \vec{d}_{\mathrm{TPL}}) \cdot \vec{n} + e_{\mathrm{TPL}} + h_{\mathrm{TPL}} + 2 \cdot \vec{g} \cdot \mathrm{VALS}(\mathrm{TPL}(\vec{n}))$$

We apply Lemma 18 to vector \vec{n} and get $\vec{n} = \sum_{i=1}^{\ell} r_i \vec{z}_i$, with \vec{z}_i, r_i satisfying the statement of Lemma 18. Observe that for some i, we have $\mathsf{Sum}_{\vec{g}}(\mathsf{TPL}(\vec{z}_i)) \leq 0$. As $\mathsf{TRANS}(\vec{z}_i) = t \cdot \mathsf{TRANS}(\vec{n})$, we have

$$GAIN(TPL(\vec{z_i})) \cdot VALS(TPL(\vec{z_i})) = t^2 \cdot (GAIN(TPL(\vec{n})) \cdot VALS(TPL(\vec{n}))) = 0$$

Therefore, $\text{TPL}(\vec{z_i})$ is a balanced cycle with $\text{Sum}_{\vec{q}}(\text{TPL}(\vec{z_i})) \leq 0$. As $supp(\vec{z_i}) \leq |\delta|$, we remove from TPL cycles that correspond to 0 components in $\vec{z_i}$. Again, TPL may not be minimal so, as in the proof of Lemma 13, we extract simple cycles and iterate the process until we get TPL' of polynomial size such that TPL' with some multiplicities defines a cycle ρ' , which satisfies $\text{Sum}_{\vec{q}}(\rho') \leq 0$ and $\text{GAIN}(\rho') \cdot \text{VALS}(\rho') = 0$.

Second, consider a template TPL of polynomial size. We nondeterministically pick a subset of states Q and write a system of equations S^Q_{GAIN} over variables \vec{x}, \vec{y} such that \vec{x}, \vec{y} is a solution of S^Q_{GAIN} if and only if there exists a path ρ_1 satisfying (a) \vec{x} are multiplicities of transitions along ρ_1 (b) ρ_1 is from some initial state of \mathcal{A} to the first state of TPL, (c) ρ_1 visits all states from Q, (d) $\vec{y} = \text{GAIN}(\rho_1)$. To write S^Q_{GAIN} we specify that all states from Q have equal in-degree and out-degree except for the initial state of \mathcal{A} , in which out-degree is greater than in-degree by 1, and the first state of TPL, where in-degree is greater than out-degree by 1.

In summary, if all templates are linear, then there exist ρ_1, ρ_2 defining a regular computation of the average value at most 0 if and only if the following holds: there is a subset of states Q and a template TPL of polynomial size such that the system of inequalities consisting of S_{GAIN}^Q and the inequality

$$\left(\vec{c}_{\text{TPL},\vec{0}} + \vec{d}_{\text{TPL}}\right) \cdot \vec{n} + e_{\text{TPL},\vec{0}} + h_{\text{TPL}} + 2 \cdot \vec{y} \cdot \text{VALS}(\text{TPL}(\vec{n})) \le 0 \tag{4}$$

has a solution over natural numbers. Note that all the components except for $\vec{y} \cdot \text{VALS}(\text{Tpl}(\vec{n}))$ are linear. Since \vec{y} and \vec{n} are variables, the component $\vec{y} \cdot \text{VALS}(\text{Tpl}(\vec{n}))$ is quadratic. Still, S_{GAIN}^Q with (4) is an instance of integer quadratic programming, which can be solved in NP [13].

We have assumed that TPL is linear. However, we cannot check that deterministically in polynomial time. Still, linearity of TPL is necessary only for the completeness of the above procedure. We can make sure that the answer is sound irrespectively of that assumption. Having a solution of the system, we can compute in polynomial time $GAIN(\rho_1)$ and $Sum_{GAIN(\rho_1)}(TPL(\vec{n}))$ and then verify $Sum_{GAIN(\rho_1)}(TPL(\vec{n})) \leq 0$.

4.2.4 Summary

We present a short summary of the non-deterministic procedure deciding whether a given VASS(\mathbb{Z}, k) \mathcal{A} has a regular computation of the value at most 0. We assume that all states in \mathcal{A} are reachable from initial states.

- Step 1. We check whether there is a cycle ρ with $GAIN(\rho) \cdot VALS(\rho) < 0$. We can do it in non-deterministic polynomial time (Lemma 9). If the answer is YES, then the answer to the average-value problem is YES (Lemma 7). Otherwise, we proceed.
- Step 2. We check whether there exists a negative template in \mathcal{A} . We can do it in non-deterministic polynomial time (Lemma 21). If the answer is YES, then the answer to the regular average-value problem is YES (Lemma 15). Otherwise, we proceed.
- Step 3. Assuming that the previous steps failed, all templates are linear. Then, we can solve the regular average-value problem in non-deterministic polynomial time (Lemma 22). Note that Step 2, could have failed due to unfortunate non-deterministic pick. However, the procedure from Lemma 22 is sound regardless of the linearity assumption.

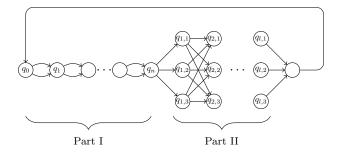


Figure 2: The structure of \mathcal{A}_{φ}

In consequence, we have the main result of this section:

Lemma 23. The regular average-value problem for $VASS(\mathbb{Z})$ with (general) cost functions is in NP.

4.3 Hardness

We show that the regular finite-value and the regular average-value problems are NP-hard. The proof is via reduction from the 3-SAT problem. Given a 3-CNF formula φ over n variables, we construct a VASS of dimension 2n, where dimensions correspond to literals in φ . Each simple cycle ρ in the VASS consists of two parts: The first part corresponds to picking a substitution σ , which is stored in the vector $GAIN(\rho)$. The second part ensures that $GAIN(\rho) \cdot VALS(\rho) = 0$ if σ satisfies φ and it is strictly positive otherwise. Therefore, if φ is satisfiable, the VASS has a regular run of the average cost 0, and otherwise all its regular runs have infinite average cost. In consequence, we have the following:

Lemma 24. The regular finite-value and the regular average-value problems for $VASS(\mathbb{Z})$ with (general) cost functions are NP-hard.

Proof. Let φ be a propositional formula in 3-CNF with n variables and l clauses. We construct a VASS($\mathbb{Z}, 2n$) \mathcal{A}_{φ} such that (1) if φ is satisfiable, then \mathcal{A}_{φ} has a regular computation of value 0, and (2) otherwise, all regular computations of \mathcal{A}_{φ} have infinite value.

The VASS \mathcal{A}_{φ} , presented in Figure 2, consists of two parts. The first part consists of the states q_0, \ldots, q_n , where q_0 is initial in \mathcal{A}_{φ} . It corresponds to picking a variable substitution. For every $i \in \{0, \ldots, n-1\}$, there are two transitions $(q_{i-1}, q_i, \mathbf{1}_i)$ and $(q_{i-1}, q_i, \mathbf{1}_{i+n})$, i.e., transitions from q_{i-1} to q_i incrementing the i-th or (i+n)-th counter. Upon reaching q_n , the vector of counter values \vec{g} has exactly n components with value 1 while other have value 0. The vector \vec{g} defines variable valuation $\sigma_{\vec{g}}$ defined as follows. A variable p_j is valuated to true, if g[j] = 1, and it is false if g[j+n] = 1. The cost function f in the first part is 0 for all states.

In the second part the values of counters do not change, but the cost function f is non-zero. We use this part to ensure that the average value is 0 if and only if $\sigma_{\vec{g}}$ satisfies φ . This part consists of l layers, each consisting of 3 states. At the j-th layer, corresponding to the clause $c_j = l_1 \vee l_2 \vee l_3$, we define the cost function f on states $q_{j,1}, q_{j,2}, q_{j,3}$ such that $f(q_{j,i}, \vec{g})$ returns the value of the component of \vec{g} corresponding to $\neg l_i$. It follows that $\sigma_{\vec{g}}$ satisfies c_j if and only if for one of these 3 states we have $f(q_{j,i}, \vec{g}) = 0$. Therefore, a single cycle over \mathcal{A} has the average 0 if and only if $\sigma_{\vec{g}}$ satisfies φ .

It follows that if φ has a satisfying valuation, then \mathcal{A}_{φ} has a regular computation of the average value 0. Conversely, if φ is unsatisfiable, then for every simple cycle ρ we have $GAIN(\rho) \cdot VALS(\rho) > 0$. Since all weights are non-negative, it follows that for all cycles ρ we have $GAIN(\rho) \cdot VALS(\rho) > 0$ and hence all regular computations of \mathcal{A}_{φ} have the value $+\infty$.

The main results of this section (Lemma 9, Lemma 23 and Lemma 24) summarize to Theorem 6. We leave the case of non-regular runs as an open question.

5 General cost functions and $VASS(\mathbb{N})$

We show that the average-value and the regular average-value problems are undecidable.

The proofs are via reduction from the halting problem for Minsky machines [19], which are automata with two natural-valued registers r_1, r_2 . There are two main differences between Minsky machines and VASS(N). First, the former can perform zero- and nonzero-tests on their registers, while the latter can take any transition as long as the counters' values remain non-negative. Second, the halting problem for Minsky machines is qualitative, i.e., the answer is YES or NO. We consider quantitative problems for VASS, where we are interested in the values assigned to computations. We exploit the quantitative features of our problems to simulate zero- and nonzero-tests.

Theorem 26. The average-value and the regular average-value problems for $VASS(\mathbb{N})$ with (general) cost functions are undecidable.

Proof sketch. We first sketch the reduction from the halting problem for Minsky machines to the *regular* average-value problem for $VASS(\mathbb{N})$. To ease the presentation, we use Büchi conditions, which can be eliminated.

Consider a Minsky machine \mathcal{M} and let I be its set of instructions. We construct a VASS(N) \mathcal{A} , which for presentation purposes has transitions labeled with the alphabet $I \cup \{\#\}$, that satisfies the following. Every regular computation labeled by $vu^{\omega} \in (I \cup \{\#\})^{\omega}$ has the average value 0 if and only if u is of the form $u_1\#\ldots u_k\#$ and all words u_1,\ldots,u_k encode terminating computations of \mathcal{M} . First, we can construct an automaton without counters, which verifies that for every subword of the form #v#, with $v \in I^*$, v encodes a terminating computations provided that all zero and non-zero test are correct. We can ensure that the cycle contains at least one transition labeled by # using a Büchi condition. This condition can be also enforced with additional counters. Such an automaton can be then encoded in the states of \mathcal{A} .

Second, to verify correctness of instructions w.r.t. the registers, we use counters c_1, c_2 in \mathcal{A} , which track the values of the registers minus 1, i.e., the value of c_i in \mathcal{A} is the maximum of 0 and the value of the *i*-th register r_i in \mathcal{M} . The VASS \mathcal{A} stores in its state whether the value of r_i is 0, 1 or the value of c_i plus 1. With that information, the VASS \mathcal{A} can verify correctness of the instructions of \mathcal{M} .

However, transitions of \mathcal{A} do not depend on the values of counters and hence it can be in a state denoting that the value of r_i is 0 or 1, while the value of c_i is greater than 0. To ensure that this is not the case, the cost function f returns the value of c_i whenever r_i is supposed to be 0 or 1. This suffices to prove the correctness in the regular average-value case.

Consider a regular computation π of the average 0 labeled with $v(u_1 \# \dots u_k \#)^{\omega}$. The values returned by the cost function are natural and hence for the average to by 0, the values at all positions need to be 0. It follows that whenever \mathcal{A} stipulates that the value of r_i is 0 or 1, then the value of c_i is 0. Thus, all zero-and non-zero tests in u_1, \dots, u_k are correct and hence these words encode terminating computations of \mathcal{M} . This concludes the reduction the the regular average-value problem for **VASS**(\mathbb{N}).

In the average-value problem however, a computation of the value 0 may be labeled with a word of the form $u_1\#u_2\#\ldots$, where the lengths of u_i 's grow fast. In that case we cannot infer that there exists i such that all values of f along u_i are 0. However, using two additional counters, we can ensure that all words u_i have length bounded by $|u_0|$. The first counter computes the value $|u_1| - |u_2| + |u_3| - \ldots$, while the second computes $|u_2| - |u_3| + |u_4| - \ldots$ Note that both counters are non-negative only if $|u_2| \le |u_1|$, $|u_3| \le |u_2|$, $|u_4| \le |u_1| - |u_2| + |u_3|$. The first three inequalities imply $|u_4| \le |u_1|$. We can show by induction that for all i we have $|u_i| \le |u_0|$. Therefore, the average-value is 0 only if there exists (exists infinitely many) words u_i such that all values along this word are 0. Then again, this word corresponds to a terminating computation of \mathcal{M} . Conversely, if \mathcal{M} has a terminating computation and u is its encoding, then \mathcal{A} has a computation of the average value 0 that is labeled with $(\#u)^{\omega}$.

Even though, the problems with an exact threshold are undecidable, we can decide existence of a regular computation of some finite value via reduction to reachability in VASS.

Theorem 27. The regular finite-value problem for $VASS(\mathbb{N})$ with (general) cost functions is decidable.

Proof sketch. Note that a VASS(\mathbb{N}, k) \mathcal{A} has a regular computation of some finite value if and only if there exist finite paths ρ_1, ρ_2 such that $\rho_1(\rho_2)^{\omega}$ corresponds to a computation and for all counters which influence the average in ρ_2 , the values of this counters do not change in a full iteration over ρ_2 .

Assume that all counters influence ρ_2 . We reduce the finite-value problem to the configuration reachability problem for VASS($\mathbb{N}, 2k$). Such a VASS \mathcal{A}_{2k} first simulates \mathcal{A} maintaining two copies $r_{i,1}, r_{i,2}$ of each counter c_i . This phase of the \mathcal{A}_{2k} computation corresponds to ρ_1 . Then, \mathcal{A}_{2k} nondeterministically picks a position, at it stores the current state of \mathcal{A} and switches to the mode where it modifies only counters $r_{i,1}$. This phase corresponds to ρ_2 . Again, \mathcal{A}_{2k} non-deterministically picks the end of that phase and verifies that the last state is the same as the first of this phase. It remains to be verified that for all i we have $r_{i,1} = r_{i,2}$. This is done by allowing to decrement $r_{i,1}, r_{i,2}$ simultaneously. Then, the final value is $\vec{0}$ if and only if for all i, at the end of the second phase we have $r_{i,1} = r_{i,2}$.

The above argument relies on regularity of computations. We leave the general case as an open question.

Open question 28. *Is the finite-value problem for* $VASS(\mathbb{N})$ *decidable?*

6 VASS games

The problems we studied correspond to finding a suitable path is a configuration-graph of a VASS, which can be considered as a single-player game on a VASS. We briefly discuss a natural extension of our problems to two-player games on VASS, which are defined in a similar way to mean-payoff games on pushdown systems [10]. We consider infinite game arenas, in which the positions are the configurations of a VASS and moves correspond to transitions between configurations. The states of the VASS are partitioned into Player-1 and Player-2 states and it gives rise to the partition of configurations into Player-1 and Player-2 configurations. On such an arena we define the *cost* of each position as the cost of the configuration defined using the *cost function*. Finally, we consider two-player games with mean-payoff (cost) objectives; As consequences of our results we have the following:

- 1. As shown in Theorem 4, configuration reachability reduces to the average-value problem, and for $VASS(\mathbb{N})$ -games configuration reachability is undecidable [5]. Hence the average-value problem is undecidable for $VASS(\mathbb{N})$ -games.
- 2. Finite-state multidimensional total-payoff games straightforwardly reduce to the average-value problem for VASS(Z)-games; basically the counters track the total payoff and the general cost function specifies the multi-dimensional requirement. Finite-state multidimensional total-payoff games are undecidable [7], and hence the average-value problem for VASS(Z)-games is undecidable.

7 Conclusion

In this work we consider VASS with long-run average properties where the cost depends linearly on the values of the counters. For state-dependent and state-independent linear combinations we establish decidability and complexity results for VASS with integer-valued and natural-valued counters. We leave open some questions (see Open questions 5, 25, and 28) which are interesting directions for future work.

Acknowledgment

We would like to thank Petr Novotn for discussion on literature on VASS and the long-run average property.

References

- [1] P. A. Abdulla, M. F. Atig, P. Hofman, R. Mayr, K. N. Kumar, and P. Totzke. Infinite-state energy games. In *CSL-LICS 2014*, pages 7:1–7:10, 2014.
- [2] R. Bloem, S. Jacobs, A. Khalimov, I. Konnov, S. Rubin, H. Veith, and J. Widder. Decidability in parameterized verification. *SIGACT News*, 47(2):53–64, 2016.
- [3] P. Bouyer, N. Markey, M. Randour, K. G. Larsen, and S. Laursen. Average-energy games. *Acta Inf.*, 55(2):91–127, 2018.
- [4] T. Brázdil, K. Chatterjee, A. Kucera, P. Novotný, D. Velan, and F. Zuleger. Efficient algorithms for asymptotic bounds on termination time in VASS. In *LICS 2018*, pages 185–194, 2018.
- [5] T. Brázdil, P. Jancar, and A. Kucera. Reachability games on extended vector addition systems with states. In ICALP 2010, pages 478–489, 2010.
- [6] T. Brázdil, S. Kiefer, A. Kucera, and P. Novotný. Long-run average behaviour of probabilistic vector addition systems. In LICS 2015, pages 44–55, 2015.
- [7] K. Chatterjee, L. Doyen, M. Randour, and J. Raskin. Looking at mean-payoff and total-payoff through windows. *Inf. Comput.*, 242:25–52, 2015.
- [8] K. Chatterjee, T. A. Henzinger, and J. Otop. Nested weighted limit-average automata of bounded width. In MFCS 2016, pages 24:1–24:14, 2016.
- [9] K. Chatterjee, T. A. Henzinger, and J. Otop. Quantitative monitor automata. In SAS 2016, pages 23–38, 2016.
- [10] K. Chatterjee and Y. Velner. The complexity of mean-payoff pushdown games. J. ACM, 64(5):34:1–34:49, 2017.
- [11] K. Chatterjee and Y. Velner. Hyperplane separation technique for multidimensional mean-payoff games. J. Comput. Syst. Sci., 88:236–259, 2017.
- [12] W. Czerwinski, S. Lasota, R. Lazic, J. Leroux, and F. Mazowiecki. The reachability problem for petri nets is not elementary. In *STOC*, pages 398–406, 2019.
- [13] A. Del Pia, S. S. Dey, and M. Molinaro. Mixed-integer quadratic programming is in np. *Mathematical Programming*, 162(1-2):225–240, 2017.
- [14] E. D'Osualdo, J. Kochems, and C. H. L. Ong. Automatic verification of erlang-style concurrency. In F. Logozzo and M. Fähndrich, editors, SAS 2013, pages 454–476, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [15] J. Esparza. Decidability and complexity of petri net problems introduction. Lectures on Petri nets I: Basic models, pages 374–428, 1998.
- [16] J. Esparza and M. Nielsen. Decidability issues for petri nets a survey. Bulletin of the European Association for Theoretical Computer Science, 52:245–262, 1994.
- [17] Y. Feng, R. Martins, Y. Wang, I. Dillig, and T. W. Reps. Component-based synthesis for complex apis. In POPL 2017, POPL 2017, pages 599–612, New York, NY, USA, 2017. ACM.
- [18] P. Ganty and R. Majumdar. Algorithmic verification of asynchronous programs. *ACM Trans. Program. Lang. Syst.*, 34(1):6:1–6:48, May 2012.
- [19] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

- [20] A. Kaiser, D. Kroening, and T. Wahl. Dynamic cutoff detection in parameterized concurrent programs. In *CAV 2010*, pages 645–659, 2010.
- [21] A. Kaiser, D. Kroening, and T. Wahl. Efficient coverability analysis by proof minimization. In M. Koutny and I. Ulidowski, editors, CONCUR 2012, pages 500–515, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [22] R. M. Karp and R. E. Miller. Parallel program schemata. J. Comput. Syst. Sci., 3(2):147–195, 1969.
- [23] S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA, pages 267–281, 1982.
- [24] J. Lambert. A structure to decide reachability in petri nets. Theor. Comput. Sci., 99(1):79–104, 1992.
- [25] J. Leroux. Vector addition systems reachability problem (a simpler solution). In *EPiC*, volume 10, pages 214–228. Andrei Voronkov, 2012.
- [26] J. Leroux. Polynomial vector addition systems with states. In ICALP 2018, pages 134:1–134:13, 2018.
- [27] R. Lipton. The reachability problem requires exponential space. Technical report 62, Yale, 1976.
- [28] E. W. Mayr. An Algorithm for the General Petri Net Reachability Problem. In STOC 1981, pages 238–246, 1981.
- [29] J. Michaliszyn and J. Otop. Average stack cost of büchi pushdown automata. In *FSTTCS 2017*, pages 42:1–42:13, 2017.
- [30] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223 231, 1978.
- [31] M. Sinn, F. Zuleger, and H. Veith. A simple and scalable static analysis for bound analysis and amortized complexity analysis. In *CAV*, pages 745–761, 2014.

8 Full proofs of selected lemmas

We prove Lemma 7 as a consequence of the proof of Lemma 11, and hence we show this after the proof of Lemma 11.

Lemma 11. There exists a regular computation π with $LimAvg(f(\pi)) \leq 0$ if and only if there exist paths ρ_1, ρ_2 such that

- (C1) $\rho_1(\rho_2)^{\omega}$ is an infinite path and ρ_2 is a cycle,
- (C2) $GAIN(\rho_2) \cdot VALS(\rho_2) = 0$, and
- (C3) $Sum_{GAIN(\rho_1)}(\rho_2) \leq 0.$

Proof. Consider a regular computation π let $\rho_1(\rho_2)^\omega$ be the path corresponding to π . Assume that $\operatorname{LimAvG}(f(\pi)) \leq 0$. Then, there is a sequence of positions n such that $\frac{1}{n} \sum_{i=1}^n f(\pi[i])$ converges to 0. Without loss of generality, we assume these positions are aligned with full occurrences of ρ_2 , i.e., $n = |\rho_1| + \ell \cdot |\rho_2|$. To see that, consider $r \in \{0, \dots, |\rho_2| - 1\}$ such that the infinite sequence of n's has an infinite subsequence with the same remainder r modulo $|\rho_2|$. Then, we can extend ρ_1 to ρ_1' so that $|\rho_1'|$ modulo $|\rho_2|$ is r and rotate the cycle ρ_2 .

Therefore, we consider the sequence of partial averages

$$\frac{1}{n}\sum_{i=1}^{n} f(\pi[i]) = \frac{1}{n} \left(\sum_{i=1}^{|\rho_1|} f(\pi[i]) + \sum_{i=|\rho_1|+1}^{n} f(\pi[i])\right).$$

where $n = |\rho_1| + \ell \cdot |\rho_2|$. First, observe that as n tends to infinity, $\frac{1}{n} (\sum_{i=1}^{|\rho_1|} f(\pi[i]))$ converges to 0. Therefore, $\frac{1}{n} \sum_{i=|\rho_1|+1}^n f(\pi[i])$ converges to 0, and we study this term.

Let $m = |\rho_2|$ and $\rho_2 = (q_1, q_2, \vec{y}_1) \dots (q_m, q_1, \vec{y}_m)$. Observe that the value of the counters at the position $|\rho_1| + \ell \cdot |\rho_2| + i$ in π equals

$$GAIN(\rho_1) + \ell \cdot GAIN(\rho_2) + \sum_{i=1}^{i-1} \vec{y_i},$$

and the state at this position is q_i . Let $l:Q\to\mathbb{N}^k$ be the labeling defining f. Then, we have

$$\sum_{i=1}^{m} f(\pi[|\rho_1| + (\ell m) + i]) = \sum_{i=1}^{m} \left(\text{GAIN}(\rho_1) + \ell \text{GAIN}(\rho_2) + \sum_{j=1}^{i-1} \vec{y_j} \right) \cdot l(q_i)$$

Note that

$$\sum_{i=1}^{m} l(q_i) = \text{VALS}(\rho_2)$$

and hence

$$\sum_{i=1}^{m} f(\pi[|\rho_1| + (\ell m) + i]) = \operatorname{GAIN}(\rho_1) \cdot \operatorname{VALS}(\rho_2) + \ell \cdot \operatorname{GAIN}(\rho_2) \cdot \operatorname{VALS}(\rho_2) + \sum_{i=1}^{m} \left(\sum_{j=1}^{i-1} \vec{y_j}\right) \cdot l(q_i)$$
 (5)

Since $GAIN(\rho_2) \cdot VALS(\rho_2) = 0$ (Lemma 7) we have

$$\sum_{i=1}^{\ell m} f(\pi[|\rho_1|+i]) = \ell \mathsf{Sum}_{\mathsf{GAIN}(\rho_1)}(\rho_2).$$

Since $\sum_{i=1}^{\ell m} f(\pi[|\rho_1|+i])$ tends to 0 as $\ell \to \infty$ we have $\mathsf{Sum}_{\mathsf{GAIN}(\rho_1)}(\rho_2) \le 0$.

Therefore, LIMAVG $(f(\pi)) \leq 0$ implies that $\mathsf{Sum}_{\mathsf{GAIN}(\rho_1)}(\rho_2) \leq 0$. Observe that the converse implication holds as well.

As the consequence of the above proof we have the following:

Lemma 7. Let π be a regular computation corresponding to a path $\rho_1(\rho_2)^{\omega}$. Then, one of the following holds:

- 1. $GAIN(\rho_2) \cdot VALS(\rho_2) < 0$ and $LIMAVG(f(\pi)) = -\infty$, or
- 2. $GAIN(\rho_2) \cdot VALS(\rho_2) = 0$ and $LIMAVG(f(\pi))$ is finite, or
- 3. $GAIN(\rho_2) \cdot VALS(\rho_2) > 0$ and $LIMAVG(f(\pi)) = \infty$.

Proof. This follows directly from the equation (5):

- 1. if $GAIN(\rho_2) \cdot VALS(\rho_2) < 0$, then $\lim_{\ell \to \infty} \sum_{i=1}^m f(\pi[|\rho_1| + (\ell m) + i]) = -\infty$ and hence $LIMAVG(f(\pi)) = -\infty$,
- 2. if $GAIN(\rho_2) \cdot VALS(\rho_2) = 0$, then $\sum_{i=1}^m f(\pi[|\rho_1| + (\ell m) + i])$ is independent of ℓ , and hence $LIMAVG(f(\pi)) = \sum_{i=1}^m f(\pi[|\rho_1| + i])$, which is finite, and
- 3. if $GAIN(\rho_2) \cdot VALS(\rho_2) > 0$ $\lim_{\ell \to \infty} \sum_{i=1}^m f(\pi[|\rho_1| + (\ell m) + i]) = \infty$ and hence $LIMAVG(f(\pi)) = \infty$.

Lemma 14. Given a template TPL we can compute in polynomial time in $|\text{TPL}| + |\mathcal{A}| + |f|$, a symmetric matrix $\mathbf{B}_{\text{TPL}} \in \mathbb{Z}^{p \times p}$, $\vec{c}_{\text{TPL}} \in \mathbb{Z}^p$ and $e_{\text{TPL}} \in \mathbb{Z}$ such that the following holds:

$$2 \cdot \mathsf{Sum}_{\vec{0}}(\mathsf{TPL}(\vec{n})) = \vec{n}^T \mathbf{B}_{\mathsf{TPL}} \vec{n} + \vec{c}_{\mathsf{TPL}} \vec{n} + e_{\mathsf{TPL}}$$
(2)

Moreover, for all $i, j \in \{1, ..., p\}$ we have

$$\mathbf{B}_{\text{TPL}}[i,j] = \text{GAIN}(\beta_{\min(i,j)}) \cdot \text{VALS}(\beta_{\max(i,j)}). \tag{3}$$

Proof. First, observe that for all $\vec{q}, \vec{h} \in \mathbb{Z}^k$ we have

$$\begin{split} \operatorname{\mathsf{Sum}}_{\vec{g}}(\tau_1 \tau_2) &= \operatorname{\mathsf{Sum}}_{\vec{g}}(\tau_1) + \operatorname{\mathsf{Sum}}_{\vec{g} + \operatorname{GAIN}(\tau_1)}(\tau_2) \\ \operatorname{\mathsf{Sum}}_{\vec{g} + \vec{h}}(\tau) &= \vec{g} \cdot \operatorname{VALS}(\tau) + \operatorname{\mathsf{Sum}}_{\vec{h}}(\tau). \end{split}$$

Therefore,

$$\mathsf{Sum}_{\vec{0}}(\tau_1 \tau_2) = \mathsf{Sum}_{\vec{0}}(\tau_1) + \mathsf{Sum}_{\vec{0}}(\tau_2) + \mathsf{GAIN}(\rho_1) \cdot \mathsf{VALS}(\rho_2) \tag{**}$$

We repeatedly apply (**) to $Sum_{\vec{0}}(TPL(\vec{n}))$, where

$$\mathrm{Tpl}(\vec{n}) = \alpha_0 \beta_1^{n[1]} \alpha_1 \beta_2^{n[2]} \dots \beta_p^{n[p]} \alpha_p$$

and get

$$\begin{aligned} \operatorname{Sum}_{\vec{0}}(\operatorname{Tpl}(\vec{n})) &= \operatorname{Sum}_{\vec{0}}(\alpha_0) + \operatorname{Sum}_{\vec{0}}(\beta_1^{n[1]}) + \ldots + \operatorname{Sum}_{\vec{0}}(\beta_p^{n[p]}) + \operatorname{Sum}_{\vec{0}}(\alpha_p) + \\ &+ \operatorname{GAIN}(\alpha_0) \cdot \operatorname{VALS}(\beta_1^{n[1]}) + \operatorname{GAIN}(\alpha_0 \beta_1^{n[1]}) \cdot \operatorname{VALS}(\alpha_1) + \ldots + \operatorname{GAIN}(\alpha_0 \beta_1^{n[1]} \ldots \beta_p^{n[p]}) \cdot \operatorname{VALS}(\alpha_p) \end{aligned} \tag{6}$$

Observe that

$$GAIN(\beta_i^{n[i]}) = n[i]GAIN(\beta_i)$$

$$VALS(\beta_i^{n[i]}) = n[i]VALS(\beta_i).$$
(7)

Therefore, by applying (7) to (6) we get:

$$2\operatorname{Sum}_{\vec{0}}(\beta_i^{n[i]}) = 2n[i]\operatorname{Sum}_{\vec{0}}(\beta_i) + n[i](n[i] - 1)\operatorname{GAIN}(\beta_i) \cdot \operatorname{VALS}(\beta_i)$$
(8)

$$GAIN(\alpha_0 \beta_1^{n[1]} \dots \alpha_{i-1} \beta_i^{n[i]}) \cdot VALS(\alpha_i) = \sum_{j=0}^{i-1} GAIN(\alpha_j) VALS(\alpha_i) + \sum_{j=1}^{i} n[j] GAIN(\beta_j) VALS(\alpha_i)$$
(9)

$$\operatorname{GAIN}(\alpha_0 \beta_1^{n[1]} \dots \alpha_i) \cdot \operatorname{VALS}(\beta_{i+1}^{n[i+1]}) =$$

$$\sum_{j=0}^{i} n[i+1] \operatorname{GAIN}(\alpha_j) \operatorname{VALS}(\beta_{i+1}) + \sum_{j=1}^{i} n[j] n[i+1] \operatorname{GAIN}(\beta_j) \operatorname{VALS}(\beta_{i+1})$$
(10)

It follows that $2Sum_{\vec{0}}(TPL(\vec{n}))$ can be presented as a quadratic function of the form

$$2 \cdot \mathsf{Sum}_{\vec{0}}(\mathsf{TPL}(\vec{n})) = \vec{n}^T \mathbf{B}_{\mathsf{TPL}} \vec{n} + \vec{c}_{\mathsf{TPL}} \vec{n} + e_{\mathsf{TPL}}$$

$$\tag{11}$$

Finally, the quadratic terms occur in (8) and (10). All quadratic terms in (8) are of the form $n[i]^2 GAIN(\beta_i) \cdot VALS(\beta_i)$ and all quadratic terms in (10) are of the form $n[i]n[j]GAIN(\beta_j) \cdot VALS(\beta_i)$, where j < i. Therefore, the matrix \mathbf{B}_{TPL} is as in the statement.

Lemma 17. (1) There exists a negative template or all templates are linear. (2) If there exists a positive template TPL, then there exists a negative one of the size bounded by $|\text{TPL}|^2$. (3) If a template TPL is linear, we can compute \vec{d}_{TPL} , h_{TPL} in polynomial time in $|\text{TPL}| + |\mathcal{A}| + |f|$.

Proof. Consider a template TPL of the form

TPL =
$$(\alpha_0, \beta_1, \dots, \beta_p, \alpha_p)$$

and a vector \vec{n} such that $\text{TPL}(\vec{n})$ is balanced $(\text{GAIN}(\text{TPL}(\vec{n})) \cdot \text{VALS}(\text{TPL}(\vec{n})) = 0.)$ We define β_{p+1} as

$$\beta_{p+1} = \text{TPL}(\vec{0}) = \alpha_0 \alpha_1 \dots \alpha_p$$

and we define $\vec{m} \in \mathbb{Z}^{p+1}$ as n[i] = m[i] for $i \in \{1, \dots p\}$ and m[p+1] = 1. We define an extended template TPL₁ such that

$$TPL_1 = (\alpha_0, \beta_1, \dots, \beta_p, \alpha_p, \beta_{p+1}, \epsilon).$$

We distinguish three cases.

The case $\vec{m}^T \mathbf{B_{Tpl_1}} \vec{m} < 0$. We show that TPL₁ is negative. Consider vectors \vec{n}_2, \vec{n}_2 such that

$$\vec{n}_1 = (n[1], \dots, n[p], 1) = \vec{m}$$

 $\vec{n}_2 = (n[1], \dots, n[p], 0).$

Then, $\vec{n}_1^T \mathbf{B}_{\text{TPL}_1} \vec{n}_1 < 0$ and $\text{TPL}_1(t\vec{n}_1 + \vec{n}_2)$ has the same multiset of edges as $\text{TPL}(\vec{n})$ repeated t + 1 times. Therefore,

$$GAIN(TPL_1(t\vec{n}_1 + \vec{n}_2)) \cdot VALS(TPL_1(t\vec{n} + 1 + \vec{n}_2)) = (t+1) \cdot GAIN(TPL(\vec{n})) \cdot VALS(TPL(\vec{n})) = 0$$

and hence TPL_1 is negative.

The case $\vec{m}^T \mathbf{B_{Tpl_1}} \vec{m} > 0$. Similarly to the previous case, we can show that $\mathrm{TPL_1}$ is positive. However, we show that there exists a template $\mathrm{TPL_2}$, which is negative. To show that, we exhibit the connection between the equation $\mathrm{GAIN}(\mathrm{TPL}(\vec{n})) \cdot \mathrm{VALS}(\mathrm{TPL}(\vec{n})) = 0$ and $\vec{n}^T \mathbf{B_{Tpl_1}} \vec{n}$.

Since $\beta_{p+1} = \text{TpL}(0)$ and m[p+1] = 1 we have

$$Gain(Tpl(\vec{n})) = \sum_{i=1}^{p+1} m[i] \cdot Gain(\beta_i)$$

$$VALS(TPL(\vec{n})) = \sum_{i=1}^{p+1} m[i] \cdot VALS(\beta_i)$$

Therefore,

$$GAIN(TPL(\vec{n})) \cdot VALS(TPL(\vec{n})) = \sum_{1 \le i, j \le p+1} m[i][j]GAIN(\beta_i) \cdot VALS(\beta_j)$$

Observe that the last equation can be presented as a quadratic form, i.e., consider a matrix $\mathbf{C} \in \mathbb{Z}^{(p+1)\times (p+1)}$ defined as for $i, j \in \{1, \dots, p+1\}$ as

$$\mathbf{C}[i, j] = \mathrm{GAIN}(\beta_i) \cdot \mathrm{VALS}(\beta_i) + \mathrm{GAIN}(\beta_i) \cdot \mathrm{VALS}(\beta_i).$$

Then,

$$GAIN(TPL(\vec{n})) \cdot VALS(TPL(\vec{n})) = \vec{m}^T \mathbf{C} \vec{m} = 0.$$

Recall, that for all $1 \le i, j \le p+1$ we have

$$\mathbf{B}_{\text{Tpl}_1}[i,j] = \text{Gain}(\beta_{min(i,j)}) \cdot \text{Vals}(\beta_{max(i,j)})$$

Consider any template TPL₂ with the same cycles as TPL₁ but in the reverse order. Then, for all $1 \le i, j \le p+1$ we have

$$\mathbf{B}_{\text{TPL}_2}[i,j] = \text{GAIN}(\beta_{\max((p+2)-i,(p+2)-j)}) \cdot \\ \text{VALS}(\beta_{\min((p+2)-i,(p+2)-j)})$$

and hence

$$\mathbf{B}_{\mathrm{Tpl}_2}[(p+2)-i,(p+2)-j] = \mathrm{GAIN}(\beta_{\max(i,j)}) \cdot \mathrm{VALS}(\beta_{\min(i,j)}).$$

It follows that $\mathbf{B}_{\text{TPL}_2}$ equals $\mathbf{C} - \mathbf{B}_{\text{TPL}_1}$ transposed along anti-diagonal, i.e., for all $1 \leq i, j \leq p+1$ we have

$$(\mathbf{C} - \mathbf{B}_{\text{TPL}_1})[i, j] = \mathbf{B}_{\text{TPL}_2}[(p+2) - i, (p+2) - j]$$

Therefore, $\vec{m}^T(\mathbf{C} - \mathbf{B}_{\text{TPL}_1})\vec{m}$, which is less than 0, equals $\vec{m}_R^T \mathbf{B}_{\text{TPL}_2} \vec{m}_R$, where \vec{m}_R is the reversed vector \vec{m} .

It remains to show that there exists such TPL₂ of the size bounded by $|\text{TPL}_1|^2$. We define TPL₂ as β_{p+1} followed by p iterations of TPL($\vec{0}$) = $\alpha_0 \dots \alpha_p$ each with a signle occrrence of a cycle β_i in the order $\beta_p, \beta_p, \dots, \beta_1$. Formally, for $1 \le i \le j \le p+1$ we define

$$\alpha[i,j] = \alpha_i \alpha_{i+1} \dots \alpha_i$$

We define TPL_2 as follows:

$$TPL_2 := (\epsilon, \beta_{p+1}, \alpha[0, p-1], \beta_p, \alpha[p]\alpha[0, p-2], \dots, \alpha[2, p+1]\alpha[1], \beta_1, \alpha[1, p+1]),$$

Observe that the path $TPL_2(\vec{0}) = \beta_{p+1}^p$, i.e., it is the same as $TPL_1(\vec{0})^p = TPL(\vec{0})^p$. Therefore, the size of TPL_2 is bounded by $|TPL_1|^2$. The template TPL_2 is not minimal, but it has not been a requirement.

The vectors

$$\vec{k}_1 = p\vec{m}_R = (p, pn[p], \dots, pn[1]), \text{ and}$$

$$\vec{k}_2 = (0, pn[p], \dots, pn[1])$$

witness TPL₂ being negative. Indeed, first

$$(\vec{k}_1)^T \mathbf{B}_{\mathrm{TPL}_1}(\vec{k}_1) = p^2 \vec{m}_R^T \mathbf{B}_{\mathrm{TPL}_2} \vec{m}_R < 0.$$

Second, $\text{TPL}_2(t\vec{k}_1 + \vec{k}_2)$ has the same multiset of edges as $\text{TPL}(\vec{n})$ repeated p(t+1) times. Therefore, as in the previous case, we have

$$Gain(Tpl_2(t\vec{k}_1 + \vec{k}_2)) \cdot Vals(Tpl_2(t\vec{k}_1 + \vec{k}_2)) = 0.$$

It follows that TPL2 is negative.

The case $\vec{m}^T \mathbf{B_{Tpl_1}} \vec{m} = 0$. We show that TPL is linear. Observe that \mathbf{B}_{TPL} is obtained from \mathbf{B}_{TPL_1} by deletion of the (p+1)-th row and the (p+1)-th column. Therefore,

$$\vec{n}^T \mathbf{B}_{\text{TPL}} \vec{n} = \vec{m}^T \mathbf{B}_{\text{TPL}_1} \vec{m} - \sum_{i=1}^p \text{GAIN}(\beta_i) \cdot \text{VALS}(\beta_{p+1}) -$$

$$\sum_{i=1}^{p} \operatorname{GAIN}(\beta_{p+1}) \cdot \operatorname{VALS}(\beta_{i}) - \operatorname{GAIN}(\beta_{p+1}) \cdot \operatorname{VALS}(\beta_{p+1}).$$

Therefore, we define \vec{d}_{TPL} for $1 \leq i \leq p$ as

$$\vec{d}_{\text{TPL}}[i] = -(\text{GAIN}(\beta_i) \cdot \text{VALS}(\beta_{p+1}) + \text{GAIN}(\beta_{p+1}) \cdot \text{VALS}(\beta_i)),$$

and we define $h_{\text{TPL}} = -\text{GAIN}(\beta_{p+1}) \cdot \text{VALS}(\beta_{p+1})$. Since $\vec{m}^T \mathbf{B}_{\text{TPL}_1} \vec{m} = 0$, we get

$$\vec{n}^T \mathbf{B}_{\mathrm{TPL}} \vec{n} = \vec{d}_{\mathrm{TPL}} \cdot \vec{n} + h_{\mathrm{TPL}}.$$

Therefore, TPL is linear.

Finally, if no template is negative, then all templates are linear.

The following lemma is a vital part of the proof of Lemma 18. Recall that for a given template TPL, we define $\text{Trans}_{\text{TPL}}(\vec{n}) \in \mathbb{N}^m$, where $m = |\delta|$, as the vector of multiplicities of transitions in $\text{TpL}(\vec{n})$. The function $\text{Trans}_{\text{TPL}}$ is linear. We write $\text{Trans}(\vec{n})$ if TpL is clear from the context.

Lemma 29. Let TPL be a template. For every vector $\vec{n} \in \mathbb{N}^p$, there exists $\vec{n}_0 \in \mathbb{N}^p$ such that

- (1) $supp(\vec{n}_0) \leq m$,
- (2) $supp(\vec{n}_0) \subseteq supp(\vec{n})$, and
- (3) there exists $t \in \mathbb{N}^+$ such that $TRANS(\vec{n}_0) = tTRANS(\vec{n})$.

Proof. We define $\operatorname{Trans}^* \colon \mathbb{Q}^p \to \mathbb{Q}^m$ as the unique linear extension of Trans . Assume that $\operatorname{supp}(\vec{n}) > m$. Then, there are m+1 components i_1,\ldots,i_{m+1} of \vec{n} such that $n[i_1],\ldots,n[i_{m+1}]$ are non-zero and $\operatorname{Trans}^*(\mathbf{1}_{i_1}),\ldots,\operatorname{Trans}^*(\mathbf{1}_{i_{m+1}})$ are linearly dependent. It follows that $\{i_1,\ldots,i_{m+1}\}$ can be partitioned into I_1,I_2 and there exist non-zero vectors $\vec{y}_1,\vec{y}_2\in\mathbb{Q}^p$ with the support included in I_1 and I_2 respectively. such that $\operatorname{Trans}^*(\vec{y}_1) = \operatorname{Trans}^*(\vec{y}_2)$. Consider the maximal $c\in\mathbb{Q}^+$ such that all components of $\vec{x}=\vec{n}-c\vec{y}_1+c\vec{y}_2$ are non-negative, i.e., $\vec{x}\in(\mathbb{Q}^{\geq 0})^p$. Observe that $\operatorname{Trans}^*(\vec{x})=\operatorname{Trans}^*(\vec{n}), \operatorname{supp}(\vec{x})\subseteq\operatorname{supp}(\vec{n})$ and at least one component of \vec{n} becomes 0 in \vec{x} .

We iterate this process until we get a vector $\vec{x}_0 \in (\mathbb{Q}^{\geq 0})^p$ such that

- (1) $supp(\vec{x}_0) \leq m$,
- (2) $supp(\vec{x}_0) \subseteq supp(\vec{n})$, and
- (3) TRANS* $(\vec{x}_0) = \text{TRANS}^*(\vec{n})$.

Finally, let $t \in \mathbb{N}$ be such that $t\vec{x}_0 \in \mathbb{N}^p$. Observe that $\vec{n}_0 = t\vec{x}_0$ satisfies the statement.

Lemma 18. Let TPL be a template and let $\vec{n} \in \mathbb{N}^p$ be a vector of multiplicities. There exist $r_1, \ldots, r_\ell \in \mathbb{Q}^+$ and $\vec{z}_1, \ldots, \vec{z}_\ell \in \mathbb{N}^p$ such that

- (1) $supp(\vec{z_i}) \leq m$ (the number of transitions of $|\mathcal{A}|$),
- (2) there exists $t \in \mathbb{N}^+$ such that $TRANS(\vec{z_i}) = t \cdot TRANS(\vec{n})$, and

(3)
$$\vec{n} = \sum_{i=1}^{\ell} r_i \vec{z}_i$$
.

Proof. We show the lemma by induction on $|supp(\vec{n})|$. Clearly, if $|supp(\vec{n})| \leq m$ the statement holds.

Consider $\vec{n} \in \mathbb{N}^p$ and assume that for all $\vec{n}' \in \mathbb{N}^p$ with $|supp(\vec{n}')| < |supp(\vec{n})|$ the lemma holds. Let \vec{n}_0 be the vector from Lemma 29. Consider the maximal $r \in \mathbb{Q}^+$ such that $\vec{n} - r\vec{n}_0$ has all the components non-negative. Let $r = \frac{p}{q}$ and consider $\vec{k} = q\vec{n} - p\vec{n}_0$. Observe that $\vec{k} \in \mathbb{N}^p$ and $|supp(\vec{k})| < |supp(\vec{n})|$ and hence there exist $s_1, \ldots, s_\ell \in \mathbb{Q}^+$ and $\vec{y}_1, \ldots, \vec{y}_\ell \in \mathbb{N}^p$ such that for all $1 \le i \le \ell$

- (1) $supp(\vec{y_i}) \leq m$ (the number of transitions of $|\mathcal{A}|$),
- (2) there exist $t \in \mathbb{N}^+$ such that $TRANS(\vec{y_i}) = t \cdot TRANS(\vec{k})$, and
- (3) $\vec{k} = \sum_{j=1}^{\ell} s_j \vec{y}_j$.

Recall that there is $t_0 \in \mathbb{N}^+$ such that $Trans(\vec{n}_0) = t_0 Trans(\vec{n})$. It follows that

$$\operatorname{Trans}(\vec{k}) = (q - t_0) \operatorname{Trans}(\vec{n})$$

Therefore,

$$r_1 = \frac{1}{q}s_1, \dots, r_\ell = \frac{1}{q}s_\ell, r_{\ell+1} = \frac{p}{q}$$

and

$$\vec{z}_1 = \vec{y}_1, \dots, \vec{z}_\ell = \vec{y}_\ell, \vec{n}_0$$

satisfy the statement of this lemma.

Lemma 19. If there exists a negative template, then there exists one of polynomial size in $|\text{TPL}| + |\mathcal{A}| + |f|$.

Proof. Assume that a template TPL₁ is positive (resp., negative). First, we show that positivity (resp., negativity) of TPL₁ is witnessed by vectors \vec{k}_1, \vec{k}_2 with at most m non-zero components.

Let $\vec{n}_1, \vec{n}_2 \in \mathbb{N}^p$ be the vectors witnessing positivity (resp., negativity). Let $r_1, \ldots, r_\ell \in \mathbb{Q}^+$ and $\vec{z}_1, \ldots, \vec{z}_\ell$ be coefficients and vectors for \vec{n}_1 satisfying Lemma 18. Then, either for some i we have $\vec{z}_i \mathbf{B}_{\text{TPL}} \vec{z}_i \neq 0$, and we define \vec{p}_1 as \vec{z}_1^i or for some pair $i \neq j$

$$(r_i \vec{z}_i + r_j \vec{z}_j) \mathbf{B}_{\text{TPL}} (r_i \vec{z}_i + r_j \vec{z}_j) \neq 0$$

and we define $\vec{p}_1 = a(r_i\vec{z}_i + r_j\vec{z}_j)$, where a is a natural number such that $ar_i, ar_j \in \mathbb{N}$. Note that one of these conditions holds; if they both fail, then due to $\vec{n}_1 = \sum_{i=1}^{\ell} r_i\vec{z}_i$, we have $\vec{n}_1\mathbf{B}_{\mathrm{TPL}}\vec{n}_1 = 0$ contrary to the assumptions.

Let $\vec{p_2}$ be a vector with $supp(\vec{m_2}) \leq |\delta|$ and $Trans(\vec{p_2}) = tTrans(\vec{n_2})$ (which exists by Lemma 18). Finally, we define $\vec{k_1} = a\vec{p_1}$ and $\vec{k_1} = b\vec{p_2}$ for $a, b \in \mathbb{N}^+$ such that for some $t \in \mathbb{N}$ we have

TRANS
$$(\vec{k}_1) = t$$
TRANS (\vec{n}_1)
TRANS $(\vec{k}_2) = t$ TRANS (\vec{n}_2)

Observe that \vec{k}_1, \vec{k}_2 also witness template TPL₁ being positive or negative, but $|supp(\vec{k}_1)|, |supp(\vec{k}_2)| \leq m$.

We remove from TPL_1 cycles corresponding to coefficient 0 in both \vec{k}_1 and \vec{k}_2 . We get a template with polynomially many cycles. If all connecting paths are still bounded by |Q| we terminate with TPL_1 , of polynomial size. However, as we remove some cycles, some connecting path are concatenated and in the result we get connecting paths longer than |Q|. For such connecting path, we extract simple cycles and group them (as in Lemma 13). We get another minimal template TPL_2 shorter than TPL_1 , which is positive or negative. By iterating this process, we get a polynomial-size template TPL^{\neq} , which is positive or negative. Then, by Lemma 17 that there exists a negative template of polynomial size.

Lemma 20. Let TPL be a template. There exist systems of linear equations and inequalities S_1, \ldots, S_l such that (1) each S_i has polynomial size in $|\text{TPL}| + |\mathcal{A}| + |f|$, (2) for all $\vec{n} \in \mathbb{N}^p$ we have $\text{GAIN}(\text{TPL}(\vec{n})) \cdot \text{VALS}(\text{TPL}(\vec{n})) = 0$ if and only if for some i the vector \vec{n} satisfies S_i .

Proof. Recall the construction of TPL₁ from the proof of Lemma 17. We repeat this construction and define a symmetric matrix $\mathbf{A} \in \mathbb{Z}^{(p+1)\times (p+1)}$ such that for all $\vec{n} \in \mathbb{N}^p$ we have

$$2Gain(Tpl(\vec{n})) \cdot Vals(Tpl(\vec{n})) = \vec{n}_1^T \mathbf{A} \vec{n}_1$$

where $\vec{n}_1 = (n[1], n[2], \dots, n[p], 1)$. We define $\beta_{p+1} = \alpha_0 \alpha_1 \dots \alpha_p$ and the **A** is defined for all $1 \le i, j \le p+1$ as

$$\mathbf{A}[i, j] = \operatorname{GAIN}(\beta_i) \cdot \operatorname{VALS}(\beta_j) + \operatorname{GAIN}(\beta_j) \cdot \operatorname{VALS}(\beta_i).$$

We have assumed that \mathcal{A} does not have a regular run of the value $-\infty$ and hence for all $\vec{m} \in \mathbb{N}^{p+1}$ we have $\vec{m}^T \mathbf{A} \vec{m} \geq 0$. We eliminate variables using the standard quadratic formula. We start with m[1] and observe that

$$\vec{m}^T \mathbf{A} \vec{m} = A_1(m[1])^2 + B_1 m[1] + C_1$$

where $A_1 = \mathbf{A}[1,1], B_1 = 2\sum_{i=2}^{p+1} \mathbf{A}[1,i], \text{ and } C_1 = \vec{n}_0^T \mathbf{A} \vec{n}_0 \text{ where } \vec{m}_0 = (0, m[2], m[3], \dots, m[p+1]).$ We compute $\Delta = (B_1)^2 - 4A_1C_1$ and assume that $\Delta \geq 0$. Observe that there are four cases:

- The case: $A_1 = B_1 = 0$. We observe that this is possible only if for all i we have $\mathbf{A}[1,i] = \mathbf{A}[i,1] = 0$. Suppose that there is i such that $\mathbf{A}[1,i] \neq 0$. Then, since $B_1 = 0$ there is j such that $\mathbf{A}[1,j] \leq -1$. Consider $\vec{m}_0 \in \mathbb{N}^{p+1}$ such that $m_0[1] = \mathbf{A}[1,j]^2$, $m_0[j] = 1$ and all the other components are 0. Then, $\vec{m}_0^T \mathbf{A} \vec{m}_0 < 0$, which contradicts the assumptions.
- The case: $A_1 = 0$ and $B_1 \neq 0$. In this case, we argue that m[1] = 0. Indeed, we have $C_1 \geq 0$. Now, if $B_1 < 0$, then for $\vec{m}_t = (t, m[2], \dots, m[p+1])$ and $t \in \mathbb{N}^+$ big enough, we get $\vec{m}_t^T \mathbf{A} \vec{m}_t < 0$ contrary to the assumptions. Therefore, $B_1 > 0$. It follows that $B_1 m[1] + C_1 = 0$ if and only if m[1] = 0 and $C_1 = 0$.
- The case $A_1 \neq 0$ and $\Delta > 0$. Then, we argue that m[1] = 0. Indeed, if $\vec{x} \in (\mathbb{R}^{\geq 0})^{p+1}$ is such that x[1] > 0, then for some small $\epsilon \in (-\frac{x[1]}{2}, \frac{x[1]}{2})$ and the vector $\vec{x}_1 = (m[1] + \epsilon, m[2], \dots, m[p+1])$ we have $\vec{x}_1^T \mathbf{A} \vec{x}_1 < 0$. It follows that there exists $\vec{y} \in (\mathbb{Q}^{\geq 0})^{p+1}$ close to \vec{x}_1 such that $\vec{y}^T \mathbf{A} \vec{y} < 0$. Finally, for some $t \in \mathbb{N}$ we have $t\vec{y} \in \mathbb{N}^{p+1}$ and

$$(t\vec{y})^T \mathbf{A}(t\vec{y}) = t^2 (\vec{y}^T \mathbf{A} \vec{y}) < 0$$

which contradicts the assumption $\vec{m}^T \mathbf{A} \vec{m} \geq 0$.

• The case: $A \neq 0$ and $\Delta = 0$. Then, $m[1] = \frac{-B_1}{2A_1}$.

We have the same identities for all components of \vec{m} . Now, we construct linear inequalities S_i for \vec{n} .

First, without loss of generality we assume that **A** does not have $\vec{0}$ rows or columns. Such rows and columns in the symmetric **A** correspond to variables, which occur only with the 0 coefficient. Second, recall that we assumed that for $1 \le i \le p$ we have m[i] = n[i] and m[p+1] = 1. We substitute m[p+1] with 1 in all the equations on \vec{n} . Third, for every subset $P \subseteq \{1, \ldots, p\}$, we define S_P such that $P = \{i : n[i] = 0\}$.

Consider $P \subseteq \{1, \ldots, p\}$. We define S_P as

$$\begin{split} &\{n[i]=0\colon i\in P\} \cup \\ &\{\mathbf{A}[i,i]n[i]=-\big(\sum_{j\neq i}\mathbf{A}[i,j]n[j]\big)-\mathbf{A}[i,p+1]\colon i\notin P\} \end{split}$$

We assume that $i, j \in \{1, ..., p\}$ which is omitted for readability.

First, note that for every solution $\vec{n} \in \mathbb{N}$ to S_P we have $\vec{n}_1 = (n[1], n[2], \dots, n[p], 1)$ satisfies $\vec{n}_1^T \mathbf{A} \vec{n}_1 = 0$. We can remove all rows and columns with indexes from P as their contribution to $\vec{n}_1^T \mathbf{A} \vec{n}_1$ is 0. Let \mathbf{A}' be the matrix and \vec{k} be the vector resulting from removal of components from P. Then, we observe that $\mathbf{A}'\vec{k}$ is the $\vec{0}$ vector. Indeed, for every i the equation

$$\mathbf{A}[i,i]n[i] = \sum_{j \neq i} \mathbf{A}[i,j]n[j] + \mathbf{A}[i,p+1]$$

implies that the *i*-th component of $\mathbf{A}\vec{n}_1$ is 0. Thus, all components of the minor \mathbf{A}' with \vec{k} are 0.

Finally, we argue that for every \vec{m} satisfying $\vec{m}^T \mathbf{A} \vec{m} = 0$ with m[p+1] = 1, the vector $\vec{n} = (m[1], \dots, m[p])$ solves some S_P . Indeed, we consider $P = \{i : n[i] = 0\}$ and observe that due to the above considerations for all $i \in \{1, \dots, p\} \setminus P$ we need to have $2A_i n[i] = -B_i$ as it is stated in S_P .