Optimal Bounds for Multiweighted and Parametrised Energy Games

Line Juhl[1], Kim Guldstrand Larsen[1], and Jean-François Raskin[2]

[1] Aalborg University, Department of Computer Science, Denmark
{linej,kgl}@cs.aau.dk
[2] Université Libre de Bruxelles, Belgium
jraskin@ulb.ac.be

Abstract. Multiweighted energy games are two-player multiweighted games that concern the existence of infinite runs subject to a vector of lower and upper bounds on the accumulated weights along the run. We assume an unknown upper bound and calculate the set of vectors of upper bounds that allow an infinite run to exist. For both a strict and a weak upper bound we show how to construct this set by employing results from previous works, including an algorithm given by Valk and Jantzen for finding the set of minimal elements of an upward closed set. Additionally, we consider energy games where the weight of some transitions is unknown, and show how to find the set of suitable weights using the same algorithm.

1 Introduction

Energy games have recently attracted considerable attention [1–9]. An energy game is played by two players on a weighted game automaton. Player 1 wins if she has a strategy such that all infinite runs respecting this strategy has nonnegative accumulated weight at all times. A variant of energy games furthermore requires an upper bound that the accumulated weight must stay below at all times in order for Player 1 to win. The upper bound can also be weak, implying that all accumulated weights going above are simply truncated. As embedded systems are often resource-constrained systems exhibiting a reactive behaviour, energy games are relevant for ensuring that the resource of the system never becomes unavailable no matter the choices of the environment. Multiweighted energy games, where the weights of the automaton are vectors, are useful for modelling systems that depend on more than one resource.

In this paper we consider multiweighted energy games with unknown upper bound (both strict and weak) and fixed initial value. When considering the existence of a vector of upper bounds such that Player 1 is winning, it is from an engineering viewpoint relevant to construct the actual vector instead of giving a boolean answer to the problem. We therefore seek to construct the exact set of upper bounds that make Player 1 win the energy game. We will denote such upper bounds as winning. For both types of upper bounds it is clear that if some vector of upper bounds is winning, then also coordinate-wise larger vectors are

Z. Liu, J. Woodcock, and H. Zhu (Eds.): He Festschrift, LNCS 8051, pp. 244–255, 2013.

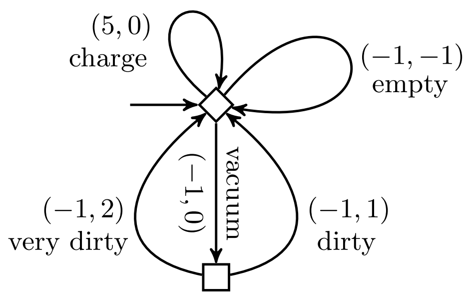
winning. In order to characterise the set of winning upper bounds, it is thus sufficient only to find the smallest vector of winning upper bounds. However, $\leq$ is not a total order on $\mathbb{Z}^k$ for $k > 1$, so instead of a unique smallest vector we search for the set of smallest incomparable vectors winning for Player 1.

To motivate the study, let us consider a small example of an automatic vacuum cleaner. The machine has a rechargeable battery and a container for the dust it collects. As we are interested in a behaviour that never empties the battery nor completely fills the dust container, it can be modelled using a 2-weighted energy game as seen in Fig. 1a. The vector attached to each transition denotes the change in battery (first coordinate) and container level (second coordinate). The diamond state is controlled by Player 1 while the square state is controlled by the environment, as the vacuum cleaner does not control how dirty the floor is when vacuuming.

The lower bound of the two resources are naturally 0, while the upper bound corresponds to the size of the battery and container, respectively. For a manufacturer it is useful to know what size she can possible make the battery and the container in order to ensure an infinite run. The set of minimal winning upper bounds consists in this case of the vectors $(6, 2)$ and $(5, 3)$. The upper bound vector $(6, 2)$ keeps the container as small as possible, while the upper bound vector $(5, 3)$ keeps the battery as small as possible. Surely, the first coordinate of an upper bound cannot be smaller than 5, as charging adds 5 to the accumulated weight in the first coordinate. Similarly, the second coordinate cannot be smaller than 2, as a very dirty floor adds 2 to the accumulated weight in the second coordinate. The winning strategy for Player 1 can be seen in Fig. 1b for $(6, 2)$ and Fig. 1c for $(5, 3)$. Any vector larger than one of the minimal vectors will also serve as a winning upper bound.

Contributions. For multiweighted energy games with an unknown upper bound (both strict and weak) and fixed initial value we calculate the set of minimal upper bounds such that the energy game is winning. For a strict upper bound we make use of results from [3] and [9] in order to construct the set, yielding an algorithm running in $2k$-exponential time. In the case of a weak upper bound we utilise an algorithm given by Valk and Jantzen in [10], that constructs the set of minimal elements of an upward-closed set (the so-called Pareto frontier), by showing that the preconditions for applying the algorithm are fulfilled. The relevant definitions are given in Section 2, while Section 3 and Section 4 treat the cases of a weak and a strict upper bound, respectively.

Furthermore we study a related problem in Section 5, where we consider multiweighted energy games where both the initial value and the upper bound (if any) are known, but where some weights of the transitions are unknown. We call these parametrised transitions. We here seek to characterise the set of possible evaluations for the parameters such that Player 1 can win the energy game. For a weak upper bound, it is again sufficient to construct the set of minimal evaluations such that Player 1 is winning, and we are once again able to apply the algorithm from [10] to construct the set.



(a) A vacuum cleaner 2-weighted game

if *battery* ≥ 1 and *container* ≥ 1
  then empty
else if *battery* ≥ 2 and *container* $= 0$
  then vacuum
else charge

(b) A winning strategy for Player 1 with upper bound $(6, 2)$

if *battery* $= 0$
  then charge
else if *battery* ≥ 2 and *container* ≤ 1
  then vacuum
else empty

(c) A winning strategy for Player 1 with upper bound $(5, 3)$

Fig. 1. A vacuum cleaner example

Related Work. Previously energy games have been considered in different settings. One-weighted energy games with both upper and lower bounds were defined in [2]. Here they study the existence of a winning strategy for Player 1 for a fixed initial value and fixed upper and lower bound and provide bounds on the complexity for the identified problems both in a timed and untimed setting. The paper [9] extends the results from [2] to the multiweighted case. The work of [7] treats multiweighted energy games with only a lower bound and show that deciding whether there exists a vector of initial values for the resources such that Player 1 can win the energy game is coNP-complete and that only finite-memory strategies are sufficient. In [3] they give a procedure running in $(k-1)$-exponential time that calculates the Pareto frontier of winning initial vectors in multiweighted energy games with k weights, a lower bound and unary weights on transitions (vector addition systems with states). For energy games with imperfect information and fixed initial value, the paper [8] proves decidability of the problem, but undecidability in case the initial value is not fixed.

2 Multiweighted Energy Games

In this paper, we let $\mathbb{Z}$ and $\mathbb{N}$ denote the sets of all integers and all nonnegative integers, respectively. We define $\mathbb{N}_\omega$ as $\mathbb{N} \cup \{\omega\}$, where ω is a new element modelling an arbitrary nonnegative integer. Thus $\omega > m$ for any $m \in \mathbb{N}$.

For two k-dimensional vectors $\bar{v}, \bar{v}' \in \mathbb{N}_\omega^k$ we use the notation $\bar{v}[i]$ to denote the ith coordinate of the vector $\bar{v}$ $(1 \leq i \leq k)$ and write $\bar{v} \leq \bar{v}'$ if $\bar{v}[i] \leq \bar{v}'[i]$ for all $i \in \{1, \ldots, k\}$. We define the sum of two vectors as the coordinate-wise sum, i.e. $\bar{v} + \bar{v}' = (\bar{v}[1] + \bar{v}'[1], \ldots, \bar{v}[k] + \bar{v}'[k])$. The notation $\bar{0} = (0, \ldots, 0)$ is used to denote the vector of all zeros and $\overline{\infty} = (\infty, \ldots, \infty)$ as the ditto for ∞.

A set $K \subseteq \mathbb{N}^k$ is said to be *upward closed* if $\bar{x} \in K$ and $\bar{x} \leq \bar{y}$ implies $\bar{y} \in K$. Furthermore we define $min(K)$ as the set of smallest incomparable vectors of K,

$$min(K) = \{\bar{x} \in K \mid \forall\ \bar{y}(\neq \bar{x}) \in K : \bar{y} \not\leq \bar{x}\}\ .$$

We call such a $min(K)$ the *minimal generating set* of K.

It it well-known that such a set of incomparable vectors of natural numbers will be finite (as stated in Dickson's lemma) and unique.

We now define a game with multiple weights as an automaton with dedicated state sets for each player and a transition function decorated with a vector of integers.

Definition 1. *A k-weighted game is a four-tuple $G = (S_1, S_2, s_0, \longrightarrow)$, where S_1 and S_2 are finite, disjoint sets of existential and universal states, respectively, $s_0 \in S_1 \cup S_2$ is the start state and $\longrightarrow \subseteq (S_1 \cup S_2) \times \mathbb{Z}^k \times (S_1 \cup S_2)$ is a finite multiweighted transition relation.*

We write $s \xrightarrow{\bar{w}} s'$ whenever $(s, \bar{w}, s') \in \longrightarrow$. In the following we consider only non-blocking automata, i.e. for every $s \in S_1 \cup S_2$ we have $s \xrightarrow{\bar{w}} s'$ for some $\bar{w} \in \mathbb{N}^k$ and $s' \in S_1 \cup S_2$.

Definition 2. *A configuration in a k-weighted game $G = (S_1, S_2, s_0, \longrightarrow)$ is a pair $(s, \bar{v})$ such that $s \in S_1 \cup S_2$ and $\bar{v} \in \mathbb{Z}^k$.*

A weighted run π in G restricted to a weak upper bound $\bar{b} \in (\mathbb{N} \cup \{\infty\})^k$ is an infinite sequence of configurations $(s_0, \bar{v}_0), (s_1, \bar{v}_1), \ldots$ such that for all $i \geq 0$ we have $s_i \xrightarrow{\bar{w}_i} s_{i+1}$ and $\bar{v}_{i+1}[j] = \min\{\bar{b}[j], \bar{v}_i[j] + \bar{w}_i[j]\}$ for all $j \in \{1, \ldots, k\}$.

By $\mathsf{WR}_{\bar{b}}(G)$ we denote all weighted runs in G with weak upper bound $\bar{b}$ starting from the initial state. Let π_i denote the ith configuration of a weighted run π.

As we are concerned with games we need a notion of a strategy for a player.

Definition 3. *A strategy for Player $i \in \{1, 2\}$ in a k-weighted game $G = (S_1, S_2, s_0, \longrightarrow)$ restricted to some weak upper bound $\bar{b}$ is a mapping σ assigning a configuration $(s, \bar{v})$ to any finite prefix of a weighted run in $\mathsf{WR}_{\bar{b}}(G)$ of the form $(s_0, \bar{v}_0), \ldots, (s_j, \bar{v}_j)$ where $s_j \in S_i$ such that $(s_0, \bar{v}_0), \ldots, (s_j, \bar{v}_j), (s, \bar{v})$ is a prefix of a weighted run in $\mathsf{WR}_{\bar{b}}(G)$.*

We say that a weighted run $(s_0, \bar{v}_0), (s_1, \bar{v}_1), \ldots$ respects a strategy σ of Player i if $\sigma((s_0, \bar{v}_0), \ldots, (s_j, \bar{v}_j)) = (s_{j+1}, \bar{v}_{j+1})$ for all $s_j \in S_i$.

We can now define the following three notions of winning vectors.

GL: Given a k-weighted game G, a vector $\bar{v}_0 \in \mathbb{N}^k$ *wins* the (multiweighted) energy game with lower bound (GL) if there exists a winning strategy σ for Player 1 such that any weighted run $(s_0, \bar{v}_0), (s_1, \bar{v}_1), \ldots \in \mathsf{WR}_{\overline{\infty}}(G)$ respecting σ satisfies $\bar{0} \leq \bar{v}_i$ for all $i \geq 0$.

GLW: Given a k-weighted game G, a vector $\bar{b} \in \mathbb{N}^k$ *wins* the (multiweighted) energy game with lower and weak upper bound (GLW) if there exists a winning strategy σ for Player 1 such that any weighted run $(s_0, \bar{0}), (s_1, \bar{v}_1), \ldots \in \mathsf{WR}_{\bar{b}}(G)$ respecting σ satisfies $\bar{0} \leq \bar{v}_i$ for all $i \geq 0$.

GLU: Given a k-weighted game G, a vector $\bar{b} \in \mathbb{N}^k$ *wins* the (multiweighted) energy game with lower and upper bound (GLU) if there exists a winning strategy σ for Player 1 such that any weighted run $(s_0, \bar{0}), (s_1, \bar{v}_1), \ldots \in \mathsf{WR}_{\overline{\infty}}(G)$ respecting σ satisfies $\bar{0} \leq \bar{v}_i \leq \bar{b}$ for all $i \geq 0$.

Notice that we may allow an initial weight vector $\bar{v}_0$ different from $\bar{0}$. This is evident by adding a new start state with one transition labelled with $\bar{v}_0$ pointing to the old start state.

Define $I = \{\bar{v}_0 \in \mathbb{N}^k \mid \bar{v}_0 \text{ wins GL}\}$, $W = \{\bar{b} \in \mathbb{N}^k \mid \bar{b} \text{ wins GLW}\}$ and $U = \{\bar{b} \in \mathbb{N}^k \mid \bar{b} \text{ wins GLU}\}$ as the winning vectors for GL, GLW and GLU, respectively. The paper [3] constructs the set $min(I)$ using $(k-1)$-exponential time for k-weighted energy games with unary weights.

This paper aims to construct the minimal generating sets of winning weak and strict upper bounds, $min(W)$ and $min(U)$.

Membership Problem. Another interesting question besides constructing the sets of winning vectors for a given game, is the question of membership; given a k-weighted game G and a vector $\bar{b} \in \mathbb{N}^k$ decide whether $\bar{b} \in W$ (or $\bar{b} \in I$ or $\bar{b} \in U$). The membership problem has been addressed in [9] among others. Table 1 (also found in [9]) gives a full overview of the so far obtained decidability and complexity results for the membership problem. In the table two further classifications of a k-weighted game $G = \{S_1, S_2, s_0, \longrightarrow\}$ are made. We say that a game G is existential if $S_2 = \emptyset$ (Player 1 controls all the states) and that G is universal if $S_1 = \emptyset$ (Player 2 controls all the states). The other subdivision concerns the number of weights, namely whether k is 1, fixed (and $k > 1$), or arbitrary.

Note that the complexity increases as more weights are added, apart from the universal case, where all problems lie in P. This is evident since any membership problem on a universal game with k weights can be solved by solving the same problem for each coordinate independently. Another thing to observe is that deciding membership in I is computationally easier than deciding membership in U in the 1-weighted case, even though membership in I is harder than U for an arbitrary number of weights. This stems from the fact that the configuration space for the problems concerning U (and W) is bounded due to the upper bounds, whereas the same a priori does not hold for the problems concerning I. The computational complexity of the problems concerning membership in W seems to follow the computationally easiest of the two other problems.

3 Weak Upper Bound

In this section we study the problem of finding the set $min(W)$ as defined in Section 2.

Table 1. Complexity bounds for the membership problem

Weights	Type	Existential	Universal	Game
One	$\in I$	$\in$ P [2]	$\in$ P [2]	$\in$ UP $\cap$ coUP [2]
	$\in W$	$\in$ P [2]	$\in$ P [2]	$\in$ NP $\cap$ coNP [2]
	$\in U$	NP-hard [2], $\in$ PSPACE [2]	$\in$ P [2]	EXPTIME-complete [2]
Fixed (k>1)	$\in I$	NP-hard [9], $\in$ k-EXPTIME [3]	$\in$ P [9]	EXPTIME-hard [9], $\in$ k-EXPTIME [3]
	$\in W$	NP-hard [9], $\in$ PSPACE [9] PSPACE-complete for $k \geq 4$ [9]	$\in$ P [9]	EXPTIME-complete [9]
	$\in U$	PSPACE-complete [9]	$\in$ P [9]	EXPTIME-complete [9]
Arbitrary	$\in I$	EXPSPACE-complete [9]	$\in$ P [9]	EXPSPACE-hard [9], decidable [3]
	$\in W$	PSPACE-complete [9]	$\in$ P [9]	EXPTIME-complete [9]
	$\in U$	PSPACE-complete [9]	$\in$ P [9]	EXPTIME-complete [9]

The paper [10] by Valk and Jantzen contains an algorithm for computing the minimal generating set of an upward closed set $K \subseteq \mathbb{N}^k$ provided that K satisfies a certain decidability criterion.

The decidability question is defined for a set $K \subseteq \mathbb{N}^k$ as a predicate p_K : $\mathbb{N}^k_\omega \to \{\texttt{true}, \texttt{false}\}$ by $p_K(\bar{d}) = (\{\bar{d}' \in \mathbb{N}^k \mid \bar{d}' \leq \bar{d}\} \cap K \neq \emptyset)$. Thus $p_K(\bar{d})$ decides whether or not the set K has any elements in common with the set of vectors smaller than or equal to $\bar{d}$. If $p_K(\bar{d})$ is decidable for any $\bar{d} \in \mathbb{N}^k_\omega$, the algorithm can be applied to compute the minimal generating of K.

We will now argue that the algorithm proposed in [10] is useful for constructing $min(W)$. The set W is upward closed since a weak upper bound $\bar{b} \in \mathbb{N}^k$ that wins GLW ensures that any $\bar{b}' \geq \bar{b}$ will also win GLW. As $min(W)$ is exactly the minimal generating set of W, $min(W)$ can be found using the algorithm in case p_W is decidable.

Lemma 1. *The predicate $p_W(\bar{d})$ is decidable for any $\bar{d} \in \mathbb{N}^k_\omega$.*

Proof. Given a vector $\bar{d} \in \mathbb{N}^k_\omega$ the following procedure will either construct $\bar{b} \in W$ such that $\bar{b} \leq \bar{d}$ or report that no such $\bar{b}$ exists. Let $\bar{d}'$ be the vector $\bar{d}$ where all ω-entries are substituted by ∞.

Starting from the configuration $(s_0, \bar{0})$ we construct a self-covering tree containing prefixes of all weighted runs. Any configuration $(s, \bar{v})$ induces the child $(s', \bar{v}')$ for any $s \xrightarrow{\bar{w}} s'$ such that $\bar{v}'[\ell] = \min\{\bar{d}'[\ell], \bar{w}[\ell] + \bar{v}[\ell]\}$ for all $\ell \in \{1, \dots, k\}$. The unfolding of the game graph stops for each branch (i.e. weighted run $(s_0, \bar{0}), (s_1, \bar{v}_1), \dots \in \mathsf{WR}_{\bar{d}'}(G)$) when reaching an i such that either

A. $\bar{v}_i[\ell] < 0$ for some $\ell \in \{1, \dots, k\}$ or
B. $s_i = s_j$ and $\bar{v}_i \geq \bar{v}_j$ for some $j < i$.

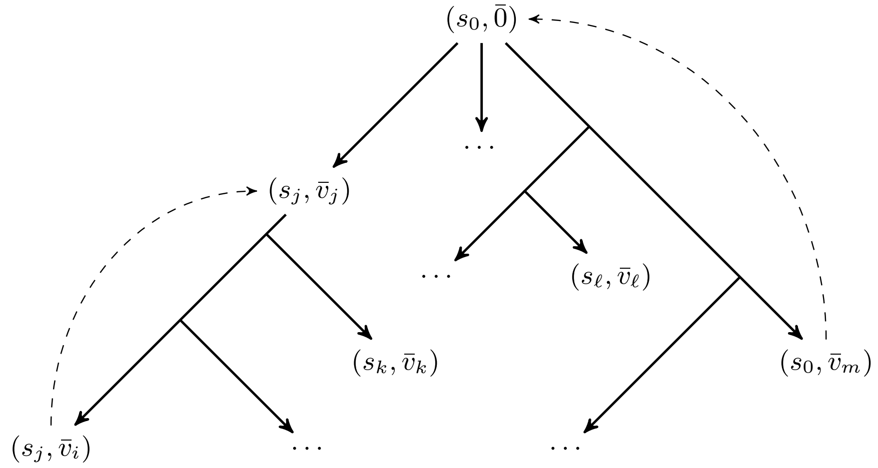


Fig. 2. Self-covering tree

Figure 2 illustrates such a self-covering tree. Here $\bar{v}_i \geq \bar{v}_j$ and $\bar{v}_m \geq \bar{0}$. Any leaf satisfies either A or B.

Notice that since the state set is finite and $(\mathbb{N}^k, \leq)$ is a wqo, such an i exists for all branches and we thus construct a finite tree. In the case of A we mark a leaf configuration $(s_i, \bar{v}_i)$ as losing and in case of B we mark $(s_i, \bar{v}_i)$ as winning. We propagate the marking of the leaves to the configuration $(s_0, \bar{0})$ in the following way, starting with configurations having only leaves as children. If the state of the configuration belongs to Player 1 and at least one child is winning, we mark the configuration as winning. Otherwise it is losing. If the state of the configuration belongs to Player 2 and all children are winning we mark the configuration as winning. Otherwise it is losing. In case $(s_0, \bar{0})$ is losing, $p_W(\bar{d}) = \texttt{false}$, as any weighted run is forced to a losing leaf if Player 2 consistently picks losing children. If $(s_0, \bar{0})$ is winning, we set $p_W(\bar{d}) = \texttt{true}$, as we can construct $\bar{b}$ and a winning strategy σ for Player 1, proving the existence of a winning vector $\bar{b}$ for GLW.

The strategy σ is determined by the tree. For each prefix of each branch $\pi_{\downarrow n} = (s_0, \bar{0}), \ldots, (s_n, \bar{v}_n)$, where $s_n \in S_1$, we let $\sigma(\pi_{\downarrow n}) = (s, \bar{v})$, where $s_n \xrightarrow{\bar{w}} s$ for some $\bar{w}$ such that $\bar{v} = \bar{v}_n + \bar{w}$ and $(s, \bar{v})$ is a winning child of $(s_n, \bar{v}_n)$. For any branch $\pi_{\downarrow m} = (s_0, \bar{0}), \ldots, (s_n, \bar{v}_n), \ldots, (s_m, \bar{v}_m)$, where $s_m = s_n$ and $\bar{v}_m \geq \bar{v}_n$ (a winning leaf) we let $\sigma(\pi_{\downarrow m}) = \sigma(\pi_{\downarrow n})$. Notice that if $(s_n, \bar{v}_n)$ does not have any winning children (or is a losing leaf) a winning strategy will never lead us to this state (and thus any next state can be picked).

It is easy to see that any weighted run respecting σ keeps all accumulated weights nonnegative, since all transitions taken by Player 1 and 2 leads to states marked as winning by the definition of σ. At some point the weighted run will enter a loop that has a nonnegative accumulated weight in all coordinates. Furthermore σ is finitely representable.

The weak upper bound $\bar{b}$ that satisfies $\bar{b} \leq \bar{d}$ and is contained in W can be found by examining the self-covering tree and pruning the tree by removing the branches not respecting σ. For the entries of $\bar{d}$ that are not ω we reuse these entries in $\bar{b}$ and for any remaining ω-entry in dimension ℓ we find the largest accumulated weight $\max_\ell$ seen in any configuration of the tree in dimension ℓ. Formally

$$\bar{b}[\ell] = \begin{cases} \max_\ell & \text{if } \bar{d}[\ell] = \omega \\ \bar{d}[\ell] & \text{otherwise} \end{cases},$$

for all $\ell \in \{1, \ldots, k\}$. This bound is safe to apply as a weak upper bound, since truncating all weights at this upper bound will not cause the accumulated weights to be negative at any point. □

As Lemma 1 allows us to use the algorithm presented in [10] we get the following corollary.

Corollary 1. *The set $min(W)$ is computable.*

Notice that we can apply the procedure seen in the proof of Lemma 1 for the special case of $\bar{d} = (\omega, \ldots, \omega)$ if we are interested in whether there exists some weak upper bound that wins GLW (e.g. determine whether W is empty or not).

4 Strict Upper Bound

For the case of a strict upper bound we see that U is also an upward closed set, but for deciding $p_U(\bar{d})$ for any $\bar{d} \in \mathbb{N}_\omega^k$ we cannot use the approach presented in the proof of Lemma 1. This is due to the construction of the self-covering tree, where we here cannot stop when reaching a cycle with positive accumulated weight in one of the coordinates, since looping forever (as indicated in Fig. 2 by dashed arrows) will eventually cause one of the strict upper bounds to be violated. For constructing $min(U)$ we instead make use of energy games with no upper bound.

Theorem 1. *The set $min(U)$ is computable in $2k$-exponential time.*

Proof. The paper [9] provides the following useful reduction. Determining whether a given upper bound $\bar{b}$ wins GLU with k weights is polynomial time reducible to determining whether the initial vector $(\bar{0}[1], \ldots, \bar{0}[k], \bar{b}[1], \ldots, \bar{b}[k])$ wins GL with $2k$ weights.

Given a k-weighted game G_k the reduction works by constructing a $2k$-weighted game G_{2k} by doubling the number of weights on each transition of G_k, adding a new start state and letting each new transition have the weight $(\bar{w}[1], \ldots, \bar{w}[k], -\bar{w}[1], \ldots, -\bar{w}[k])$ for any old transition with weight $\bar{w}$. The reduction is seen in Fig. 3, where the circular states denote either a Player 1 or Player 2 state. Now if one of the first k weights goes above $\bar{b}$, one of the last k weights will go below 0.

The paper [3] provides an algorithm running in $(k-1)$-exponential time for constructing the set $min(I)$ for any k-weighted game with only unary updates, that is a game $G = \{S_1, S_2, s_0, \longrightarrow\}$, where each $s \xrightarrow{\bar{w}} s'$ satisfies $\bar{w} \in \{-1, 0, -1\}^k$.

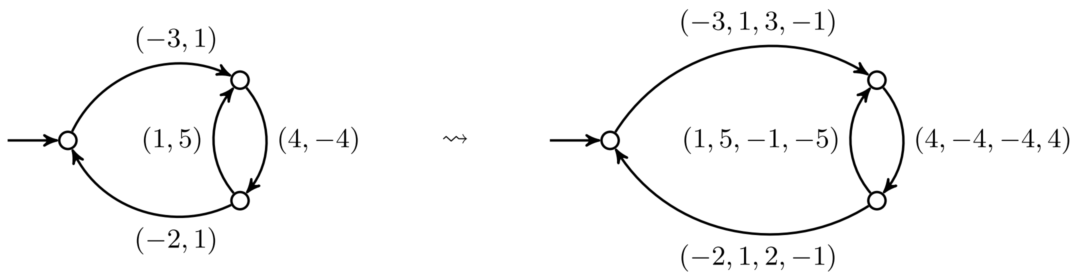


Fig. 3. Example of a reduction from G_k to G_{2k}

We can reduce G_{2k} with arbitrary updates to the unary setting by introducing intermediate transitions that repeatedly add or subtract 1 (causing an exponential blowup in the size of the automaton) and obtain the finite set $min(I)$ by applying the algorithm presented in [3].

Now we can easily construct $min(U)$ for G_k from $min(I)$ for G_{2k} as "subvectors" of the vectors in $min(I)$ with all 0's in the first k coordinates,

$$min(U) = \{\bar{b} \in \mathbb{N}^k \mid (\bar{0}[1], \ldots, \bar{0}[k], \bar{b}[1], \ldots, \bar{b}[k]) \in min(I)\} \ .$$

This procedure presented in [3] runs in $(k-1)$-exponential time for en k-weighted game with unary updates on transitions. Since we in our setting double the number of weights and reduce the arbitrary weights to unary weights, we achieve a procedure running in $2k$-exponential time. □

In case of an energy game with both unknown strict upper bound and unknown initial value, the above proof can as well be applied for finding the set of all pairs of initial values and upper bounds that will win the energy game (this set corresponds to the set $min(I)$ for G_{2k}). The problem of simultaneous synthesis with initial value and strict upper bound can therefore be solved in $2k$-exponential time.

5 Parametrised Transitions

A variant of the problem of parametrised bounds is parametrised transitions. Instead of letting the upper bound or initial value be unknown, we may also consider multiweighted energy games where not all weights of the transitions that gain resources are known. As in the case of an unknown upper bound, we are interested in not only knowing whether there exists an assignment of weights such that Player 1 has a winning strategy in the various energy games, but in constructing the actual set of assignments such that Player has a winning strategy. For no upper bound or a weak upper bound this set is upward closed and can thus be characterised by its minimal generating set. For a strict upper bound this is not the case and we must to represent the set otherwise.

Consider the automatic vacuum cleaner in Fig. 4, where the first coordinate of the weight of the charge transition is unknown (the parameter p). For a strict

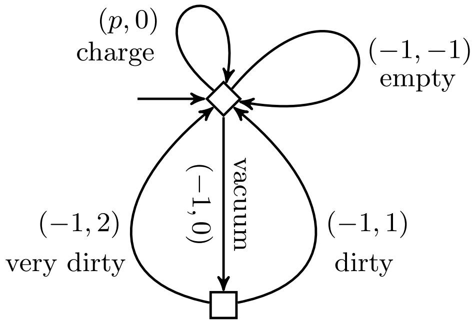


Fig. 4. A parametrised vacuum cleaner example

upper bound of $(5, 3)$ we now seek to compute the set of possible weight assignments to p such that Player 1 has a winning strategy. The smallest possible weight assigned to p is 2 (using the strategy from Fig. 1c). As it turns out, both 3, 4 and 5 as the value of p give rise to a winning strategy for Player 1 (again as seen in Fig. 1c). Surely p cannot be assigned a larger weight than the upper bound in the first coordinate, as this would enable us from charging at any time. The full set of suitable values of p is therefore $\{2, 3, 4, 5\}$.

Formally we let a parametrised k-weighted game $G = \{S_1, S_2, s_0, \longrightarrow\}$ over a set of parameters $P = \{p_1, \ldots, p_\ell\}$ be a game where $\longrightarrow \subseteq (S_1 \cup S_2) \times (\mathbb{Z} \cup P)^k \times (S_1 \cup S_2)$. Given an evaluation function $e : P \to \mathbb{N}$, we let $\longrightarrow_e$ be the set $\longrightarrow$ where any parameter $p_i \in P$ is substituted with its evaluation $e(p_i)$. In case there exists an evaluation function e for a parametrised game G with upper bound $\bar{b}$ such that $\bar{b}$ wins GLU given the game $G_e = (S_1, S_2, s_0, \longrightarrow_e)$, we say that e *wins* GLU with parametrised transitions. The same winning notion can be defined for GLW and GL with parametrised transitions. Given two evaluations e, e', we say that $e \leq e'$ if $e(p_i) \leq e'(p_i)$ for all $i \in \{1, \ldots, \ell\}$. We denote the set of winning evaluations for GLU, GLW and GL with parametrised transitions by U_T, W_T and I_T, respectively. Notice that the sets W_T and I_T are upward closed, implying that these sets can be characterised by their minimal generating set of evaluations, $min(W_T)$ and $min(I_T)$.

For $min(W_T)$ and $min(I_T)$ we as in Section 3 seek to use the algorithm presented in [10] to construct the two sets. The predicates p_{W_T} and p_{I_T} must decide for any parametrised game G whether there exists an evaluation for G in W_T or I_T, respectively.

Lemma 2. *The predicates $p_{W_T}(\bar{d})$ and $p_{I_T}(\bar{d})$ are decidable for any $\bar{d} \in \mathbb{N}^\ell_\omega$.*

Proof. Given a k-weighted game G with parametrised transitions and either a weak upper bound $\bar{b}$ or no upper bound, the existence of a winning evaluation implies the existence of a winning evaluation e that for all $i \in \{1, \ldots, \ell\}$ satisfies $e(p_i) \leq M$, where M is the largest sum obtained by adding all negative weight updates in one coordinate, i.e. $M = \max_{j \in \{1,\ldots,k\}} \left(\sum_{s \xrightarrow{\bar{w}} s'} \max(0, -\bar{w}[j]) \right)$. To see this we note that between each subsequent visit to any state we only need to visit all other states at most once (otherwise we could remove a loop or Player 2 could force an arbitrary low accumulated weight), and thus we subtract at most M in each coordinate between subsequent visits. Setting $e(p_i) = M$ for all $i \in \{1, \ldots, \ell\}$ we can apply the decidability results from [9] on the game G_e (either with or without $\bar{b}$ as weak upper bound). Thus p_{W_T} and p_{I_T} can be answered. □

Using the algorithm presented in [10], this leads to the following corollary.

Corollary 2. *The sets $min(W_T)$ and $min(I_T)$ are computable.*

In the case of GLU with parametrised transitions and $\bar{b}$ as the upper bound the set U_T is not upward closed. However, the set of useful evaluations is finite, since any winning evaluation must satisfy $-\bar{b}[i] \leq e(t)[i] \leq \bar{b}[i]$ for all transitions t and all $i \in \{1, \ldots, k\}$. This set $min(U_T)$ can therefore be constructed by an exhaustive search of the possible winning evaluations (again using the decidability results from [9]).

6 Conclusion and Future Work

Using the algorithm of Valk and Jantzen [10] we have shown how to characterise the set of winning upper bounds for multiweighted energy games with fixed initial value and a weak upper bound. For a strict upper bound the problem is solvable using $2k$-exponential time. Furthermore we have studied multiweighted energy games with parametrised transitions. For a fixed initial value and either a weak upper bound or no upper bound the same algorithm is applied to construct the set of winning evaluations. For a strict upper bound the set is shown computable as well.

Future work should include an investigation of the complexity of the above problems. As there is no upper bound on the complexity of the Valk and Jantzen algorithm, we have so far no complexity results for the results relying on the algorithm. Another future work regards parametrised transitions, where we so far are able to synthesise only nonnegative weights, and thus require that weights known to be negative are not parametrised. A likely expansion is therefore to synthesising the set of winning evaluations for energy games where any weight coordinate can be unknown. Furthermore the subject of simultaneous synthesis should be explored, where we consider games with combinations of parametrised values, this may be initial value, upper bound (strict or weak) or weight coordinates of transitions. Another direction of research is to study the problems in connection with imperfect information.

References

1. Bouyer, P., Fahrenberg, U., Larsen, K.G., Markey, N.: Timed Automata with Observers under Energy Constraints. In: 13th ACM International Conference on Hybrid Systems: Computation and Control (HSCC 2010), pp. 61–70. ACM (2010)

2. Bouyer, P., Fahrenberg, U., Larsen, K.G., Markey, N., Srba, J.: Infinite Runs in Weighted Timed Automata with Energy Constraints. In: Cassez, F., Jard, C. (eds.) FORMATS 2008. LNCS, vol. 5215, pp. 33–47. Springer, Heidelberg (2008)
3. Brázdil, T., Jančar, P., Kučera, A.: Reachability Games on Extended Vector Addition Systems with States. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010, Part II. LNCS, vol. 6199, pp. 478–489. Springer, Heidelberg (2010)
4. Chakrabarti, A., de Alfaro, L., Henzinger, T.A., Stoelinga, M.: Resource Interfaces. In: Alur, R., Lee, I. (eds.) EMSOFT 2003. LNCS, vol. 2855, pp. 117–133. Springer, Heidelberg (2003)
5. Chaloupka, J.: Z-Reachability Problem for Games on 2-Dimensional Vector Addition Systems with States Is in P. In: Kučera, A., Potapov, I. (eds.) RP 2010. LNCS, vol. 6227, pp. 104–119. Springer, Heidelberg (2010)
6. Chatterjee, K., Doyen, L.: Energy Parity Games. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010, Part II. LNCS, vol. 6199, pp. 599–610. Springer, Heidelberg (2010)
7. Chatterjee, K., Doyen, L., Henzinger, T.A., Raskin, J.F.: Generalized Mean-payoff and Energy Games. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010). LIPIcs, vol. 8, pp. 505–516. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2010)
8. Degorre, A., Doyen, L., Gentilini, R., Raskin, J.-F., Toruńczyk, S.: Energy and Mean-Payoff Games with Imperfect Information. In: Dawar, A., Veith, H. (eds.) CSL 2010. LNCS, vol. 6247, pp. 260–274. Springer, Heidelberg (2010)
9. Fahrenberg, U., Juhl, L., Larsen, K.G., Srba, J.: Energy Games in Multiweighted Automata. In: Cerone, A., Pihlajasaari, P. (eds.) ICTAC 2011. LNCS, vol. 6916, pp. 95–115. Springer, Heidelberg (2011)
10. Valk, R., Jantzen, M.: The residue of vector sets with applications to decidability problems in Petri nets. In: Rozenberg, G. (ed.) APN 1984. LNCS, vol. 188, pp. 234–258. Springer, Heidelberg (1985)