

Bounding Average-energy Games

Patricia Bouyer^{1*}, Piotr Hofman^{1,2}, Nicolas Markey^{1,3*}, Mickael Randour^{4**}, and Martin Zimmermann^{5***}

¹ LSV, CNRS & ENS Cachan, Université Paris Saclay, France

² University of Warsaw, ul. Banacha 2, 02-097 Warszawa, Poland

³ IRISA, CNRS & INRIA & U. Rennes 1, France

⁴ Computer Science Department, ULB - Université libre de Bruxelles, Belgium

⁵ Reactive Systems Group, Saarland University, 66123 Saarbrücken, Germany

Abstract. We consider average-energy games, where the goal is to minimize the long-run average of the accumulated energy. While several results have been obtained on these games recently, decidability of average-energy games with a lower-bound constraint on the energy level (but no upper bound) remained open; in particular, so far there was no known upper bound on the memory that is required for winning strategies.

By reducing average-energy games with lower-bounded energy to infinite-state mean-payoff games and analyzing the density of low-energy configurations, we show an almost tight doubly-exponential upper bound on the necessary memory, and that the winner of average-energy games with lower-bounded energy can be determined in doubly-exponential time. We also prove EXPSPACE-hardness of this problem.

Finally, we consider multi-dimensional extensions of all types of average-energy games: without bounds, with only a lower bound, and with both a lower and an upper bound on the energy. We show that the fully-bounded version is the only case to remain decidable in multiple dimensions.

1 Introduction

Quantitative two-player games of infinite duration provide a natural framework for synthesizing controllers for reactive systems with resource restrictions in an antagonistic environment (see e.g., [1,24]). In such games, player P_0 (who represents the system to be synthesized) and player P_1 (representing the antagonistic environment) construct an infinite path by moving a pebble through a graph, which describes the interaction between the system and its environment. The objective, a subset of the infinite paths that encodes the controller’s specification, determines the winner of such a play. Quantitative games extend this classical model by weights on edges for modeling costs, consumption, or rewards, and by a quantitative objective to encode the specification in terms of the weights.

As an example, consider the game in Fig. 1: we interpret negative weights as energy consumption and correspondingly positive weights as recharges. Then, P_0 (who moves the pebble at the circular states) can always maintain an energy level (the sum of the weights seen along a play prefix starting with energy zero) between zero and six using the following strategy: when at state s_0 with energy level at least two, go to state s_1 , otherwise go to state s_2 in order to satisfy the lower bound. At state s_1 , always move to s_0 . It is straightforward to verify that the strategy has the desired property when starting at the initial state s_0 with initial energy zero. Note that this strategy requires memory to be implemented, as its choices depend on the current energy level and not only on the state the pebble is currently at.

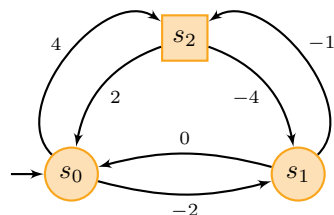


Fig. 1: Simple weighted game.

* Supported by ERC project EQualIS (308087).

** F.R.S.-FNRS postdoctoral researcher.

*** Supported by the DFG project TriCS (ZI 1516/1-1).

Formally, the *energy objective* requires P_0 to maintain an energy level within some (lower and/or upper) bounds, which are either given as input or existentially quantified. In the example above, P_0 has a strategy to win for the energy objective with lower bound zero and upper bound six. Energy objectives [3,11,18,19] and their combinations with parity objectives [9,11] have received significant attention in the literature.

However, a plain energy (parity) objective is sometimes not sufficient to adequately model real-life systems. For example, consider the following specification for the controller of an oil pump, based on a case study [7]: it has to keep the amount of oil in an accumulator within given bounds (an energy objective with given lower and upper bounds) while keeping the average amount of oil in the accumulator below a given threshold *in the long run*. The latter requirement reduces the wear and tear of the system, but cannot be expressed as an energy objective nor as a parity objective. Constraints on the long-run average energy level (which exactly represents the amount of oil in our example) can be specified using the *average-energy objective* [5]. As seen in this example, they are typically studied in conjunction with bounds on the energy level.

Recall the example in Fig. 1. The aforementioned strategy for P_0 guarantees a long-run average energy level, i.e., *average-energy*, of at most $11/4$ (recall we want to minimize it): the outcome with worst average-energy is $(s_0s_2(s_0s_1)^3)^\omega$, with energy levels $(4, 6, 4, 4, 2, 2, 0, 0)^\omega$.

The average-energy objective was first introduced by Thuijsman and Vrieze under the name *total-reward* [25] (there is an unrelated, more standard, objective called total-reward, see [5] for a discussion). Recently, the average-energy objective was independently revisited by Boros *et al.* [2] and by Bouyer *et al.* [5]. The former work studies Markov decision processes and stochastic games with average-energy objectives. The latter studies non-stochastic games with average-energy objectives, with or without lower and upper bounds on the energy level; it determines the complexity of computing the winner and the memory requirements for winning strategies in such games. In particular, it solves games with average-energy objectives with *both* an upper and a lower bound on the energy level by a reduction to mean-payoff games: to this end, the graph is extended to track the energy level between these bounds (a losing sink for P_0 is reached if these bounds are exceeded). Thus, the bounds on the energy level are already taken care of and the average-energy objective can now be expressed as a *mean-payoff objective* [13], as the new graph encodes the current energy level in its weights. This reduction yields an exponential-time decision algorithm. Moreover, it is shown in [5] that these games are indeed EXPTIME-complete. Note that the algorithm crucially depends on the upper bound being given as part of the input, which implies that the graph of the extended game is still finite.

One problem left open in [5] concerns average-energy games *with only a lower bound* on the energy level: computing the winner is shown to be EXPTIME-hard, but it is problem is decidable at all. Similarly, pseudo-polynomial lower bounds (i.e., lower bounds that are polynomial in the *values* of the weights, but possibly exponential in the size of their binary representations) on the necessary memory to implement a winning strategy for P_0 are given, but no upper bound is known. The major obstacle toward solving these problems is that without an upper bound on the energy, a strategy might allow arbitrarily large energy levels while still maintaining a bounded average, by enforcing long stretches with a small energy level to offset the large levels.

A step toward resolving these problems was taken by considering two variants of energy and average-energy objectives where (i) the upper bound on the energy level, or (ii) the threshold on the average-energy, is existentially quantified [21]. It turns out that these two variants are equivalent. One direction is trivial: if the energy is bounded, then the average-energy is bounded. On the other hand, if P_0 can guarantee some upper bound on the average, then he can also guarantee an upper bound on the energy level, i.e., an (existential) average-energy objective can always be satisfied with bounded energy levels. This is shown by transforming a strategy satisfying a bound on the average (but possibly allowing arbitrarily high energy levels) into one that bounds the energy by skipping parts of plays where the energy level is much higher than the threshold on the average. However, the proof is not effective: it does not yield an upper bound on the necessary energy level, just a guarantee that some bound exists. Even more so, it is still possible that the average has to increase when keeping the energy bounded. Hence, it does not answer our open problem: does achieving a *given* threshold on the average-energy require unbounded energy levels and infinite memory?

Another potential approach toward solving the problem is to extend the reduction presented in [5] (which goes from average-energy games with both lower and upper bound on the energy level to mean-payoff games) to games without such an upper bound, which results in an infinite graph. This graph can be seen as the configuration graph of a one-counter pushdown system, i.e., the stack height corresponds to the current energy level, and the average-energy objective is again transformed into a mean-payoff objective, where the weight of an edge is given by the stack height at the target of the edge. Hence, the weight function is unbounded. To the best of our knowledge, such mean-payoff games have not been studied before. However, mean-payoff games on pushdown systems with bounded weight functions are known to be undecidable [12].

Our Contribution. This paper gives a full presentation of the results published in its conference version [4]. We develop the first algorithm for solving games with average-energy objectives and a lower bound on the energy level, and give an upper bound on the necessary memory to implement a winning strategy for P_0 in such games.

First, we present an algorithm solving such games in doubly-exponential time (for the case of a binary encoding of the weights). The algorithm is based on the characterization of an average-energy game as a mean-payoff game on an infinite graph described above. If the average-energy of a play is bounded by the threshold t , then configurations with energy level at most t have to be visited frequently. As there are only finitely many such configurations, we obtain cycles on this play. By a more fine-grained analysis, we obtain such a cycle with an average of at most t and whose length is bounded exponentially. Finally, by analyzing strategies for reachability objectives in pushdown games, we show that P_0 can ensure that the distance between such cycles is bounded doubly-exponentially. From these properties, we obtain a doubly-exponential upper bound on the necessary energy level to ensure an average-energy of at most t . The resulting equivalent average-energy game with a lower and an upper bound can be solved in doubly-exponential time. Furthermore, if the weights and the threshold are encoded in unary (or are bounded polynomially in the number of states), then we obtain an exponential-time algorithm.

Second, from the reduction sketched above, we also obtain a doubly-exponential upper bound on the necessary memory for P_0 , the first such bound. In contrast, a certain succinct one-counter game due to Hunter [16], which can easily be expressed as an average-energy game with threshold zero, shows that our bound is almost tight: in the resulting game of size n , energy level $2^{(2^{\sqrt{n}}/\sqrt{n})-1}$ is necessary to win. Again, in the case of unary encodings, we obtain an (almost) tight exponential bound on the memory requirements.

Third, we improve the lower bound on the complexity of solving average-energy games with only a lower bound on the energy level from EXPTIME to EXPSPACE by a reduction from succinct one-counter games [17].

Fourth, we show that multi-dimensional average-energy games are undecidable, both for the case without any bounds and for the case of only lower bounds. Only the case of games with both lower and upper bounds turns out to be decidable: it is shown to be both in NEXPTIME and in coNEXPTIME. This problem trivially inherits EXPTIME-hardness from the one-dimensional case.

2 Preliminaries

Graph games. We consider finite turn-based weighted games played on graphs between two players, denoted by P_0 and P_1 . Such a game is a tuple $G = (S_0, S_1, E)$ where (i) S_0 and S_1 are disjoint sets of *states* belonging to P_0 and P_1 , with $S = S_0 \uplus S_1$, (ii) $E \subseteq S \times [-W; W] \times S$, for some $W \in \mathbb{N}$, is a set of integer-weighted *edges*. Given an edge $e = (s, w, t) \in E$, we write $src(e)$ for the source state s of e , $tgt(e)$ for its target state t , and $w(e)$ for its weight w . We assume that for every $s \in S$, there is at least one outgoing edge $(s, w, s') \in E$.

Let $s \in S$. A *play* from s is an infinite sequence of edges $\pi = (e_i)_{1 \leq i}$ such that $src(e_1) = s$ and $tgt(e_i) = src(e_{i+1})$ for all $i \geq 1$. A play induces a corresponding sequence of states, denoted $\hat{\pi} = (s_j)_{0 \leq j}$, such that for any e_i , $i \geq 1$, in π , $s_{i-1} = src(e_i)$ and $s_i = tgt(e_i)$. We write $first(\pi) = s_0$ for its initial state (here, s). A *play prefix* from s is a finite sequence of edges $\rho = (e_i)_{1 \leq i \leq k}$ following the same rules and notations. We additionally write $last(\rho) = s_k = tgt(e_k)$ for its last state. We let ϵ_s (or ϵ when s is clear from the context) denote the empty play prefix from s , with $last(\epsilon_s) = first(\epsilon_s) = s$. A non-empty prefix ρ such that $last(\rho) = first(\rho)$ is called a *cycle*. The length of a prefix $\rho = (e_i)_{1 \leq i \leq k}$ is its number of edges,

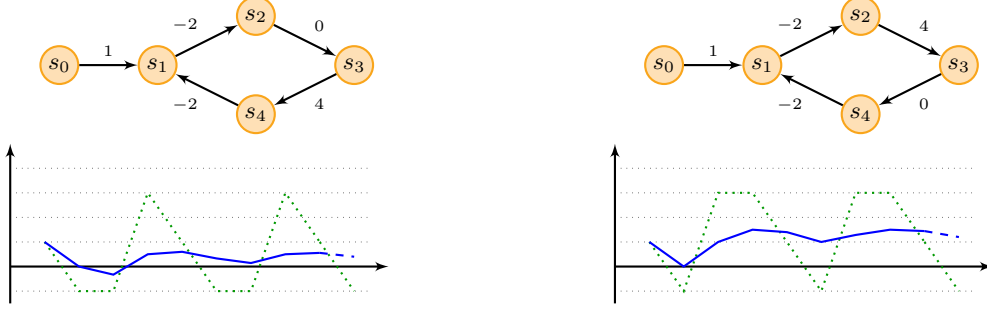


Fig. 2: Two plays with identical mean-payoffs and total-payoffs. The left one has average-energy $1/2$, in contrast to $3/2$ for the right one. Green (dotted) and blue curves respectively represent the energy level and the average-energy over prefixes.

i.e., $\ell(\rho) = k$. For a play π , $\ell(\pi) = \infty$. Given a prefix ρ and a play (or prefix) π with $\text{last}(\rho) = \text{first}(\pi)$, the concatenation between ρ and π is denoted by $\rho \cdot \pi$.

For a play $\pi = (e_i)_{1 \leq i}$ and $1 \leq j \leq k$, we write $\pi_{[j,k]}$ to denote the finite sequence $(e_i)_{j \leq i \leq k}$, which is a prefix from $\text{src}(e_j)$; we write $\pi_{\leq k}$ for $\pi_{[1,k]}$. For any $i \geq 1$ and $j \geq 0$, we write π_i for edge e_i and $\hat{\pi}_j$ for state s_j . Similar notations are used for prefixes ρ , with all indices bounded by $\ell(\rho)$.

The set of all plays in G from a state s is denoted by $\text{Plays}(G, s)$, and the set of all such prefixes is denoted by $\text{Pref}_s(G, s)$. We write $\text{Plays}(G)$ and $\text{Pref}_s(G)$ for the unions of those sets over all states. We say that a prefix $\rho \in \text{Pref}_s(G)$ belongs to P_i , for $i \in \{0, 1\}$, if $\text{last}(\rho) \in S_i$. The set of prefixes that belong to P_i is denoted by $\text{Pref}_i(G)$, and we define $\text{Pref}_i(G, s) = \text{Pref}_i(G) \cap \text{Pref}_s(G, s)$.

Payoffs. Given a non-empty prefix $\rho = (e_i)_{1 \leq i \leq n}$, we define the following payoffs:

- its *energy level* as $\text{EL}(\rho) = \sum_{i=1}^n w(e_i)$;
- its *mean-payoff* as $\text{MP}(\rho) = \frac{1}{n} \sum_{i=1}^n w(e_i) = \frac{1}{n} \text{EL}(\rho)$;
- its *average-energy* as $\text{AE}(\rho) = \frac{1}{n} \sum_{i=1}^n \text{EL}(\rho_{\leq i})$.

These definitions are extended to plays by taking the upper limit of the respective functions applied to the sequence of prefixes of the plays, e.g.,

$$\overline{\text{AE}}(\pi) = \limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \text{EL}(\pi_{\leq i}).$$

Example 1. We illustrate those definitions on a small example depicted in Fig. 2: it displays two small (1-player, deterministic) weighted games, together with the evolution of the energy level and average-energy along their unique play. As noted in [5], the *average-energy* can help in discriminating plays that have identical *total-payoffs* (i.e., the limits of high and low points in the sequence of energy levels), in the same way that total-payoff can discriminate between plays having the same *mean-payoff*. Indeed, in our example, both plays have mean-payoff equal to zero and supremum (resp. infimum) total-payoff equal to three (resp. -1), but they end up having different averages: the average-energy is $1/2$ for the left play, while it is $3/2$ for the right one.

Strategies. A *strategy* for P_i , with $i \in \{0, 1\}$, from a state s is a function $\sigma_i: \text{Pref}_i(G, s) \rightarrow E$ satisfying $\text{src}(\sigma_i(\rho)) = \text{last}(\rho)$ for all $\rho \in \text{Pref}_i(G, s)$. We denote by $\text{Strats}_i(G, s)$, the set of strategies for P_i from state s . We drop G and s when they are clear from the context.

A play $\pi = (e_j)_{1 \leq j}$ from s is called an *outcome* of strategy σ_i of P_i if, for all $k \geq 0$ where $\pi_{\leq k} \in \text{Pref}_i(G, s)$, we have $\sigma_i(\pi_{\leq k}) = e_{k+1}$. Given a state $s \in S$ and strategies σ_0 and σ_1 from s for both players, we denote by $\text{Out}(s, \sigma_0, \sigma_1)$ the unique play that starts in s and is an outcome of both σ_0 and σ_1 . When fixing the strategy of only P_i , we denote the set of outcomes by

$$\text{Out}(s, \sigma_i) = \{\text{Out}(s, \sigma_0, \sigma_1) \mid \sigma_{1-i} \in \text{Strats}_{1-i}(G, s)\}.$$

Game objective	1-player	2-player	memory
MP	PTIME [20]	$NP \cap \text{coNP}$ [28]	memoryless [13]
EGL	PTIME [3]	$NP \cap \text{coNP}$ [3,8]	memoryless [8]
$EGLU$	PSPACE-c. [14]	EXPTIME-c. [3]	exponential [5]
AE	PTIME [5]	$NP \cap \text{coNP}$ [5]	memoryless [5]
AEU	PSPACE-c. [5]	EXPTIME-c. [5]	exponential [5]
AEL	PSPACE-e. / NP-h. [5]	2-EXPTIME-e. / EXPSPACE-h.	super-exp. / doubly-exp.

Table 1: Complexity of deciding the winner and memory requirements for quantitative games: MP stands for mean-payoff, EGL (resp. $EGLU$) for lower-bounded (resp. lower- and upper-bounded) energy, AE for average-energy, AEL (resp. AEU) for average-energy under a lower bound (resp. and upper bound $U \in \mathbb{N}$) on the energy, c. for complete, e. for easy, and h. for hard. All memory bounds are tight (except for AEL).

Objectives. An objective in G is a set $\mathcal{W} \subseteq \text{Plays}(G)$. Given a game G , an initial state s_{init} , and an objective \mathcal{W} , a strategy $\sigma_0 \in \text{Strats}_0$ is winning for P_0 if $\text{Out}(s_{\text{init}}, \sigma_0) \subseteq \mathcal{W}$.

We consider the following objectives for P_0 :

- The **lower-bounded energy** objective $\text{Energy}_L = \{\pi \in \text{Plays}(G) \mid \forall n \geq 1, \text{EL}(\pi_{\leq n}) \geq 0\}$ requires a non-negative energy level at all times.⁶
- Given an upper bound $U \in \mathbb{N}$, the **lower- and upper-bounded energy** objective $\text{Energy}_{LU}(U) = \{\pi \in \text{Plays}(G) \mid \forall n \geq 1, \text{EL}(\pi_{\leq n}) \in [0, U]\}$ requires that the energy always remains non-negative and below the upper bound U along a play.
- Given a threshold $t \in \mathbb{Q}$, the **mean-payoff** objective $\text{MeanPayoff}(t) = \{\pi \in \text{Plays}(G) \mid \overline{\text{MP}}(\pi) \leq t\}$ requires that the mean-payoff is at most t .
- Given a threshold $t \in \mathbb{Q}$, the **average-energy** objective $\text{AvgEnergy}(t) = \{\pi \in \text{Plays}(G) \mid \overline{\text{AE}}(\pi) \leq t\}$ requires that the average-energy is at most t .

For the MeanPayoff and AvgEnergy objectives, P_0 aims to *minimize* the payoff.

Decision problem. In this paper, we focus on weighted games with a combination of energy and average-energy objectives, by a detour via mean-payoff objectives. The exact problem we tackle is named the *AEL threshold problem* and is defined as follows: given a finite game G , an initial state $s_{\text{init}} \in S$, and a threshold $t \in \mathbb{Q}$ given as a fraction $\frac{t_1}{t_2}$ with t_1 and t_2 two natural numbers given in binary, decide whether P_0 has a winning strategy from s_{init} for the objective $\text{AvgEnergy}_L(t) = \text{Energy}_L \cap \text{AvgEnergy}(t)$. As for the threshold, we consider a binary encoding of the weights in G : we thus study the complexity of the problem with regard to the length of the input’s binary encoding (i.e., the number of bits used to represent the graph and the numbers involved).

Variants of this problem involving the above-mentioned payoff functions, and combinations thereof, had been previously investigated, see Table 1 for a summary of the results. In this paper, we focus on the remaining case, namely 2-player games with AEL objectives, for which decidability was not known, and proving the computational- and memory complexities given in the corresponding cells of the table.

3 Equivalence with an Infinite-State Mean-payoff Game

Let $G = (S_0, S_1, E)$ be a finite weighted game, $s_{\text{init}} \in S$ be an initial state, and $t \in \mathbb{Q}$ be a threshold. We define its *expanded infinite-state weighted game* as $G' = (I_0, I_1, \Delta)$ defined by

⁶ For the sake of readability, we assume the initial credit to be zero for energy objectives throughout this paper. Still, our techniques can easily be generalized to an arbitrary initial credit $c_{\text{init}} \in \mathbb{N}$.

- $\Gamma_0 = S_0 \times \mathbb{N}$, and $\Gamma_1 = S_1 \times \mathbb{N} \uplus \{\perp\}$ (where \perp is a fresh sink state that does not belong to G); then $\Gamma = \Gamma_0 \uplus \Gamma_1$ is the global set of states;
- Δ is composed of the following edges:
 - a transition $((s, c), c', (s', c')) \in \Delta$ whenever there is $(s, w, s') \in E$ with $c' = c + w \geq 0$;
 - a transition $((s, c), \lceil t \rceil + 1, \perp) \in \Delta$ whenever there is $(s, w, s') \in E$ such that $c + w < 0$;
 - finally, a transition $(\perp, \lceil t \rceil + 1, \perp) \in \Delta$.

In this expanded game, elements of Γ are called *configurations*, and the initial configuration is set to $(s_{\text{init}}, 0)$.

Lemma 1. *Player P_0 has a winning strategy in G from state s_{init} for the objective $\text{AvgEnergy}_L(t)$ if, and only if, he has a winning strategy in G' from configuration $(s_{\text{init}}, 0)$ for the objective $\text{MeanPayoff}(t)$.*

Proof. Let ι be the mapping that assigns to every play $\pi = (e_i)_{1 \leq i}$ from s_{init} an expanded play in G' from $(s_{\text{init}}, 0)$, defined inductively as follows.

- $\iota(\epsilon) = \epsilon$;
- Let $\gamma_n = \text{last}(\iota(\pi_{\leq n}))$ for $n \geq 0$ with $\gamma_0 = \text{last}(\iota(\epsilon)) = (s_{\text{init}}, 0)$, the initial configuration. For $n \geq 1$, $\pi_{\leq n} = \pi_{\leq n-1} \cdot (s_{n-1}, w_n, s_n)$, we define $\iota(\pi_{\leq n}) = \iota(\pi_{\leq n-1}) \cdot \delta_n$ where $\delta_n \in \Delta$ is defined as follows:
 - if $\gamma_{n-1} = \perp$ then $\delta_n = (\perp, \lceil t \rceil + 1, \perp)$;
 - else $\gamma_{n-1} = (s_{n-1}, c_{n-1})$ and if $c_{n-1} + w_n < 0$ then $\delta_n = (\gamma_{n-1}, \lceil t \rceil + 1, \perp)$;
 - else $\delta_n = (\gamma_{n-1}, c_n, \gamma_n)$ for $c_n = c_{n-1} + w_n$ and $\gamma_n = (s_n, c_n)$.

One easily checks that the following properties hold:

- for any $n \geq 0$, $\ell(\iota(\pi_{\leq n})) = \ell(\pi_{\leq n})$,
- $\iota(\pi)$ reaches \perp if, and only if, $\text{EL}(\pi_{\leq j}) < 0$ for some $j \geq 1$,
- if $\iota(\pi)$ never reaches \perp , then for all $j \geq 1$,

$$\text{MP}(\iota(\pi_{\leq j})) = \text{AE}(\pi_{\leq j}).$$

Hence, if a play π in G always keeps its energy level non-negative, we have that $\overline{\text{MP}}(\iota(\pi)) = \overline{\text{AE}}(\pi)$, and if it does not, that $\overline{\text{MP}}(\iota(\pi)) = \lceil t \rceil + 1$ thanks to the sink state \perp . Furthermore, observe that if a play π' in G' does not reach \perp , then there is a unique corresponding play $\pi = \iota^{-1}(\pi')$ in G : i.e., we have a bijection for plays that avoid \perp .

Now, pick a winning strategy σ_0 for P_0 in G from s_{init} for $\text{AvgEnergy}_L(t)$. First notice that for every $\pi \in \text{Out}(s_{\text{init}}, \sigma_0)$, for every $j \geq 1$, $\text{EL}(\pi_{\leq j}) \geq 0$; in particular, $\iota(\pi)$ never visits \perp in G' . We define the strategy σ'_0 in G' that mimics σ_0 based on the aforementioned bijection. By construction, we have that any outcome π' of σ'_0 avoids \perp and thus that its mean-payoff is equal to the average-energy of $\iota^{-1}(\pi')$ in G , which is an outcome of σ_0 . Hence, by hypothesis on σ_0 , σ'_0 is winning for $\text{MeanPayoff}(t)$.

Conversely, pick a winning strategy σ'_0 in G' from $(s_{\text{init}}, 0)$ for objective $\text{MeanPayoff}(t)$. No outcome may visit \perp , otherwise the mean-payoff would be $\lceil t \rceil + 1 > t$. Hence the bijection holds for all outcomes of σ'_0 and we use it to define the strategy σ_0 that mimics σ'_0 in G . By hypothesis on σ'_0 , any outcome π of σ_0 will keep the energy level non-negative at all times, and will be such that $\overline{\text{AE}}(\pi) = \overline{\text{MP}}(\iota(\pi))$, hence will have its average-energy bounded by t since $\iota(\pi)$ is an outcome of σ'_0 . Thus σ_0 is winning for $\text{AvgEnergy}_L(t)$, which concludes our proof. \square

For the rest of this paper, we fix a weighted game $G = (S_0, S_1, E)$ and a threshold $t \in \mathbb{Q}$, and work on the corresponding expanded weighted game $G' = (\Gamma_0, \Gamma_1, \Delta)$. We write $t = \frac{t_1}{t_2} = \lfloor t \rfloor + \frac{t'}{t_2}$, where $t_1, t_2, t' \in \mathbb{N}$ (recall they are given in binary), and $0 \leq t' < t_2$, and $\lfloor t \rfloor$ stands for the integral part of t . We also let $\tilde{t} = \lfloor t \rfloor + 1 - t = 1 - \frac{t'}{t_2}$. Hence $\tilde{t} = 1$ indicates that t is an integer. For a given threshold $t \in \mathbb{Q}$, we consider $\Gamma^{\leq t} = \{(s, c) \in \Gamma \mid c \leq t\}$, i.e., the set of configurations below the threshold.

Note that G' can be interpreted as a one-counter pushdown mean-payoff game with an unbounded weight function. While it is well-known how to solve mean-payoff games over *finite* arenas, not much is known for

infinite arenas (see Section 1). However, our game has a special structure that we will exploit to obtain an algorithm. Roughly, our approach consists in transforming the $AvgEnergy_L(t)$ objective into an equivalent $AvgEnergy_{LU}(t, U) = Energy_{LU}(U) \cap AvgEnergy(t)$ objective, where (the value of) U is doubly-exponential in the input by analyzing plays and strategies in G' . In other terms, we show that any winning strategy for $AvgEnergy_L(t)$ can be transformed into another winning strategy along which the energy level remains upper-bounded by U .

The proof is two-fold: we first show (in Section 4) that we can bound the energy level for the special case where the objective consists in reaching a finite set of configurations of the game (with only a lower bound on the energy level). This is achieved by a detour to pushdown games: while there are known algorithms for solving reachability pushdown games, to the best of our knowledge, there are no (explicit) results bounding the maximal stack height.

As a second step (in Section 5), we identify *good cycles* in winning outcomes, and prove that they can be shown to have bounded length. The initial configurations of those cycles will then be the targets of the reachability games above. Combining these two results yields the desired upper bound on the energy levels.

4 Bounding One-counter Reachability Games

We focus here on a reachability objective in G' , where the target set is a subset $\Gamma' \subseteq \Gamma^{\leq t}$: we aim at bounding the maximal energy level that needs to be visited with a winning strategy.

The game G' is a particular case of a pushdown game [27]. Hence we use results on pushdown games, and build a new winning strategy, in which we will be able to bound the energy level at every visited configuration. Note that the bound M' in the following lemma is doubly-exponential, as we encode W , the largest absolute weight in G , and the threshold t , in binary. The proof of the next lemma is based on the reformulation of the algorithm from [27] made in [15].

Lemma 2. *Fix $M \in \mathbb{N}$. There exists $M' = 2^{\mathcal{O}(M+|S|+|E| \cdot W+|S| \cdot (\lceil t \rceil + 1))}$ such that for every $\gamma = (s, c)$ with $c \leq M$ and for every $\Gamma' \subseteq \Gamma^{\leq t}$, if there is a strategy for P_0 to reach Γ' from γ in G' , then there is also a strategy which ensures reaching Γ' from γ without exceeding energy level M' .*

Proof. We rely on Walukiewicz's [27] solution of parity games on configuration graphs of pushdown machines via a simulation by parity games on finite graphs as presented in [15]. First, we construct such a pushdown parity game \mathcal{P} that is won by P_0 if, and only if, there is a strategy for P_0 to reach Γ' from γ in G' . Then, we apply Walukiewicz's reduction, which yields a winning strategy $\sigma_{\mathcal{P}}$ for \mathcal{P} that is induced by a memoryless winning strategy σ_G for the parity game \mathcal{G} constructed in the reduction. Finally, we show that this strategy can be turned into a strategy for G' that ensures reaching Γ' without exceeding energy level M' .

When interpreting G' as a pushdown game, the energy level is stored in unary on the stack, and the stack alphabet has only two symbols, one for the counter, and one (denoted by \perp) for the bottom of the stack. Our weights here are encoded in binary, which implies that an exponential number of stack symbols may be pushed onto or may be popped of the stack during a single transition. On the other hand, Walukiewicz's construction considers pushdown machines where at most one symbol may be pushed or popped during a single transition. Obviously, pushing and popping multiple symbols can be simulated in the latter model by adding additional states, at the price of an exponential blow-up.

We begin with the definition of \mathcal{P} , which runs in three phases. In the first phase, the configuration γ is produced. This phase is deterministic, i.e., the players have no choices in this phase. Then, in the second phase, a play of G' starting in γ is simulated. At any point of this simulation, P_0 can stop the simulation and start a third phase. Say, he stops the simulation of G' in configuration γ' . The third phase is designed to check whether γ' is in Γ' . If this is the case, then the configuration (q_f, \perp) is reached, where q_f is a fresh state. If not, then a configuration (q_r, \perp) is reached, where q_r is again a fresh state. This is implemented by deterministically determining the stack height of the configuration γ' , which can be stopped as soon as it turns out that the stack height is greater than t , as this implies $\gamma' \notin \Gamma'$. If P_0 does not stop the simulation, the configuration (q_f, \perp) is never reached.

Thus, the number of states of the pushdown system underlying \mathcal{P} can be bounded by $\mathcal{O}(M + |S| + |E| \cdot W + |S| \cdot (\lceil t \rceil + 1))$, $M + 1$ states for the first phase, $\mathcal{O}(|S| + |E| \cdot W)$ for the second one, and $|S| \cdot (\lceil t \rceil + 1) + 2$ states for the third phase.

By construction, the premise of the statement we prove, i.e., there is a strategy for P_0 to reach Γ' from γ in G' , is equivalent to P_0 having a strategy for \mathcal{P} to reach the configuration (q_f, \perp) from the initial configuration, which is a reachability objective. We turn this into a parity objective as usual: All states are colored by 1 save for a fresh one with color 0 that is only reachable from the configuration (q_f, \perp) . This fresh state is equipped with a self-loop, so that the parity objective is satisfied if, and only if, the fresh state is reached via (q_f, \perp) .

Using the adapted construction of Walukiewicz as presented in [15], we obtain a parity game \mathcal{G} on a finite graph of size $2^{\mathcal{O}(|\mathcal{P}|)}$ with a designated initial state s_0 , and such that 0 is the only even color in \mathcal{G} (the larger harmless color needed in [15] for auxiliary vertices can be taken as 3). The crucial property of \mathcal{G} is that P_0 has a winning strategy in \mathcal{G} from s_0 if, and only if, P_0 has a winning strategy in \mathcal{P} from its initial configuration (that we denote from now on by (q_0, \perp)). Even more so, a memoryless winning strategy $\sigma_{\mathcal{G}}$ for P_0 in \mathcal{G} from s_0 can be turned into a winning strategy $\sigma_{\mathcal{P}}$ for P_0 in \mathcal{P} from (q_0, \perp) so that: for every play prefix $\rho \in \text{Out}((q_0, \perp), \sigma_{\mathcal{P}})$ which has not reached yet a configuration of color 0, there is a play prefix $\rho' \in \text{Out}(s_0, \sigma_{\mathcal{G}})$ which does not contain any vertex of color 0, such that the stack height of ρ corresponds to the number of vertices of color 1 occurring in ρ' .⁷

Now, assume such a play prefix ρ in \mathcal{P} reaches stack height greater than $|\mathcal{G}|$. Then, the corresponding play prefix ρ' in \mathcal{G} visits at least one vertex of color 1 twice without an occurrence of color 0 in between. From such a play prefix, since $\sigma_{\mathcal{G}}$ is memoryless, one can construct an infinite outcome of $\sigma_{\mathcal{G}}$ from s_0 , which visits vertices of color 1 infinitely often, but no vertex of color 0. Such a play is losing for P_0 , which contradicts the fact that $\sigma_{\mathcal{G}}$ is winning from s_0 . Hence, such a play prefix ρ does not exist, i.e., the stack height of plays that start in the initial configuration, that are consistent with $\sigma_{\mathcal{P}}$, and have not yet reached (q_f, \perp) is bounded by $|\mathcal{G}|$.

Finally, the strategy $\sigma_{\mathcal{P}}$ can now easily be transformed into a strategy for G' that ensures reaching Γ' from γ without exceeding energy level $M' = |\mathcal{G}| = 2^{\mathcal{O}(M + |S| + |E| \cdot W + |S| \cdot (\lceil t \rceil + 1))}$. \square

5 A Doubly-exponential Time Algorithm

Let $\rho = (e_i)_{1 \leq i \leq n}$ be a prefix in G' such that its configurations are denoted $(s_0, c_0), \dots, (s_n, c_n)$. Observe that by construction of the expanded game, $\text{MP}(\rho) = \frac{1}{n} \sum_{i=1}^n c_n$. In particular, note that the initial configuration (s_0, c_0) does not appear when computing its mean-payoff.

Let $\tilde{\Gamma} \subseteq \Gamma$ be a set of configurations of G' and ρ a play prefix. We define $\mathbf{d}(\tilde{\Gamma}, \rho)$ by:

$$\mathbf{d}(\tilde{\Gamma}, \rho) = \frac{|\{1 \leq i \leq \ell(\rho) \mid \hat{\rho}_i \in \tilde{\Gamma}\}|}{\ell(\rho)},$$

which denotes the proportion (or *density*) of configurations belonging to $\tilde{\Gamma}$ along ρ . Observe that the initial configuration $\hat{\rho}_0$ is not taken into account: this is because $\mathbf{d}(\tilde{\Gamma}, \rho)$ will be strongly linked to the mean-payoff, as we now explain.

5.1 Analyzing Winning Plays

In this section, we analyze winning plays in G' , and prove that they must contain a cycle that is “short enough” and has mean-payoff less than or equal to t . To achieve this, we first establish a lower bound on the density of configurations below the threshold along a winning play.

⁷ The result proven in [15] is actually more general. The statement here follows from the fact that the stack height of ρ is equal to the stair-score of color 1 in the setting we consider here.

Lemma 3. *Let π be a play in G' from $(s_{\text{init}}, 0)$ with $\overline{\text{MP}}(\pi) \leq t$. Then, there exists $n \in \mathbb{N}$ such that for every $n' \geq n$,*

$$\mathbf{d}(\Gamma^{\leq t}, \pi_{\leq n'}) \geq \frac{\tilde{t}}{2(t+1)}.$$

Proof. Write $\pi = (e_i)_{1 \leq i}$, and write $\hat{\pi}_j = (s_j, c_j)$ for all $j \geq 0$. That $\overline{\text{MP}}(\pi) \leq t$ means that for any $\varepsilon > 0$, there exists $n \in \mathbb{N}$ such that for every $n' \geq n$,

$$\frac{\sum_{i=1}^{n'} c_i}{n'} \leq t + \varepsilon.$$

Pick $0 < \varepsilon \leq \frac{\tilde{t}}{2}$, and $n \in \mathbb{N}$ for which the above property holds. For any $n' \geq n$, let I be the subset of positions i in $[1, n']$ such that $c_i \leq t$, and J its complement: $I \uplus J = [1, n']$. We can then compute:

$$\begin{aligned} t + \varepsilon &\geq \frac{\sum_{i=1}^{n'} c_i}{n'} = \frac{\sum_{i \in I} c_i + \sum_{i \in J} c_i}{n'} \\ &\geq 0 + \frac{\sum_{i \in J} ([t] + 1)}{n'} = \frac{|J| \cdot ([t] + 1)}{n'} \end{aligned}$$

since any $c_i \in \mathbb{N}$ that is strictly greater than t is larger than or equal to $[t] + 1$. Hence $t + \varepsilon \geq \frac{|J| \cdot ([t] + 1)}{n'}$ or equivalently

$$\frac{t + \varepsilon}{[t] + 1} \geq \frac{|J|}{n'}. \quad (1)$$

Now:

$$\begin{aligned} \mathbf{d}(\Gamma^{\leq t}, \pi_{\leq n'}) &= \frac{|I|}{n'} = 1 - \frac{|J|}{n'} \\ &\geq 1 - \frac{t + \varepsilon}{[t] + 1} && \text{(due to equation 1)} \\ &\geq \frac{[t] + 1 - t - \varepsilon}{[t] + 1} = \frac{\tilde{t} - \varepsilon}{[t] + 1} \\ &\geq \frac{\tilde{t}}{2([t] + 1)} && \text{(since } \varepsilon \leq \frac{\tilde{t}}{2}\text{)} \\ &\geq \frac{\tilde{t}}{2(t+1)}. \end{aligned}$$

This concludes the proof of the lemma. \square

We now deduce that there is at least one configuration below the threshold that appears with lower-bounded density.

Lemma 4. *Let π be a play in G' from $(s_{\text{init}}, 0)$ with $\overline{\text{MP}}(\pi) \leq t$. There exists $\gamma \in \Gamma^{\leq t}$ such that for any $n \in \mathbb{N}$, there exists infinitely many positions $n' \geq n$ for which*

$$\mathbf{d}(\{\gamma\}, \pi_{[n, n']}) \geq \frac{\tilde{t}}{4(t+1)^2 |S|}.$$

Proof. Since $|\Gamma^{\leq t}| \leq (t+1) \cdot |S|$,⁸ we easily get from Lemma 3 that there is some $n \in \mathbb{N}$ and some configuration $\gamma \in \Gamma^{\leq t}$ such that for any $n' \geq n$, $\mathbf{d}(\{\gamma\}, \pi_{\leq n'}) \geq \frac{\tilde{t}}{2(t+1)|\Gamma^{\leq t}|} = \frac{\tilde{t}}{2(t+1)^2 |S|}$.

⁸ This is thanks to the special structure of our infinite-state game.

For this n and such a configuration γ , we have that for any $n' > n$:

$$\begin{aligned} \mathbf{d}(\{\gamma\}, \pi_{\leq n'}) &= \frac{|\{1 \leq i \leq n \mid (s_i, c_i) = \gamma\}| + |\{n+1 \leq i \leq n' \mid (s_i, c_i) = \gamma\}|}{n'} \\ &\leq \frac{n}{n'} + \frac{|\{n+1 \leq i \leq n' \mid (s_i, c_i) = \gamma\}|}{n'} \\ &\leq \frac{n}{n'} + \frac{|\{n+1 \leq i \leq n' \mid (s_i, c_i) = \gamma\}|}{n' - n} \\ &= \frac{n}{n'} + \mathbf{d}(\{\gamma\}, \pi_{[n, n']}) \end{aligned}$$

Thus $\mathbf{d}(\{\gamma\}, \pi_{[n, n']}) \geq \frac{\tilde{t}}{2(t+1)^2|S|} - \frac{n}{n'}$ for infinitely many n' . For those n' larger than $\frac{4n(t+1)^2|S|}{\tilde{t}}$ (still infinitely many), it holds $\mathbf{d}(\{\gamma\}, \pi_{[n, n']}) \geq \frac{\tilde{t}}{4(t+1)^2|S|}$. Observe that this statement uses the n from Lemma 3. Nonetheless, it follows that it holds for any $n \in \mathbb{N}$ otherwise it would not hold either for the one given by Lemma 3 (as it needs to hold for infinitely many $n' > n$). \square

We make a simple observation regarding cycles that have a mean-payoff exceeding the threshold.

Lemma 5. *Consider a cycle $\rho = (e_i)_{1 \leq i \leq n}$ in G' , and write $\hat{\rho}_j = (s_j, c_j)$ for $0 \leq j \leq n$. Then:*

$$\text{MP}(\rho) > t = \frac{t_1}{t_2} \quad \Rightarrow \quad \sum_{i=1}^n c_i \geq \frac{t_1 n + 1}{t_2}.$$

Proof. Indeed, $\text{MP}(\rho) > \frac{t_1}{t_2}$ implies that $t_2 \cdot \sum_{i=1}^n c_i > t_1 \cdot n$. Since both sides are integers, it must be $t_2 \cdot \sum_{i=1}^n c_i \geq t_1 \cdot n + 1$, which entails the result. \square

Finally, we define the notion of *good cycle*, i.e., a cycle $\rho = (e_i)_{1 \leq i \leq n}$ such that $c_0 \leq t$ and $\text{MP}(\rho) \leq t$. That is, both the initial configuration and the mean-payoff of the cycle are below the threshold. Using the previous results, we prove that any winning play must contain a good cycle.

Lemma 6. *Let π be a play in G' from $(s_{\text{init}}, 0)$ with $\overline{\text{MP}}(\pi) \leq t$. Then there exist $1 \leq i \leq j$ such that $\pi_{[i, j]}$ is a good cycle.*

Proof. Pick $\gamma = (s, c)$ a frequent low-energy configuration as given by Lemma 4. First notice that for n and n' two indices given by Lemma 4, $\mathbf{d}(\{\gamma\}, \pi_{[n, n'-1]}) \geq \mathbf{d}(\{\gamma\}, \pi_{[n, n']})$ if $\hat{\pi}_{n'} \neq \gamma$. Hence we may reinforce Lemma 4 by stating that $\mathbf{d}(\{\gamma\}, \pi_{[n, n']}) \geq \frac{\tilde{t}}{4(t+1)^2|S|}$ for infinitely many n' such that $\hat{\pi}_{n'} = \gamma$. We can thus write $\pi = \rho \cdot \mathcal{C}_1 \cdot \mathcal{C}_2 \dots$ where $\text{last}(\rho) = \gamma$ and for all $i \geq 1$, $\text{first}(\mathcal{C}_i) = \text{last}(\mathcal{C}_i) = \gamma$, and $\mathbf{d}(\{\gamma\}, \mathcal{C}_1 \cdots \mathcal{C}_i) \geq \frac{\tilde{t}}{4(t+1)^2|S|}$. That is, we decompose π according to positions where γ appears and where the density lower bound holds.

Each cycle \mathcal{C}_i may itself contain several occurrences of γ . We can further decompose it as $\mathcal{C}_i = \mathcal{D}_i^1 \cdot \mathcal{D}_i^2 \cdots \mathcal{D}_i^{\ell_i}$ where for all $1 \leq j \leq \ell_i$, $\text{first}(\mathcal{D}_i^j) = \text{last}(\mathcal{D}_i^j) = \gamma$ and γ does not occur anywhere else in \mathcal{D}_i^j .

We claim that one of these cycles \mathcal{D}_i^j must be good. Toward a contradiction, assume that none is good. Then:⁹

$$\begin{aligned}
\overline{\text{MP}}(\pi) &= \overline{\text{MP}}(\pi_{>|\rho|}) && \text{(by prefix-independence of } \overline{\text{MP}}) \\
&\geq \limsup_{k \rightarrow \infty} \frac{\sum_{i=1}^k \sum_{j=1}^{\ell_i} \frac{1}{t_2} (t_1 |\mathcal{D}_i^j| + 1)}{\sum_{i=1}^k \sum_{j=1}^{\ell_i} |\mathcal{D}_i^j|} && \text{(by Lemma 5)} \\
&= t + \frac{1}{t_2} \limsup_{k \rightarrow \infty} \frac{\sum_{i=1}^k \sum_{j=1}^{\ell_i} 1}{\sum_{i=1}^k \sum_{j=1}^{\ell_i} |\mathcal{D}_i^j|} \\
&\geq t + \frac{1}{t_2} \limsup_{k \rightarrow \infty} \frac{\tilde{t}}{4(t+1)^2 |S|} && \text{(by Lemma 4)} \\
&= t + \frac{\tilde{t}}{4t_2(t+1)^2 |S|} > t
\end{aligned}$$

This contradicts the fact that π is winning in G' . Hence one of the cycles \mathcal{D}_i^j must be good. \square

Next we establish an upper-bound on the length of the shortest good cycle that can be observed along the path. First we prove the following lemma.

Lemma 7. *Let \mathcal{C} be a reachable good cycle with no good strict sub-cycle. Then $\ell(\mathcal{C}) \leq 8t_1 t_2 (t+1)^3 |S|^2$.*

Proof. We pick a good cycle $\mathcal{C} = (e_i)_{1 \leq i \leq n}$, and assume that it contains no good strict sub-cycle. We write $\hat{\mathcal{C}}_j = (s_j, c_j)$ for all $0 \leq j \leq n$. Let $\gamma \in \Gamma^{\leq t}$ be the most frequent configuration of $\Gamma^{\leq t}$ in $\hat{\mathcal{C}}$. Applying Lemma 4 to the winning play $\rho \cdot \mathcal{C}^\omega$, for some prefix ρ , we easily obtain that $\mathbf{d}(\{\gamma\}, \mathcal{C}) \geq \frac{\tilde{t}}{4(t+1)^2 |S|}$.

Write this cycle \mathcal{C} as $\rho_0 \cdot \mathcal{D}_1 \cdot \mathcal{D}_2 \dots \mathcal{D}_k \cdot \rho_1$, where γ occurs in ρ_0 only at the last position, in ρ_1 only at the first position, and \mathcal{D}_i are cycles containing γ only as first and last configurations. Then decompose ρ_0 and ρ_1 as $\rho_i^0 \cdot \mathcal{E}_i^1 \cdot \rho_i^1 \cdot \mathcal{E}_i^2 \dots \rho_i^{\ell_i}$ such that $\rho_i^0 \cdot \rho_i^1 \dots \rho_i^{\ell_i}$ does not contain cycles starting and ending in $\Gamma^{\leq t}$, and such that the \mathcal{E}_i^j are cycles. This can be performed by iteratively ‘‘factoring’’ any maximal cycle over configurations in $\Gamma^{\leq t}$, until no such cycle exists.

Let Ω be the following multiset of cycles:

$$\Omega = \{\mathcal{D}_i \mid 1 \leq i \leq k\} \cup \{\mathcal{E}_i^j \mid i \in \{0, 1\}, 1 \leq j \leq \ell_i\}.$$

We then write I_Ω for the set of positions $1 \leq i \leq |\mathcal{C}|$ along \mathcal{C} such that (s_i, c_i) belongs to some cycle $\omega \in \Omega$, and $J_\Omega = [1, n] \setminus I_\Omega$. Also, we define $J_\Omega^{\bowtie t} = \{i \in J_\Omega \mid c_i \bowtie t\}$ for $\bowtie \in \{\leq, >\}$. By hypothesis, the cycles ω in Ω are not good and are such that $(s, c) = \text{first}(\omega)$ satisfies $c \leq t$: hence it holds that $\text{MP}(\omega) > t$ for any cycle $\omega \in \Omega$. Thus

$$\begin{aligned}
\text{MP}(\mathcal{C}) &= \frac{\sum_{i \in I_\Omega} c_i + \sum_{i \in J_\Omega} c_i}{|\mathcal{C}|} \geq \frac{\sum_{\omega \in \Omega} \frac{t_1 |\omega| + 1}{t_2} + \sum_{i \in J_\Omega^{\leq t}} c_i + \sum_{i \in J_\Omega^{> t}} c_i}{\sum_{\omega \in \Omega} |\omega| + |J_\Omega^{\leq t}| + |J_\Omega^{> t}|} \\
&\geq \frac{\sum_{\omega \in \Omega} \frac{t_1 |\omega| + 1}{t_2} + \sum_{i \in J_\Omega^{\leq t}} 0 + \sum_{i \in J_\Omega^{> t}} c_i}{\sum_{\omega \in \Omega} |\omega| + |J_\Omega^{\leq t}| + |J_\Omega^{> t}|} \\
&\geq \frac{t \cdot \sum_{\omega \in \Omega} |\omega| + \frac{|\Omega|}{t_2} + ([t] + 1) \cdot |J_\Omega^{> t}|}{\sum_{\omega \in \Omega} |\omega| + 2 \cdot |S| \cdot ([t] + 1) + |J_\Omega^{> t}|}
\end{aligned}$$

where the first inequality follows from Lemma 5 and for the last one, we use $\sum_{i \in J_\Omega^{> t}} c_i \geq ([t] + 1) \cdot |J_\Omega^{> t}|$ (obvious) and $|J_\Omega^{\leq t}| \leq 2 \cdot |S| \cdot ([t] + 1)$ (because $J_\Omega^{\leq t}$ contains each configuration of $\Gamma^{\leq t}$ at most twice, by construction of the decompositions of ρ_0 and ρ_1 —at most once in each of them).

⁹ For the sake of readability, we sometimes write the length of a prefix $\ell(\rho)$ as $|\rho|$ in complicated expressions.

Now, we have that $|\Omega| \geq \mathbf{d}(\{\gamma\}, \mathcal{C}) \cdot |\mathcal{C}| \geq \frac{\tilde{t}}{4(t+1)^2|S|} \cdot |\mathcal{C}|$ and we deduce:

$$\frac{t \cdot \sum_{\omega \in \Omega} |\omega| + \frac{|\mathcal{C}|}{t_2} \frac{\tilde{t}}{4(t+1)^2|S|} + ([t] + 1) \cdot |J_{\Omega}^{\geq t}|}{\sum_{\omega \in \Omega} |\omega| + 2 \cdot |S| \cdot ([t] + 1) + |J_{\Omega}^{\geq t}|} \leq \text{MP}(\mathcal{C}).$$

Since $\text{MP}(\mathcal{C}) \leq t$, we get:

$$t \cdot \sum_{\omega \in \Omega} |\omega| + \frac{|\mathcal{C}|}{t_2} \cdot \frac{\tilde{t}}{4(t+1)^2|S|} + ([t] + 1) \cdot |J_{\Omega}^{\geq t}| \leq t \cdot (\sum_{\omega \in \Omega} |\omega| + 2 \cdot |S| \cdot ([t] + 1) + |J_{\Omega}^{\geq t}|)$$

and we deduce

$$\frac{|\mathcal{C}|}{t_2} \frac{\tilde{t}}{4(t+1)^2|S|} \leq t \cdot 2 \cdot |S| \cdot ([t] + 1) - |J_{\Omega}^{\geq t}| \cdot \tilde{t}.$$

From this, we obtain

$$\begin{aligned} |\mathcal{C}| &\leq \frac{8t_2t(t+1)^3|S|^2}{\tilde{t}} && \text{(since } \tilde{t} > 0 \text{ and } [t] + 1 \leq t + 1) \\ &\leq \frac{8t_1t_2(t+1)^3|S|^2}{t_2 - t'} && \text{(using } \tilde{t} = \frac{t_2 - t'}{t_2} \text{ and } t_2t = t_1) \\ &\leq 8t_1t_2(t+1)^3|S|^2 && \text{(using } t_2 - t' \geq 1) \end{aligned}$$

This concludes the proof. \square

For the rest of the paper, we set $N = 8t_1t_2(t+1)^3|S|^2$, a bound on the size of a minimal good cycle along a winning play.

Proposition 8. *Let π be a play in G' from $(s_{\text{init}}, 0)$ with $\overline{\text{MP}}(\pi) \leq t$. Then there exist $1 \leq i \leq j$ such that $\pi_{[i,j]}$ is a good cycle of length at most N .*

Proof. This is a consequence of Lemma 6 and Lemma 7. \square

5.2 Strategies Described by Finite Trees

So far, we have proven that P_0 should “target” short good cycles. However in a two-player context, P_1 might prevent P_0 from doing so. We therefore need to consider the *branching* behaviour of the game, and not only the linear point-of-view given by a play. Toward that aim, we represent strategies (of P_0) as *strategy trees*, and use them to bound the amount of memory and the counter values needed to win in G' .

We consider labelled trees with backward edges $\mathcal{T} = (\mathcal{N}, \mathcal{E}, \lambda, \dashrightarrow)$, where \mathcal{N} is a finite set of nodes, $\lambda: \mathcal{N} \rightarrow S \times \mathbb{N}$ (each node is labelled with a configuration of the game G'), and $\dashrightarrow \subseteq \mathcal{N} \times \mathcal{N}$. We assume \mathcal{T} has at least two nodes. The relation \mathcal{E} is the successor relation between nodes. A node with no \mathcal{E} -successor is called a *leaf*; other nodes are called *internal nodes*. The root of \mathcal{T} , denoted by \mathbf{n}_{root} , is the only node having no predecessor. The relation \dashrightarrow is an extra relation between nodes that will become clear later.

For such a tree to represent a strategy, we require that each internal node \mathbf{n} that is labelled by a P_0 -configuration (s, c) has only one successor \mathbf{n}' , with $\lambda(\mathbf{n}') = (s', c')$ such that there is a transition $((s, c), c', (s', c'))$ in the game G' ; we require that each internal node \mathbf{n} that is labelled with a P_1 -state (s, c) has exactly one successor per transition $((s, c), c', (s', c'))$ in G' , each successor being labelled with its associated (s', c') . Finally, we require that each leaf \mathbf{n} of \mathcal{T} has a (strict) ancestor node \mathbf{n}' such that $\lambda(\mathbf{n}') = \lambda(\mathbf{n})$. The relation \dashrightarrow will serve witnessing that property. So we assume that for every leaf \mathbf{n} , there is a unique ancestor node \mathbf{n}' such that $\mathbf{n} \dashrightarrow \mathbf{n}'$; furthermore it should be the case that $\lambda(\mathbf{n}') = \lambda(\mathbf{n})$. Under all these

constraints, \mathcal{T} is called a *strategy tree*. It basically represents a (finite) memory structure for a strategy, as we now explain.

Let $\mathcal{T} = (\mathcal{N}, \mathcal{E}, \lambda, \dashrightarrow)$ be strategy tree for G' . We define $\mathcal{G}_{\mathcal{T}} = (\mathcal{N}, \mathcal{E}')$, a directed graph obtained from \mathcal{T} by adding extra edges $(\mathbf{n}, \mathbf{n}'')$ for each leaf \mathbf{n} and node \mathbf{n}'' for which there exists another node \mathbf{n}' satisfying $\mathbf{n} \dashrightarrow \mathbf{n}'$ and $(\mathbf{n}', \mathbf{n}'') \in \mathcal{E}$. We refer to these extra edges as *back-edges*. One may notice that for any $(\mathbf{n}, \mathbf{n}') \in \mathcal{E}'$ there is an edge from $\lambda(\mathbf{n})$ to $\lambda(\mathbf{n}')$ in G' . Given two nodes \mathbf{n} and \mathbf{n}' such that \mathbf{n}' is an ancestor of \mathbf{n} in \mathcal{T} , we write $[\mathbf{n}' \rightsquigarrow \mathbf{n}]$ for the play prefix from \mathbf{n}' to \mathbf{n} (inclusive) using only transitions from \mathcal{E} .

Now, we associate with any prefix ρ in $\mathcal{G}_{\mathcal{T}}$ from \mathbf{n}_{root} a prefix $\bar{\rho}$ in G' from $\lambda(\mathbf{n}_{\text{root}}) = (s_{\text{root}}, c_{\text{root}})$ such that $\text{last}(\bar{\rho}) = \lambda(\text{last}(\rho))$. The construction is inductive:

- with the empty prefix in $\mathcal{G}_{\mathcal{T}}$ we associate the empty prefix in G' : $\overline{\epsilon_{\mathbf{n}_{\text{root}}}} = \epsilon_{(s_{\text{root}}, c_{\text{root}})}$,
- if $\rho = \rho' \cdot (\mathbf{n}', \mathbf{n})$ with $(\mathbf{n}, \mathbf{n}') \in \mathcal{E}'$, writing $(s', c') = \lambda(\mathbf{n}')$ and $(s, c) = \lambda(\mathbf{n})$, then $\bar{\rho} = \bar{\rho}' \cdot ((s', c'), c, (s, c))$ (which by construction is indeed a prefix in G').

We now explain how $\mathcal{G}_{\mathcal{T}}$ corresponds to a strategy in G' : for any prefix ρ in $\mathcal{G}_{\mathcal{T}}$, if $\lambda(\text{last}(\rho)) = (s, c) \in \Gamma_0$, then $\text{last}(\rho)$ has a unique successor \mathbf{n}' in $\mathcal{G}_{\mathcal{T}}$, and, writing $(s', c') = \lambda(\mathbf{n}')$, we define $\sigma_{\mathcal{T}}(\bar{\rho}) = ((s, c), c', (s', c'))$: $\sigma_{\mathcal{T}}$ is a (partially defined) strategy in G' . The following lemma states that $\mathcal{G}_{\mathcal{T}}$ actually represents the outcomes of the well-defined strategy $\sigma_{\mathcal{T}}$ from $\lambda(\mathbf{n}_{\text{root}})$ in G' :

Lemma 9. *Let μ be a prefix in G' from $(s_{\text{root}}, c_{\text{root}})$. Assume that for any $i \leq \ell(\mu)$ such that $\text{last}(\mu_{\leq i}) \in \Gamma_0$, the function $\sigma_{\mathcal{T}}$ is defined on $\mu_{\leq i}$ and $\mu_{\leq i+1} = \mu_{\leq i} \cdot \sigma_{\mathcal{T}}(\mu_{\leq i})$. Then there exists a unique prefix ρ in $\mathcal{G}_{\mathcal{T}}$ such that $\mu = \bar{\rho}$. Moreover, if $\text{last}(\mu) \in \Gamma_0$, then $\sigma_{\mathcal{T}}(\mu)$ is defined.*

Proof. The proof is by induction on $\ell(\mu)$. The base case where μ is the empty prefix is straightforward.

Now, assume that the result holds for a prefix μ' , and consider a prefix $\mu = \mu' \cdot ((s', c'), c, (s, c))$ satisfying the hypotheses of the lemma. Then μ' also satisfies those hypotheses, and we have $\mu' = \bar{\rho}'$ for some prefix ρ' in $\mathcal{G}_{\mathcal{T}}$. Then, each successor \mathbf{n} of $\mathbf{n}' = \text{last}(\rho')$ is associated with a transition of G' from $(s', c') = \lambda(\mathbf{n}')$. We consider two cases:

- if $(s', c') \in \Gamma_0$ is a P_0 -state, then, by hypothesis, $\sigma_{\mathcal{T}}(\mu') = ((s', c'), c, (s, c))$, and the unique successor node \mathbf{n} of \mathbf{n}' in $\mathcal{G}_{\mathcal{T}}$ is indeed labelled with (s, c) . We can thus let $\rho = \rho' \cdot (\mathbf{n}', \mathbf{n})$.
- if $(s', c') \notin \Gamma_0$ belongs to P_1 , then each transition of G' corresponds to a successor \mathbf{n} of \mathbf{n}' . In particular, there is a unique successor \mathbf{n} corresponding to $((s', c'), c, (s, c))$, and again we can let $\rho = \rho' \cdot (\mathbf{n}', \mathbf{n})$.

The last statement follows by noticing that $\sigma_{\mathcal{T}}(\bar{\rho})$ is defined for all prefixes ρ such that $\lambda(\text{last}(\rho)) \in \Gamma_0$. \square

We now give conditions for $\sigma_{\mathcal{T}}$ to be a winning strategy from $(s_{\text{root}}, c_{\text{root}})$ in G' . With a finite outcome $\mu = \bar{\rho}$ of $\sigma_{\mathcal{T}}$ from $(s_{\text{root}}, c_{\text{root}})$, we associate a sequence $\text{decomp}_{\mathcal{T}}(\mu)$ of cycles in G' , defined inductively as follows:

- $\text{decomp}_{\mathcal{T}}(\epsilon_{(s_{\text{root}}, c_{\text{root}})})$ is empty;
- if $\rho = \rho' \cdot (\mathbf{n}', \mathbf{n})$ and \mathbf{n} is not a leaf of \mathcal{T} , then $\text{decomp}_{\mathcal{T}}(\bar{\rho}) = \text{decomp}_{\mathcal{T}}(\bar{\rho}')$;
- if $\rho = \rho' \cdot (\mathbf{n}', \mathbf{n})$ and \mathbf{n} is a leaf of \mathcal{T} , we let \mathbf{n}'' be such that $\mathbf{n} \dashrightarrow \mathbf{n}''$; the prefix $[\mathbf{n}'' \rightsquigarrow \mathbf{n}]$ in \mathcal{T} corresponds to a cycle C in G' , and we let $\text{decomp}_{\mathcal{T}}(\bar{\rho}) = \text{decomp}_{\mathcal{T}}(\bar{\rho}') \cdot C$.

The sequence $\text{decomp}_{\mathcal{T}}(\bar{\rho})$ hence contains all full cycles (closed at leaves) encountered while reading $\bar{\rho}$ in \mathcal{T} : hence it comprises all edges of $\bar{\rho}$ except a prefix starting at \mathbf{n}_{root} and a suffix since the last back-edge has been taken. It is not hard to see that those can actually be concatenated. By induction, we can easily show:

Proposition 10. *Let μ be a non-empty finite outcome of $\sigma_{\mathcal{T}}$ from $(s_{\text{root}}, c_{\text{root}})$ in G' . Write $\text{decomp}_{\mathcal{T}}(\mu) = C_0 \cdot C_1 \cdot \dots \cdot C_h$ (where each C_i is a cycle). Let ρ be the prefix in $\mathcal{G}_{\mathcal{T}}$ such that $\mu = \bar{\rho}$, $\mathbf{n} = \text{last}(\rho)$, and $\nu = [\mathbf{n}_{\text{root}} \rightsquigarrow \mathbf{n}]$. Write $(s_j, c_j) = \lambda(\nu_j)$. Then:*

$$\text{MP}(\mu) = \frac{\sum_{i=0}^h \text{MP}(C_i) \cdot \ell(C_i) + \sum_{j=1}^{\ell(\nu)} c_j}{\ell(\mu)}$$

We say that a tree is *good* if, for every $\mathbf{n} \dashrightarrow \mathbf{n}'$ in \mathcal{T} , writing $\rho = [\mathbf{n}' \rightsquigarrow \mathbf{n}]$ and letting $\lambda(\hat{\rho}_j) = (s_j, c_j)$, it holds $\frac{\sum_{j=1}^{\ell(\rho)} c_j}{\ell(\rho)} \leq t$.

Proposition 11. *If \mathcal{T} is a finite good strategy tree, then $\sigma_{\mathcal{T}}$ is a winning strategy from $(s_{\text{root}}, c_{\text{root}})$ in G' .*

Proof. We apply Proposition 10; the sum $\sum_{j=1}^{\ell(\nu)} c_j$ is bounded because \mathcal{T} is finite, and $\text{MP}(C_i) \cdot \ell(C_i) \leq t \cdot \ell(C_i)$ since \mathcal{T} is good. Hence for any infinite outcome μ of $\sigma_{\mathcal{T}}$ from $(s_{\text{root}}, c_{\text{root}})$, we get $\overline{\text{MP}}(\mu) \leq t$. \square

Note that \mathcal{T} can be interpreted as a *finite memory structure* for strategy $\sigma_{\mathcal{T}}$: to know which move is given by $\sigma_{\mathcal{T}}$, it is sufficient to move a token in tree \mathcal{T} , going down in the tree, and following back-edges when stuck at leaves.

5.3 Analyzing Winning Strategies

We proved in the previous section that the finite-memory strategy associated with a finite good strategy tree is winning. In this section, we first show the converse direction, proving that from a winning strategy, we can obtain a finite good tree. In that tree, backward edges correspond to (short) good cycles. We will then use the result of Section 4 for showing that those parts of the tree that do not belong to a segment $[\mathbf{n} \rightsquigarrow \mathbf{n}']$ with $\mathbf{n}' \dashrightarrow \mathbf{n}$ can be replaced with other substrategies in which the counter value is uniformly bounded. That way, we show that if there is a winning strategy for our games, then there is one where the counter value is uniformly bounded along all outcomes. This will allow to apply algorithms for solving games with average-energy payoff under lower- and upper-bound constraints [5].

Fix a winning strategy σ for P_0 from (s_0, c_0) in G' . We let

$$\text{goodpref}(\sigma) = \{ \pi_{\leq n} \mid \pi \in \text{Out}((s_0, c_0), \sigma), \text{ there is } m < n \text{ s.t. } \pi_{[m+1, n]} \text{ is a good cycle with no good strict sub-cycle and } \pi_{\leq n-1} \text{ does not contain any good cycle} \}$$

and for every $\pi_{\leq n} \in \text{goodpref}(\sigma)$, we define $\text{back}(\pi_{\leq n})$ as $\pi_{\leq m}$ such that $\pi_{[m+1, n]}$ is a good cycle with no good strict sub-cycle.

We build a strategy tree \mathcal{T}_{σ} as follows:

- the nodes of \mathcal{T}_{σ} are all the prefixes of the finite plays in $\text{goodpref}(\sigma)$; the edges relate each prefix of length k to its extensions of length $k+1$. For a node \mathbf{n} corresponding to prefix $\rho_{\leq k}$ (with $\rho \in \text{goodpref}(\sigma)$), we let $\lambda(\mathbf{n}) = \text{last}(\rho_{\leq k})$; we let $\lambda(\mathbf{n}_{\text{root}}) = (s_0, c_0)$. The leaves of \mathcal{T}_{σ} then correspond to the play prefixes that are in $\text{goodpref}(\sigma)$.
- backward edges in \mathcal{T}_{σ} are defined by noticing that each leaf \mathbf{n} of \mathcal{T}_{σ} corresponds to a finite path $\pi_{\leq n}$ in $\text{goodpref}(\sigma)$, so that the prefix $\pi_{\leq m} = \text{back}(\pi_{\leq n})$ is associated with some node \mathbf{m} of \mathcal{T}_{σ} such that $\pi_{[m+1, n]}$ is a good cycle. This implies $\lambda(\pi_{\leq n}) = \lambda(\pi_{\leq m})$, and we define $\mathbf{n} \dashrightarrow \mathbf{m}$. This way, \mathcal{T}_{σ} is a strategy tree as defined in Section 5.2.

Lemma 12. *Tree \mathcal{T}_{σ} is a finite good strategy tree.*

Proof. If \mathcal{T}_{σ} were infinite, by König's Lemma it would contain an infinite branch. This infinite branch would correspond to an infinite outcome of σ containing no good cycle, contradicting Lemma 6.

The fact that the tree is good is because all prefixes selected by $\text{goodpref}(\sigma)$ correspond to good cycles. \square

Applying Proposition 11, we immediately get:

Corollary 13. *Strategy $\sigma_{\mathcal{T}_{\sigma}}$ is a winning strategy from (s_0, c_0) .*

Let $\mathbf{n} \dashrightarrow \mathbf{n}'$ be two related nodes in \mathcal{T}_{σ} . We say that a node \mathbf{n}'' is *just below* $[\mathbf{n}' \rightsquigarrow \mathbf{n}]$ in \mathcal{T}_{σ} whenever its predecessor appears along $[\mathbf{n}' \rightsquigarrow \mathbf{n}]$, but node \mathbf{n}'' itself does not appear along any path $[\mathbf{n}_1 \rightsquigarrow \mathbf{n}_2]$ for which $\mathbf{n}_2 \dashrightarrow \mathbf{n}_1$. Such nodes, together with the root of the tree, are called the *critical* nodes. Fig. 3 illustrates this definition.

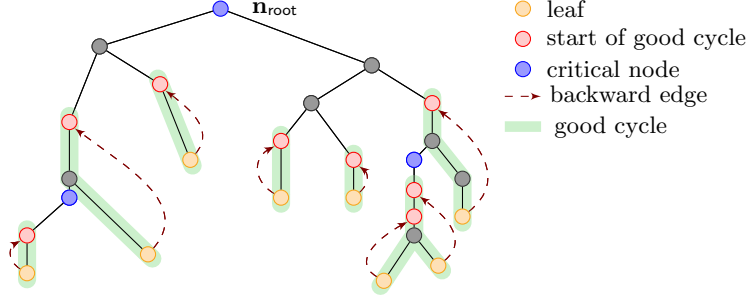


Fig. 3: Example of a finite strategy tree, with backward edges and critical nodes.

Lemma 14. *If \mathbf{n} is a critical node in \mathcal{T}_σ , then writing $\lambda(\mathbf{n}) = (s, c)$, we have that $c \leq t + W \cdot (N + 1)$.*

Proof. By Proposition 8, for every $\mathbf{n} \dashrightarrow \mathbf{n}'$, the length of the prefix $[\mathbf{n}' \rightsquigarrow \mathbf{n}]$ in \mathcal{T}_σ is at most N . So the length of the prefix from \mathbf{n}' to a critical node is at most $N + 1$. Since node \mathbf{n}' corresponds to the beginning of a good cycle, we have $\lambda(\mathbf{n}') = (s', c')$ with $c' \leq t$. Finally, since edges in G' have weight at most W , we get our result. \square

Given a critical node \mathbf{n} , we define

$$\text{target}(\mathbf{n}) = \{ \mathbf{n}' \text{ in subtree of } \mathbf{n} \mid \text{there exists } \mathbf{n}'' \text{ such that } \mathbf{n}'' \dashrightarrow \mathbf{n}' \\ \text{and } [\mathbf{n} \rightsquigarrow \mathbf{n}'] \text{ contains no other such node} \}.$$

Looking again at Fig. 3, the targets of a critical node are the start nodes of the good cycles that are closest to that critical node. In particular, for the rightmost critical node on Fig. 3, there are two candidate target nodes (because there are two overlapping good cycles), but only the topmost one is a target.

For every critical node \mathbf{n} , we apply Lemma 2 with $\gamma = \lambda(\mathbf{n})$ and $I'_\gamma = \{ \gamma' = \lambda(\mathbf{n}') \mid \mathbf{n}' \in \text{target}(\mathbf{n}) \}$, setting $M = t + W \cdot (N + 1)$. We write $\sigma_{\mathbf{n}}$ for the corresponding strategy: applying $\sigma_{\mathbf{n}}$ from $\lambda(\mathbf{n})$, player P_0 will reach some configuration (s', c') such that there is a node $\mathbf{n}' \in \text{target}(\mathbf{n})$ with $\lambda(\mathbf{n}') = (s', c')$.

Now, for any node \mathbf{n}' that is the target of a backward edge $\mathbf{n} \dashrightarrow \mathbf{n}'$, but whose immediate predecessor does not belong to any segment $[\mathbf{n}_1 \rightsquigarrow \mathbf{n}_2]$ with $\mathbf{n}_2 \dashrightarrow \mathbf{n}_1$, we define strategy $\sigma_{[\mathbf{n}]}$ which follows good cycles as much as possible; when a leaf \mathbf{m} is reached, the strategy replays similarly as from the equivalent node \mathbf{m}' for which $\mathbf{m} \dashrightarrow \mathbf{m}'$. If, while playing that strategy, the play ever leaves a good cycle (due to a move of player P_1), then it reaches a critical node \mathbf{n}'' . From that node, we will apply strategy $\sigma_{\mathbf{n}''}$ as defined above, and iterate like this.

This defines a strategy σ' . Applying Lemma 2 to strategies $\sigma_{\mathbf{n}}$ when \mathbf{n} is critical, and the previous analysis of good cycles, we get the following doubly-exponential bound on the counter value (which is only exponential in case constants W , t_1 , and t_2 are encoded in unary):

Proposition 15. *Strategy σ' is a winning strategy from (s_0, c_0) , and all visited configurations (s, c) when applying σ' are such that $c \leq M'$ with*

$$M' = 2^{\mathcal{O}(t+W \cdot (8t_1t_2(t+1)^3|S|^2+1)+|S|+|E| \cdot W+|S| \cdot (\lceil t \rceil+1))}.$$

5.4 Conclusion

Gathering everything we have done above, we get the following equivalence.

Proposition 16. *Player P_0 has a winning strategy in game G from s_{init} for the objective $\text{AvgEnergy}_L(t)$ if, and only if, he has a winning strategy in G from s_{init} for the objective $\text{AvgEnergy}_{LU}(t, U) = \text{Energy}_{LU}(U) \cap \text{AvgEnergy}(t)$, where $U = M'$ is the bound from Proposition 15.*

Hence we can use the algorithm for games with objectives $AvgEnergy_{LU}(t, U)$ in [5], which is polynomial in $|S|$, $|E|$, t , and U (hence pseudo-polynomial only). Having in mind that the upper bound U is doubly-exponential, we can deduce our main decidability result. The memory required is also a consequence of [5].

Theorem 17. *The AEL threshold problem is in 2-EXPTIME. Furthermore doubly-exponential memory is sufficient to win (for player P_0).*

In [16], a super-exponential lower bound is given for the required memory to win a succinct one-counter game. While the model of games is not exactly the same, the actual family of games witnessing that lower bound on the memory happens to be usable as well for the AEL threshold problem (with threshold zero). The reduction is similar to the one in the proof of Theorem 20. This yields a lower bound on the required memory to win games with $AvgEnergy(t)$ objectives which is $2^{(2^{\sqrt{n}}/\sqrt{n})-1}$.

For unary encodings or small weights we get better results from our technique:

Corollary 18. *The AEL threshold problem is in EXPTIME and exponential memory is sufficient to win (for player P_0), if the weights and the threshold are encoded in unary or polynomial in the size of the graph.*

6 EXPSPACE-hardness

In this section, we show that the AEL threshold problem is EXPSPACE-hard by a reduction from succinct one-counter reachability games with counter values in \mathbb{N} , thereby improving the previously best EXPTIME lower bound. Such a game is played in a graph (S_0, S_1, E) as defined above, but without parallel edges. However, the semantics are slightly different: assume a play prefix ρ has been produced by the players thus far. Then, the player whose turn it is may only pick an edge e such that $EL(\rho) + w(e) \geq 0$, i.e., edges that would lead to a negative energy level are disabled. If the player has no enabled edges available, then he loses immediately. Note that this indeed differs from the semantics of the games we consider here, where a negative energy level is a direct loss for P_0 , no matter who picked the edge leading to the negative level. The goal of P_0 in a succinct one-counter game for is to reach, from a given initial state $s_{\text{init}} \in S$ with energy level zero, a state $s \neq s_{\text{init}}$ with energy level zero.

Proposition 19 ([17]). *The following problem is EXPSPACE-complete: given a graph (S_0, S_1, E) without parallel edges and a vertex $s_{\text{init}} \in S$, has P_0 a strategy (respecting the blocking semantics described above) so that every outcome is infinite and has a prefix that ends in a state $s \neq s_{\text{init}}$ and has energy level zero.*

We reduce this problem to the AEL threshold problem by simulating a one-counter reachability game by an AEL game: the reachability objective can easily be encoded by giving P_0 the ability to stop the simulation at any state $s \neq s_{\text{init}}$ by going (with weight zero) to a fresh sink state s_{sink} that is equipped with a self-loop of weight zero. Then, if he ever encounters a prefix of energy level zero, he can stop the simulation, go to the sink state, and achieve an average-energy of zero. On the other hand, if he never reaches energy level zero during the simulation, he has two options: either simulating the one-counter game ad infinitum while never reaching energy level zero or going to the sink state with non-zero energy level. In both cases, the average-energy is greater than zero. Thus, the reachability objective can be taken care of by an average-energy objective with threshold zero.

It remains to show how to implement the disabling of edges leading to a negative energy level. As already remarked above, such edges lead to a direct loss for P_0 in an average-energy game. Thus, a winning strategy for P_0 never picks a disabled edge, i.e., we only have to consider the case of disabled edges at vertices of P_1 . Intuitively, we do the following: every time P_1 picks an edge e with negative weight $w(e)$, P_0 has the choice to let this edge be taken or to claim that the energy level is strictly below $-w(e)$. If his claim is correct, he will be able to reach the sink state s_{sink} with energy level zero, and thereby ensure an average-energy of zero. In contrast, if his claim is incorrect, then he either reaches a negative energy level or cannot ensure an average-energy of zero.

As an example, assume there is an edge $e = (s, -13, s')$ in the one-counter game. It is replaced by the gadget shown in Fig. 4, where all states but s and s' are fresh. It is straightforward to show that P_0 can,

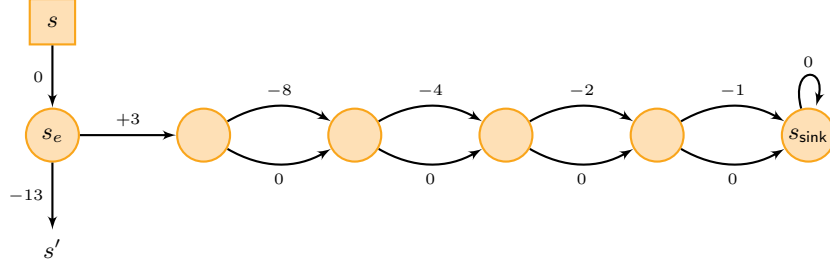


Fig. 4: The gadget implementing the blocking of edges (here, $e = (s, -13, s')$): if a play prefix ends in s with energy level c , then P_0 can reach s_{sink} with energy level zero while maintaining non-negative energy levels if, and only if, $c \leq 12$. If $c \geq 13$ he has to go to s' when maintaining non-negative energy levels.

when at from s_e , enter the gadget (i.e., not go to s') and ensure an non-negative energy level and average energy level zero if, and only if, the current energy level at s is at most 12, by reaching the sink state with energy level zero. On the other hand, if the energy level is larger than 12, then he cannot reach the sink with energy level zero. Thus, he has to take the edge to s' as intended by P_1 .

Obviously, this gadget can be generalized to arbitrary negative weights $w(e)$ with $\log(w(e))$ many states, which may even be shared among gadgets.

Using the encoding of the reachability objective by an average-energy objective and the blocking-gadget described above, it is straightforward to reduce succinct once-counter games with counters in \mathbb{N} to the AEL threshold problem with threshold zero.

Theorem 20. *The AEL threshold problem is EXPSPACE-hard, even for the fixed threshold zero.*

7 Multi-dimensional Average-energy Games

We now turn to a more general class of games where integer weights on the edges are replaced by *vectors of integer weights*, representing changes in different quantitative aspects. That is, for a game $G = (S_0, S_1, E)$ of dimension $k \geq 1$, we now have $E \subseteq S \times [-W, W]^k \times S$ for $W \in \mathbb{N}$. Multi-dimensional games have recently gained interest as a powerful model to reason about interplays and trade-offs between different resources; and multi-dimensional versions of many classical objectives have been considered in the literature: e.g., mean-payoff [11,26], energy [11,19,26], or total-payoff [10]. We consider the natural extensions of threshold problems in the multi-dimensional setting: we take the zero vector in \mathbb{N}^k as lower bound for the energy, a vector $U \in \mathbb{N}^k$ as upper bound, a vector $t \in \mathbb{Q}^k$ as threshold for the average-energy, and the payoff functions are defined using component-wise limits. That is, we essentially take the *conjunction* of our objectives for all dimensions. We quickly review the situation for the three types of average-energy objectives.

Average-energy games (without energy bounds). In the one-dimensional version of such games, memoryless strategies suffice for both players and the threshold problem is in $\text{NP} \cap \text{coNP}$ [5]. We prove here that already for games with three dimensions, the threshold problem is undecidable, based on a reduction from two-dimensional robot games [23]. Decidability for average-energy games with two dimensions remains open.

Theorem 21. *The threshold problem for average-energy games with three or more dimensions is undecidable. That is, given a finite k -dimensional game $G = (S_0, S_1, E)$, for $k \geq 3$, an initial state $s_{\text{init}} \in S$, and a threshold $t \in \mathbb{Q}^k$, determining whether P_0 has a winning strategy from s_{init} for objective $\text{AvgEnergy}(t)$ is undecidable.*

Proof. Two-dimensional robot games [23] are a special case of counter reachability games, expressible as follows: $R = (\{q_0\}, \{q_1\}, T)$, where q_0 (resp. q_1) is the unique state belonging to P_0 (resp. P_1), $Q = \{q_0, q_1\}$, and $T \subseteq Q \times [-V, V]^2 \times Q$, $V \in \mathbb{N}$, is a finite set of transitions. The game starts in q_0 with given counter

values $(x_0, y_0) \in \mathbb{Z}^2$ and, when in configuration $(q, (x, y))$, taking transition $t = (q, (a, b), q')$ takes the game to configuration $(q', (x+a, y+b))$. The goal of P_0 is to reach configuration $(q_0, (0, 0))$ and P_1 tries to prevent it. It was recently proved that deciding the winner in such games is undecidable [23].

Given $R = (\{q_0\}, \{q_1\}, T)$ with initial configuration $(q_0, (x_0, y_0))$, we build a three-dimensional average-energy game G with threshold $(0, 0, 0)$ such that P_0 wins in G if, and only if, he wins in R , which implies undecidability of the AEL threshold problem. Let $G = (S_0, S_1, E)$ with $S_0 = \{q_{\text{init}}, q_0, q_{\text{stop}}\}$ (with q_{init} and q_{stop} being fresh states), $S_1 = \{q_1\}$ and $E \subseteq S \times [-W, W]^3 \times S$, $W \in \mathbb{N}$, built as follows:

- if $(q, (a, b), q') \in T$, then $(q, (a, b, -a - b), q') \in E$,
- $(q_0, (-1, -1, -1), q_{\text{stop}}) \in E$ and $(q_{\text{stop}}, (0, 0, 0), q_{\text{stop}}) \in E$,
- $(q_{\text{init}}, (x_0 + 1, y_0 + 1, -x_0 - y_0 + 1), q_0) \in E$ (where (x_0, y_0) are the initial counter values in R).

Essentially, we add a third dimension that contains the opposite of the sum of the first two dimensions of each transition in T ; we add the possibility to branch from q_0 to q_{stop} using a $(-1, -1, -1)$ -edge to reach an absorbing state with a $(0, 0, 0)$ -loop; and we also add an initial edge that encodes the initial configuration of R . Now, the initial state of G is q_{init} and the threshold is $t = (0, 0, 0)$.

We proceed to prove the reduction to be correct. First, let σ_0 be a winning strategy for $\text{AvgEnergy}(t)$ in G . We will show that P_0 can also win in R . We start by claiming that P_0 has to branch to q_{stop} at some point otherwise he cannot win. If the $(-1, -1, -1)$ -edge is not taken along a play π , then it holds that for all $i \geq 1$, the energy level $\text{EL}(\pi_{\leq i}) = (x_i, y_i, z_i)$ satisfies $x_i + y_i + z_i = 3$. Indeed, it holds in the first step thanks to the initial edge and it continues to hold at each step as for each edge issued from T , the sum of the three dimensions is zero. Now observe the following:

$$\begin{aligned} 3 &= \limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n (x_i + y_i + z_i) \\ &\leq \limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n x_i + \limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n y_i + \limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n z_i \\ &= x + y + z \end{aligned}$$

for $(x, y, z) = \overline{\text{AE}}(\pi)$. Hence, at least one dimension has an average-energy strictly greater than zero (otherwise their sum cannot be greater than or equal to three), and the threshold $t = (0, 0, 0)$ cannot be met. Thus, we know that branching to q_{stop} is necessary since σ_0 is winning. But using the decomposition lemma of [5] and the fact that q_{stop} is an absorbing state with a $(0, 0, 0)$ -loop, we also know that for the average-energy to be less than $(0, 0, 0)$, the energy level when branching must be no more than $(1, 1, 1)$ (as we will use a $(-1, -1, -1)$ -edge). Furthermore, if the energy level on some dimension is strictly less than one, then it must be strictly greater on another dimension (because the sum is always equal to three before branching), and we are not below $(1, 1, 1)$ either. In conclusion, strategy σ_0 must ensure reaching the exact configuration $(q_0, (1, 1, 1))$ in order to win. Finally, observe that by construction of our game G , reaching this configuration is equivalent to reaching $(q_0, (0, 0))$ in R . Hence, we can easily build a strategy σ_0^R in R that mimics σ_0 in order to win the robot game. This strategy σ_0^R could in general use arbitrary memory (since we start with an arbitrary strategy σ_0) while robot games as defined in [23] only allow strategies to look at the current configuration. Still, from σ_0^R , one can easily build a corresponding strategy that meets this restriction (R being a counter reachability game, there is no reason to choose different actions in two visits of the same configuration).

Similarly, from a winning strategy σ_0^R in R , we can define a strategy σ_0 that mimics it in G in order to reach $(q_0, (1, 1, 1))$, and at that point, branches to q_{stop} and wins in G for $\text{AvgEnergy}(t)$. Thus, P_0 wins in G if, and only if, he wins in R , which concludes our proof. \square

Average-energy games with lower and upper bounds. One-dimensional versions of those games were proved to be EXPTIME-complete in [5]. The algorithm consists in reducing (in two steps) the original game to a mean-payoff game on an expanded graph of pseudo-polynomial size (polynomial in the original game but

also in the upper bound $U \in \mathbb{N}$) and applying a pseudo-polynomial time algorithm for mean-payoff games (e.g., [6]). Intuitively, the trick is that the bounds give strong constraints on the energy levels that can be visited along a play without losing and thus one can restrict the game to a particular graph where acceptable energy levels are encoded in the states and exceeding the bounds is explicitly represented by moving to “losing” states, just as we did in Section 3 for the lower bound. Carefully inspecting the construction of [5], we observe that the same construction can be generalized straightforwardly to the multi-dimensional setting. However, the overall complexity is higher: first, the expanded graph will be of *exponential size in k* , the number of dimensions, while still polynomial in S and U . Second, multi-dimensional *limsup* mean-payoff games are in $\text{NP} \cap \text{coNP}$ [26].

Theorem 22. *The threshold problem for multi-dimensional average-energy games with lower and upper bounds is in $\text{NEXPTIME} \cap \text{coNEXPTIME}$. That is, given a finite k -dimensional game $G = (S_0, S_1, E)$, an initial state $s_{\text{init}} \in S$, an upper bound $U \in \mathbb{N}^k$, and a threshold $t \in \mathbb{Q}^k$, determining if P_0 has a winning strategy from s_{init} for objective $\text{Energy}_{LU}(U) \cap \text{AvgEnergy}(t)$ is in $\text{NEXPTIME} \cap \text{coNEXPTIME}$.*

Whether the EXPTIME-hardness that trivially follows from the one-dimensional case [5] can be enhanced to meet this upper bound (or conversely) is an open problem.

Average-energy games with lower bound but no upper bound. Finally, we consider the core setting of this paper, which we just proved decidable in one-dimension, solving the open problem of [5]. Unfortunately, we show that those games are undecidable *as soon as two-dimensional weights are allowed*. To prove it, we reuse some ideas of the proof of undecidability for multi-dimensional total-payoff games presented in [10], but specific gadgets need to be adapted. Hence we provide a full proof here.

Theorem 23. *The threshold problem for lower-bounded average-energy games with two or more dimensions is undecidable. That is, given a finite k -dimensional game $G = (S_0, S_1, E)$, for $k \geq 2$, an initial state $s_{\text{init}} \in S$, and a threshold $t \in \mathbb{Q}^k$, determining whether P_0 has a winning strategy from s_{init} for objective $\text{AvgEnergy}_L(t)$ is undecidable.*

Proof. We reduce the halting problem for two-counter machines (2CMs) to the threshold problem for lower-bounded average-energy games with two dimensions. From a two-counter machine \mathcal{M} , we construct a two-player game G with two dimensions and a lower-bounded average-energy objective such that P_0 wins for threshold $(0, 0)$ if, and only if, the 2CM halts. Counters take values $(v_1, v_2) \in \mathbb{N}^2$ along an execution, and can be incremented or decremented (if positive). A counter can be tested for equality to zero, and the machine can branch accordingly. The halting problem for 2CMs is undecidable [22]. Assume w.l.o.g. that we have a 2CM \mathcal{M} such that if it halts, it halts with the two counters equal to zero, which is possible as it suffices to plug a machine that decreases both counters to zero at the end of the execution of the considered machine. In the game we construct, P_0 has to faithfully simulate the 2CM \mathcal{M} and wins only if it halts. The role of P_1 is to ensure that he does so by retaliating if it is not the case, hence making the outcome losing for the lower-bounded average-energy objective.

The game is built as follows. The states of G are copies of the control states of \mathcal{M} plus some special states discussed in the following. Edges represent transitions between these states. Let C_1 and C_2 be the two counters, starting with value zero. We start the game by taking an edge of weight $(1, 1)$ and we build our game such that, at all times along a faithful execution, the counters have value (v_1, v_2) if, and only, if the current energy level is $(v_1 + 1, v_2 + 1)$. Each increment of C_1 (resp. C_2) in \mathcal{M} is implemented in G by an edge of weight $(1, 0)$ (resp. $(0, 1)$); each decrement of C_1 (resp. C_2) by an edge of weight $(-1, 0)$ (resp. $(0, -1)$).

We first present how to ensure a faithful simulation of the 2CM \mathcal{M} by P_0 . The necessary gadgets are depicted in Fig. 5.

Increments and decrements of counters are easily simulated as presented above.

Values of counters may never go below zero. To ensure this, we allow P_1 to branch after every step of the 2CM simulation to a special gadget represented in Subfig. 5(a). If P_0 cheats by reaching a negative value on counter C_1 or C_2 , P_1 can easily win by branching: indeed, say C_1 has value $v_1 < 0$, then the energy level on the first dimension before branching is a most zero and because of the $(-1, -1)$ edge, the energy becomes

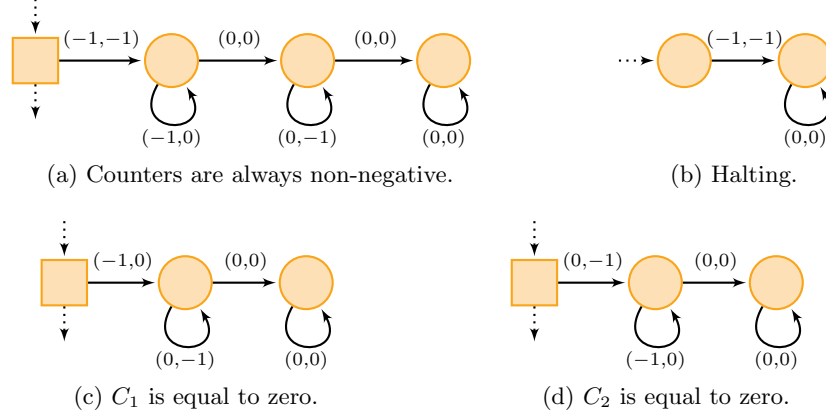


Fig. 5: Gadgets for the reduction from the 2CM halting problem to lower-bounded average-energy games.

negative and P_0 loses. On the contrary, if P_0 does not cheat and maintains both counters non-negative at all times, taking the $(-1, -1)$ edge is safe and the energy level after this edge is (v_1, v_2) . Then, P_0 can win the game by taking the first loop v_1 times, the second one v_2 times, and finally going to the absorbing state with energy level $(0, 0)$. Indeed, because of the decomposition lemma of [10], the average-energy of the resulting play will be equal to $(0, 0)$.

Zero-tests are correctly executed. Consider counter C_1 . To ensure that P_0 does not cheat by claiming that C_1 has value zero while it is strictly positive, we give the possibility to P_1 to check zero-tests on C_1 by branching using the gadget in Subfig. 5(c). Assume C_1 has value $v_1 > 0$ and P_0 claims it has value zero. Then, P_1 branches and after the $(-1, 0)$ edge, the energy level on the first dimension is still strictly positive. Since P_0 can never decrease it afterwards, he will not meet the threshold $(0, 0)$ on the average-energy and will lose the play. Now, assume C_1 has indeed value $v_1 = 0$. If P_1 decides to branch nonetheless, the energy level after branching is $(0, v_2)$ and P_0 can win by looping v_2 times in the first cycle and then going to the absorbing state. For counter C_2 , we use the symmetric gadget depicted in Subfig. 5(d). Similarly, we can check that P_0 does not cheat by claiming that C_1 (resp. C_2) is strictly positive while it is not. To do so, we give the possibility to P_1 to branch after such a claim and decrement C_1 (resp. C_2) using an edge $(-1, 0)$ (resp. $(0, -1)$) and then go to the gadget in Subfig. 5(a): if C_1 (resp. C_2) had value zero, then the play is losing because the energy level drops below zero, and if C_1 (resp. C_2) was strictly positive, P_0 can win as described before.

Therefore, if P_0 does not faithfully simulate \mathcal{M} , he is guaranteed to lose in G . On the other hand, if P_1 stops a faithful simulation, P_0 is guaranteed to win.

It remains to argue that P_0 wins if, and only if, the machine halts. Indeed, if the machine \mathcal{M} halts, then P_0 simulates its execution faithfully and either he is interrupted and wins, or the simulation ends in the gadget depicted in Subfig. 5(b) and he also wins. Indeed, given that this gadget can only be reached with values of counters equal to zero (by hypothesis on \mathcal{M} and by our assumption on \mathcal{M}), the energy level reaches $(0, 0)$ after the $(-1, -1)$ edge, and using the decomposition lemma mentioned above, the average-energy for the play is $(0, 0)$. Hence, P_0 wins. On the opposite, if \mathcal{M} does not halt, P_0 has no way to reach the halting gadget by means of a faithful simulation, hence if P_0 never cheats and P_1 never branches, the energy level in both dimensions is at all times at least equal to one (since it is equal to $(v_1 + 1, v_2 + 1)$): the average-energy will be at least $(1, 1)$, hence above the threshold, and P_0 loses.

Consequently, we have that P_0 wins if, and only if, the 2CM halts, which implies undecidability for lower-bounded average-energy games with two or more dimensions. \square

8 Conclusion

We have presented the first algorithm for solving average-energy games with only a lower bound, but no upper bound, on the energy level, thereby solving an open problem from [5]. The algorithm is based on the first upper bound on the necessary memory to implement a winning strategy for P_0 in such a game, which solves another open problem from [5].

Our algorithm has a doubly-exponential running time, which we complemented by showing the problem to be EXPSpace-hard. This is an improvement over the previous EXPTIME-hardness result [5]. An obvious open problem concerns closing this gap. On the other hand, a game due to Hunter [16] shows our doubly-exponential upper bound on the memory to be almost tight. Finally, for a unary encoding of the weights and the threshold (or if they are polynomially bounded in the size of the graph), our algorithm runs in exponential time and the upper bound on the memory is exponential as well.

Finally, we also investigated multi-dimensional average-energy games in various settings: we showed such games to be undecidable, both for the case without any bounds on the energy level and for the case of only lower bounds. In contrast, the case of games with both a lower and an upper bound in every dimension is in $\text{NEXPTIME} \cap \text{coNEXPTIME}$ and trivially EXPTIME-hard, another gap to be closed in further research. Also, we left open the decidability of two-dimensional average-energy games without bounds.

Our results were obtained by solving a certain type of mean-payoff (one-counter) pushdown game with unbounded weight function. In further research, we investigate whether our techniques are able to solve arbitrary mean-payoff pushdown games with such a weight function.

References

- [1] Roderick Bloem, Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. Better quality in synthesis through quantitative objectives. In Ahmed Bouajjani and Oded Maler, editors, *CAV 2009*, volume 5643 of *LNCS*, pages 140–156. Springer, 2009.
- [2] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. Markov decision processes and stochastic games with total effective payoff. In Ernst W. Mayr and Nicolas Ollinger, editors, *STACS 2015*, volume 30 of *LIPICs*, pages 103–115. Leibniz-Zentrum für Informatik, 2015.
- [3] Patricia Bouyer, Uli Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jiří Srba. Infinite runs in weighted timed automata with energy constraints. In Franck Cassez and Claude Jard, editors, *FORMATS 2008*, volume 5215 of *LNCS*, pages 33–47. Springer, 2008.
- [4] Patricia Bouyer, Piotr Hofman, Nicolas Markey, Mickael Randour, and Martin Zimmermann. Bounding average-energy games. In Javier Esparza and Andrzej Murawski, editors, *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, Lecture Notes in Computer Science. Springer, 2017. To appear.
- [5] Patricia Bouyer, Nicolas Markey, Mickael Randour, Kim Guldstrand Larsen, and Simon Laursen. Average-energy games. *Acta Informatica*, 2016. Article in Press.
- [6] Luboš Brim, Jakub Chaloupka, Laurent Doyen, Raffaella Gentilini, and Jean-François Raskin. Faster algorithms for mean-payoff games. *Formal Methods in System Design*, 38(2):97–118, 2011.
- [7] Franck Cassez, Jan J. Jensen, Kim G. Larsen, Jean-François Raskin, and Pierre-Alain Reynier. Automatic synthesis of robust and optimal controllers – an industrial case study. In Rupak Majumdar and Paulo Tabuada, editors, *HSCC 2009*, volume 5469 of *LNCS*, pages 90–104. Springer, 2009.
- [8] Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource interfaces. In Rajeev Alur and Insup Lee, editors, *EMSOFT 2003*, volume 2855 of *LNCS*, pages 117–133. Springer, 2003.
- [9] Krishnendu Chatterjee and Laurent Doyen. Energy parity games. *Theoretical Computer Science*, 458:49–60, 2012.
- [10] Krishnendu Chatterjee, Laurent Doyen, Mickael Randour, and Jean-François Raskin. Looking at mean-payoff and total-payoff through windows. *Information and Computation*, 242:25–52, 2015.
- [11] Krishnendu Chatterjee, Mickael Randour, and Jean-François Raskin. Strategy synthesis for multi-dimensional quantitative objectives. *Acta Informatica*, 51(3-4):129–163, 2014.
- [12] Krishnendu Chatterjee and Yaron Velner. Mean-payoff pushdown games. In *LICS 2012*, pages 195–204. IEEE Computer Society, 2012.

- [13] Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, 1979.
- [14] John Fearnley and Marcin Jurdzinski. Reachability in two-clock timed automata is PSPACE-complete. *Information and Computation*, 243:26–36, 2015.
- [15] Wladimir Fridman and Martin Zimmermann. Playing pushdown parity games in a hurry. In Marco Faella and Aniello Murano, editors, *GandALF 2012*, volume 96 of *EPTCS*, pages 183–196, 2012.
- [16] Paul Hunter. Reachability in succinct one-counter games. *arXiv*, 1407.1996, 2014.
- [17] Paul Hunter. Reachability in succinct one-counter games. In Mikolaj Bojanczyk, Slawomir Lasota, and Igor Potapov, editors, *RP 2015*, volume 9328 of *LNCS*, pages 37–49. Springer, 2015.
- [18] Line Juhl, Kim Guldstrand Larsen, and Jean-François Raskin. Optimal bounds for multiweighted and parametrised energy games. In Zhiming Liu, Jim Woodcock, and Yunshan Zhu, editors, *Theories of Programming and Formal Methods – Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday*, volume 8051 of *LNCS*, pages 244–255. Springer, 2013.
- [19] Marcin Jurdziński, Ranko Lazić, and Sylvain Schmitz. Fixed-dimensional energy games are in pseudo-polynomial time. In Magnús M. Halldórsson, Kazuo Iwana, Naoki Kobayashi, and Bettina Speckmann, editors, *ICALP 2015 – Part II*, volume 9135 of *LNCS*, pages 260–272. Springer, 2015.
- [20] Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23(3), 1978.
- [21] Kim G. Larsen, Simon Laursen, and Martin Zimmermann. Limit your consumption! Finding bounds in average-energy games. In Mirco Tribastone and Herbert Wiklicky, editors, *Proceedings 14th International Workshop Quantitative Aspects of Programming Languages and Systems, QAPL’16 2016, Eindhoven, The Netherlands, April 2-3, 2016.*, volume 227 of *EPTCS*, pages 1–14, 2016.
- [22] Marvin L. Minsky. Recursive unsolvability of Post’s problem of “Tag” and other topics in theory of Turing machines. *Annals of Mathematics*, 74(3):437–455, 1961.
- [23] Reino Niskanen, Igor Potapov, and Julien Reichert. Undecidability of two-dimensional robot games. In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *MFCS 2016*, volume 58 of *LIPICs*, pages 73:1–73:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
- [24] Mickael Randour. Automated synthesis of reliable and efficient systems through game theory: A case study. In Thomas Gilbert, Markus Kirkilionis, and Gregoire Nicolis, editors, *ECCS 2012*, Springer Proceedings in Complexity, pages 731–738. Springer, 2013.
- [25] Frank Thuijsman and Otto J. Vrieze. The bad match; a total reward stochastic game. *OR Spektrum*, 9(2):93–99, 1987.
- [26] Yaron Velner, Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, Alexander Rabinovich, and Jean-François Raskin. The complexity of multi-mean-payoff and multi-energy games. *Information and Computation*, 241:177–196, 2015.
- [27] Igor Walukiewicz. Pushdown processes: Games and model-checking. *Information and Computation*, 164(2):234–263, 2001.
- [28] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1-2):343–359, 1996.