Fast value iteration: A uniform approach to efficient algorithms for energy games

Michaël Cadilhac[1], Antonio Casares[2], and Pierre Ohlmann[3]

[1] DePaul University, Chicago, IL, USA
[2] University of Warsaw, Poland
[3] CNRS, Université Aix-Marseille, LIS, France

Abstract. We study algorithms for solving parity, mean-payoff and energy games. We propose a systematic framework, which we call Fast value iteration, for describing, comparing, and proving correctness of such algorithms. The approach is based on potential reductions, as introduced by Gurvich, Karzanov and Khachiyan (1988). This framework allows us to provide simple presentations and correctness proofs of known algorithms, unifying the Optimal strategy improvement algorithm by Schewe (2008) and the quasi dominions approach by Benerecetti et al. (2020), amongst others. The new approach also leads to novel symmetric versions of these algorithms, highly efficient in practice, but for which we are unable to prove termination. We report on empirical evaluation, comparing the different fast value iteration algorithms, and showing that they are competitive even to top parity game solvers.

Keywords: Mean-payoff games · energy games · pseudopolynomial algorithm · value iteration

This document contains hyperlinks. Each occurrence of a notion is linked to its *definition*. On an electronic device, the reader can click on words or symbols (or just hover over them on some PDF readers) to see their definition.

1 Introduction

Mean-payoff and energy games. The games under study are infinite duration games where two players, Min and Max, move a token over a finite directed graph with no sink, where the edges of the graph are labelled by payoffs in $\mathbb{Z}$. When playing a mean-payoff game, the players optimise (minimise or maximise, respectively) the asymptotic average payoff. In an energy game, they instead optimise the supremum cumulative sum of payoffs within $[0, \infty]$. These games are positionaly determined [13,6]: the two players can play optimally even when restricted to strategies that only depend on the current position of the game. We refer to Figure 1 for a complete example.

In this paper, we are interested in the problem of computing energy values of the vertices in a given game which we call *solving* the energy game. It easily follows from positional determinacy that the energy value of a vertex is finite if and only if its mean-payoff value is non-positive [7]. Therefore solving an

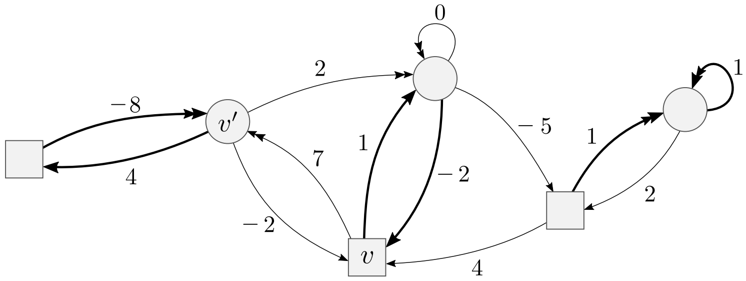


Fig. 1. Example of a game; circles belong to Min and squares belong to Max. From left to right, the mean-payoff values are $-2, -2, -\frac{1}{2}, -\frac{1}{2}, 1$ and 1, and positional strategies for mean-payoff values are identified in bold. Energy values are $0, 2, 9, 0, \infty$ and ∞, and with optimal strategies given by the double-headed arrows.

energy game also solves the so called threshold problem for the associated mean-payoff game. As it turns out, all state-of-the-art algorithms [3,5,7,12,31,33] for the mean-payoff threshold problem actually solve the energy game.

Mean-payoff values achieved by positional strategies can be computed in polynomial time, and therefore the threshold problem belongs to NP ∩ coNP. Despite numerous efforts, no polynomial algorithm is known. Mean-payoff games are known [32] to generalise parity games [14,30] which also belong to NP∩coNP but for which algorithms with quasipolynomial runtime were recently devised [8]. However, quasipolynomial algorithms for parity games do not generalise to mean-payoff games [16].

Algorithmic paradigms. There are two well-established paradigms for solving energy games: value iteration (sometimes called "progress measure") and strategy improvement. The standard value iteration for energy games (which we will call *Simple value iteration*, SVI for short) was introduced by Brim et al. [7]. While subject to good theoretical (pseudopolynomial) bounds, it is well-known to be prohibitively slow in practice, as its worse-case behaviour is frequently displayed. On the other hand, strategy improvement algorithms [4] typically solve practical instances in a constant number of iterations. Although it offers a useful categorization of older algorithms, the value iteration versus strategy improvement dichotomy fails to accurately describe a new wave of efficient algorithms.

In recent years, multiple hybrid algorithms – borrowing ideas from both paradigms – have been put forward. In 2008, Schewe [33] introduced an algorithm called *Optimal strategy improvent* (OSI) for solving parity or mean-payoff games. As explained by Luttenberger [26], Schewe's presentation of OSI is in fact closer to value iteration, but it can also be formally cast as a strategy improvement in a carefully generalised framework allowing for nondeterministic strategies. In 2019, Dorfman et al. [12] presented a value iteration method augmented by a carefully crafted acceleration mechanism (which we call DKZ), thereby improving on the best theoretical guarantees (this algorithm can be seen as a reformulation of the

GKK algorithm [19], see also [31] for further analyses). Based on the idea of quasi dominions (similar to Fearnley's snares [15] in a strategy improvement context), Benerecetti et al. [2,3] proposed another such acceleration mechanism, obtaining the algorithm QDPM. Some of these algorithms are extremely efficient: a version of OSI is a key component in the LTL-synthesis tool STRIX [29,27], which is one of the top competitors in the annual synthesis competition SYNTCOMP [20]. On the other hand, QDPM is currently the top-performing mean-payoff game solver [3] while, remarkably, preserving state-of-the-art theoretical guarantees.

Although differences in the performances of these algorithms have been observed empirically [3], we lack a good understanding of how they compare to each other theoretically, and more generally, of what are the fundamental algorithmic mechanisms that lead to efficient game solvers in practice.

Contributions. Our contributions are as follows.

(1) Fast value iteration framework. We consider potential reductions, as introduced by Gurvich, Karzanov and Khachiyan [19], to design a systematic method for producing algorithms for energy games, which we call the *fast value iteration framework*. A potential is a mapping which assigns a positive weight to each vertex. Such a potential naturally induces a transformation (a potential reduction) of the game, which preserves the weight of every cycle and thus the values in the mean-payoff game. The fast value iteration meta-algorithm (Algorithm 1) simply iterates on potential reductions until a fixpoint is reached. This meta-algorithm can be instantiated on any given class of potentials, leading to different algorithms, whose correctness is automatically guaranteed under mild assumptions on the potentials (Theorem 2). Interestingly, the framework also provides a *symmetric* meta-algorithm, for which termination is observed in practice, but we have not been able to prove it theoretically.

(2) Unifying and simplifying existing algorithms. We revisit various algorithms in the light of the above framework. Naturally, the classic SVI [7] is captured (Example 1), as well as the algorithms GKK [19] and DKZ [12] (Section 4.3), whose original presentations fit the potential reduction framework.

More interestingly, we also capture algorithms showcasing an excellent performance in practice, defying the common belief that VI algorithms are slow. We unify and simplify the algorithms OSI by Schewe [33] and the involved QDPM by Benerecetti et al. [2,3]. Our presentations are streamlined (see Section 4 for details), leading to immediate correctness proofs. It also allows to isolate the core algorithmic idea underlying these two algorithms, which is a natural adaptation of Dijkstra's algorithm to the two-player setting. We call the obtained reinterpretation of OSI and QDPM within the fast value iteration framework, the Positive Path Iteration (PPI).

The abstraction provided by our approach sets the stage to easily craft new algorithms. Showcasing its applicability, we propose a dynamic version of PPI (DPPI), which provably breaks the theoretical barrier set by OSI and QDPM (Theorem **??**). Many possibilities for future work are proposed in the conclusion.

(3) Empirical evaluation. We compare the implementations of the algorithms described in the fast value iteration framework to OSI and QDPM, as well as to the top parity game solvers. This evaluation shows: (i) fast value iteration algorithms are highly efficient in practice, and especially robust towards hard instances; (ii) alternating versions of the algorithms not only terminate, but are remarkably efficient.

2 Preliminaries

A *game* is a tuple $\mathcal{G} = (G, w, V_{\text{Min}}, V_{\text{Max}})$, where $G = (V, E)$ is a finite sinkless directed graph, $w : E \to \mathbb{Z}$ is a labelling of its edges by integer weights, and $V_{\text{Min}}, V_{\text{Max}}$ is a partition of V. We set $n = |V|, m = |E|$ and $W = \max_{e \in E} |w(e)|$. We say that vertices in V_{Min} belong to Min and that those in V_{Max} belong to Max. We now fix a game $\mathcal{G} = (G, w, V_{\text{Min}}, V_{\text{Max}})$.

We simply write vv' for an edge $(v, v') \in E$. A path is a (possibly empty, possibly infinite) sequence of edges $\pi = e_0 e_1 \dots$, with $e_i = v_i v_i'$, such that $v_i' = v_{i+1}$. We write $v_0 \to v_1 \to \dots$ to denote such a path. The sum of a finite path π is the sum of the weights appearing on it, we denote it by $\text{sum}(\pi)$. Given a finite or infinite path $\pi = e_0 e_1 \dots$ and an integer $k \geq 0$, we let $\pi_{<k} = e_0 e_1 \dots e_{k-1}$, and we let $\pi_{\leq k} = \pi_{<k+1}$. Note that $\pi_{<0}$ is the empty path, and that $\pi_{<k}$ has length k. By convention, the empty path starts and ends in all vertices.

A *valuation* is a map $\text{val} : \mathbb{Z}^\omega \to \mathbb{R} \cup \{\infty\}$ assigning a potentially infinite value to infinite sequences of weights. We use $\mathbb{R}^\infty, \mathbb{Z}^\infty$ and $\mathbb{N}^\infty$ to denote respectively $\mathbb{R} \cup \{\infty\}, \mathbb{Z} \cup \{\infty\}$ and $\mathbb{N} \cup \{\infty\}$. The four valuations studied in this paper are the mean-payoff, energy, positive-energy, and first-if-positive valuations given by:

$$\begin{aligned} \text{MP}(w) &= \limsup_k \tfrac{1}{k} \textstyle\sum_{i=0}^{k-1} w_i \in \mathbb{R}, & \text{En}^+(w) &= \textstyle\sum_{i=0}^{k_{\text{neg}}-1} w_i \in \mathbb{N}^\infty, \\ \text{En}(w) &= \sup_k \textstyle\sum_{i=0}^{k-1} w_i \quad \in \mathbb{N}^\infty, & \text{First}^+(w) &= \max(w_0, 0) \in \mathbb{N}, \end{aligned}$$

where $w = w_0 w_1 \dots$ is a sequence of weights and $k_{\text{neg}} = \min\{k \mid w_k < 0\} \in \mathbb{N}^\infty$ is the first index of a negative weight. For technical convenience, we will also consider games in which weights are potentially (positively) infinite. We extend the definitions of En, En^+ and First^+ to words in $(\mathbb{Z}^\infty)^\omega$, using the same formula. Note that for any $w \in (\mathbb{Z}^\infty)^\omega$ we have $\text{En}^+ \leq \text{En}$. The four valuations are illustrated on a given sequence of weights in Figure 2.

A *strategy* for Min is a map $\sigma : V_{\text{Min}} \to E$ such that for all $v \in V_{\text{Min}}$, it holds that $\sigma(v)$ is an edge outgoing from v. We say that a (finite or infinite) path $\pi = e_0 e_1 \dots$ is consistent with σ if whenever $e_i = v_i v_{i+1}$ is defined and $v_i \in V_{\text{Min}}$, it holds that $e_i = \sigma(v_i)$. We write in this case $\pi \models \sigma$. Strategies for Max are defined similarly and written $\tau : V_{\text{Max}} \to E$. The theorem below states that the three valuations are determined with positional strategies over finite games. It is well known for MP and En and easy to prove for En^+. We remark that positional determinacy also holds for the two energy valuations En and En^+ over games where we allow for infinite weights.

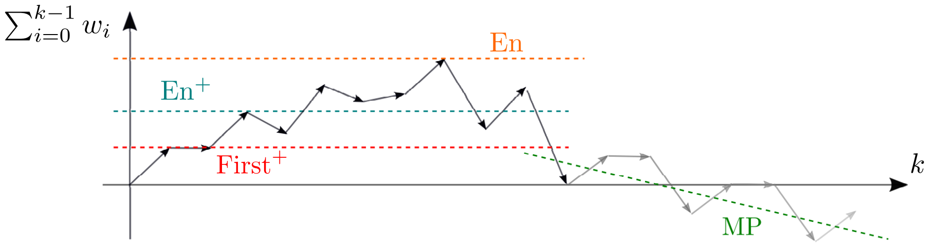


Fig. 2. The three valuations over a given sequence of weights. The mean-payoff value is given by the slope of the line, which corresponds to the long-term average. In this case, the mean-payoff is ≤ 0, and both En and En$^+$ are finite.

Theorem 1 ([13,6]). *For each* val $\in \{\mathrm{MP}, \mathrm{En}, \mathrm{En}^+\}$, *there exist strategies* σ_0 *for Min and* τ_0 *for Max such that for all* $v \in V$ *we have*

$$\sup_{\pi \models \sigma_0} \mathrm{val}(w(\pi)) = \inf_{\sigma} \sup_{\pi \models \sigma} \mathrm{val}(w(\pi)) = \sup_{\tau} \inf_{\pi \models \tau} \mathrm{val}(w(\pi)) = \inf_{\pi \models \tau_0} \mathrm{val}(w(\pi)),$$

where σ, τ *and* π *respectively range over strategies for Min, strategies for Max, and infinite paths from* v.

The quantity defined by the equilibrium above is called the *value* of v in the val game, and we denote it by $\mathrm{val}_{\mathcal{G}}(v) \in \mathbb{R}^\infty$; the strategies σ_0 and τ_0 are called val-optimal, note that they do not depend on v. The following result relates the values in the mean-payoff and energy games; this direct consequence of Theorem 1 was first stated in [7].

Corollary 1 ([7]). *For all* $v \in V$ *it holds that*

$$\mathrm{MP}_{\mathcal{G}}(v) \leq 0 \iff \mathrm{En}_{\mathcal{G}}(v) < \infty \iff \mathrm{En}_{\mathcal{G}}(v) \leq (n-1)W.$$

Therefore computing En-values of the games is harder than the threshold problem. It is easy deduce En-optimal strategies for Min (which always achieve MP-value ≤ 0) from the knowledge of the En-values; however no knowledge is gained about Max strategies besides the winning region (over which En values are ∞). As explained in the introduction, all state-of-the-art algorithms for the threshold problem actually compute En values. This shifts our focus from mean-payoff to energy games.

Attractors. Given a subset $S \subseteq V$, the attractor $\mathrm{Attr}_{\mathcal{G}}^{\mathrm{Max}}(S)$ to S in $\mathcal{G}$ is defined to be the set of vertices v such that Max can ensure to reach S from v.

Simple games. A finite path $v_0 \to v_1 \to \ldots \to v_k$ is simple if there is no repetition in $v_0, v_1, \ldots, v_{k-1}$; note that a cycle may be simple. A game is *simple* if all simple cycles have nonzero sum. The following result is folklore and states that one may reduce to a simple game at the cost of a linear blow up on W. It holds thanks to the fact that positive mean-payoff values are $\geq 1/n$, which is a well-known consequence of Theorem 1.

Lemma 1. *Let $\mathcal{G} = (G, w, V_{\text{Min}}, V_{\text{Max}})$ be an arbitrary game. The game $\mathcal{G}' = (G, (n+1)w - 1, V_{\text{Min}}, V_{\text{Max}})$ is simple and has the same vertices of positive mean-payoff values as $\mathcal{G}$.*

3 Fast value iteration: A meta-algorithm based on potential reductions

3.1 Potential reductions

Fix a game $\mathcal{G} = (G = (V, E), w, V_{\text{Min}}, V_{\text{Max}})$. A *potential* is a map $\varphi : V \to \mathbb{N}^\infty$. Potentials are partially ordered coordinatewise. We write $\varphi = 0$ if $\varphi(v) = 0$ for all $v \in V$. Given an edge $vv' \in E$, we define its φ-modified weight to be

$$w_\varphi(vv') = \begin{cases} \infty & \text{if } \varphi(v), \varphi(v') \text{ or } w(vv') \text{ is } \infty, \\ w(vv') + \varphi(v') - \varphi(v) & \text{otherwise.} \end{cases}$$

The *φ-modified game $\mathcal{G}_\varphi$* is simply the game $(G, w_\varphi, V_{\text{Min}}, V_{\text{Max}})$; informally, all weights are replaced by the modified weights. Note that the underlying graph does not change, in particular paths in $\mathcal{G}$ and $\mathcal{G}_\varphi$ are the same. Moving from $\mathcal{G}$ to $\mathcal{G}_\varphi$ for a given potential φ is called a *potential reduction*.

Weights of cycles are preserved by finite potential reductions, and therefore, as an easy consequence of positionality (Theorem 1), mean-payoff values are preserved. Note that any edge outgoing from a vertex v with potential $\varphi(v) = \infty$ has weight ∞ in the modified game, therefore v has En and En^+-values ∞ in G_φ. Note also that sequential applications of potential reductions correspond to reducing with respect to the sum of the potentials: $(\mathcal{G}_\varphi)_{\varphi'} = \mathcal{G}_{\varphi+\varphi'}$.

Potential reductions were introduced by Gallai [18] for studying network related problems such as shortest-paths problems. In the context of mean-payoff or energy games, they were introduced in [19] and later sometimes rediscovered.

3.2 The fast value iteration meta-algorithm

A *potential assigner* is a function Ψ that assigns a potential $\Psi(\mathcal{G}) : V \to \mathbb{N}^\infty$ to each game $\mathcal{G}$. A potential assigner Ψ induces a *fast value iteration algorithm* (called Ψ-FVI) as follows: successively apply potential reductions using the potentials given by Ψ, until a game $\mathcal{G}'$ is reached with $\Psi(\mathcal{G}')(V) \subseteq \{0, \infty\}$. For an arbitrary potential assigner, this algorithm might not terminate, or provide a final game $\mathcal{G}'$ carrying irrelevant information. However, we show that under mild hypotheses on Ψ, this algorithm terminates, and $\text{En}_{\mathcal{G}'} = \Psi(\mathcal{G}')$, with the vertices with En-value 0 corresponding to the vertices with finite value in the original game. Moreover, the exact En-values of the original game can be recovered from the sequence of potentials obtained during the computation.

We formalise this idea in Algorithm 1 and Theorem 2. To ensure termination, we need to artificially increase the potential of some vertices to ∞ whenever a threshold is reached, and then remove Max's attractor to ∞. This technique is standard in value iteration algorithms, see e.g. [7].

Algorithm 1 *Ψ-Fast value iteration algorithm.*

```
**Input:** Game $\mathcal{G}$ with $n$ vertices and maximal weight $W$
$\Phi \leftarrow 0$                                      ▷ $\Phi$ carries the cumulative sum of potentials
**do**
    $\varphi \leftarrow \Psi(\mathcal{G})$
    $\Phi \leftarrow \Phi + \varphi$                      ▷ Update cumulative sum over $\mathcal{G}$[4]
    $\mathcal{G} \leftarrow \mathcal{G}_\varphi$
    $A \leftarrow \mathrm{Attr}^{\mathrm{Max}}_{\mathcal{G}}(\Phi^{-1}([(n-1)W+1,\infty]))$
    Set $\Phi(v) = \infty$ for all $v \in A$
    $\mathcal{G} \leftarrow \mathcal{G} \setminus A$
**while** $\varphi \neq 0$ and $\mathcal{G} \neq \emptyset$
**return** $\Phi$
```

Let us isolate two relevant properties of potential assigners: (1) *Soundness*: for any game $\mathcal{G}$, $\Psi(\mathcal{G}) \leq \mathrm{En}_{\mathcal{G}}$; (2) *Completeness*: for any $\mathcal{G}$, if $\Psi(\mathcal{G}) = 0$ then $\mathrm{En}_{\mathcal{G}} = 0$. We also say that a potential φ is sound over a given game if condition (1) is met. We may now state our first main result.

Theorem 2. *Let Ψ be a sound and complete potential assigner. Then Algorithm 1 terminates in at most n^2W iterations, and returns $\Phi = \mathrm{En}_{\mathcal{G}}$.*

Remark 1. Note that the hypotheses of the theorem are minimal. If a potential assigner Ψ is not sound, there is a game $\mathcal{G}$ for which the algorithm returns $\Phi \geq \Psi(\mathcal{G}) > \mathrm{En}_{\mathcal{G}}$. If it does not satisfy (ii), there is a game for which the algorithm stops in the first iteration, returning the potential $\Phi = 0 \neq \mathrm{En}_{\mathcal{G}}$.

Example 1 (Simple value iteration of Brim et al. [7]). Define the potential assigner Ψ_{First^+} by assigning the potential $\mathrm{First}^+_{\mathcal{G}}(v)$, the first-if-positive value, to a vertex v. This potential is easily computed in linear time as it coincides for each Max (resp. Min) vertex v, with the maximal (resp. minimal) value of $\max(w, 0)$ where w ranges over outgoing weights. Clearly $\Psi_{\mathrm{First}^+} \leq \mathrm{En}_{\mathcal{G}}$, since for any sequence of weights $w_0w_1\dots$, it holds that $\mathrm{First}^+(w_0w_1\dots) \leq \mathrm{En}(w_0w_1\dots)$. Finally, if $\Psi_{\mathrm{First}^+} = 0$, then from any vertex Min can ensure that no positive weight is ever seen, which entails $\mathrm{En}_{\mathcal{G}} = 0$. We conclude that Ψ_{First^+} is sound; the fast value iteration algorithm coincides with that of [7][5].

Example 2. Any (determined) valuation $\mathrm{val}\colon \mathbb{Z}^\omega \to \mathbb{R}^\infty$ induces a potential assigner Ψ_{val}, namely, the one that assigns to each game $\mathcal{G}$ the potential given by $\mathrm{val}_{\mathcal{G}}(v)$. If the valuation satisfies $\mathrm{val} \leq \mathrm{En}$ over weight sequences, then Ψ_{val} is sound. Moreover, if $\mathrm{val}(w_0w_1 ...) > 0$ whenever $w_0 > 0$, then Ψ_{val} is complete. This includes the previous example, and more interestingly, this includes the valuation En^+, which is the object of Section 4.1.

[4] By a small abuse of notation, we allow to sum potential with different domains. If $\varphi\colon V \to \mathbb{N}^\infty$ and $\varphi'\colon V' \to \mathbb{N}^\infty$ with $V' \subseteq V$, then $\varphi + \varphi'(v) = \varphi(v)$ for all $v \notin V'$.

[5] Formally, reducing from complexity $O(n^2mW)$ to $O(nmW)$ requires some additional bookkeeping.

Of course, an important requirement over Ψ to make Algorithm 1 relevant is that we should be able to compute $\Psi(\mathcal{G})$ efficiently. Note that the potential assigner corresponding to the En-values of a game satisfies all the required hypothesis, and makes Algorithm 1 terminate in a single iteration.

∞-attraction. In many occurrences, the algorithm can be simplified by removing lines 6-8 and stopping when a fixpoint is reached (which can be implemented by replacing line 9 with "while $\mathcal{G}_{\Psi(\mathcal{G})} \neq \mathcal{G}$"). We say that potential assigners with this property are *∞-attracting*.

Modularity. As noted in the introduction (a version of) OSI is the algorithm used by STRIX [29,27], the top LTL-synthesis tool in SYNTCOMP [20]. However, it is well-known that other parity game solvers are faster over most instances (c.f. Section 5). Then, why OSI? The main reason is that it allows for modularity (it was ported to the GPU by Meyer and Luttenberger [28]). Games coming from LTL-formulas are typically huge, one key feature of STRIX is the ability to solve games modularly, avoiding loading the entire games into memory. The framework of fast value iteration is specially well suited for a modular approach, which also opens exciting perspectives for parallelised implementations.

We note that, since $\mathcal{G}$ and $\mathcal{G}_\varphi$ have the same positive cycles, for all finite φ, we can safely apply any potential reduction to a subgame without modifying the winning region of the players in the global game. To moreover preserve the En-values, or allow for potentials taking the value ∞, we need a further requirement. Let $\mathcal{S}$ be a class of *subgames*[6]. We say that a potential assigner Ψ is *$\mathcal{S}$-sound* if, for all games $\mathcal{G}$ and subgames $\mathcal{G}' \subseteq \mathcal{G}$ such that $(\mathcal{G}', \mathcal{G}) \in \mathcal{S}$, it holds that $\Psi(\mathcal{G}') \leq \mathrm{En}_{\mathcal{G}}|_{\mathcal{G}'}$.[7] Therefore, if Ψ is $\mathcal{S}$-sound, we can identify subgames of $\mathcal{G}$ in $\mathcal{S}$, solve them partially, and apply the corresponding potential reduction in the whole game $\mathcal{G}$.

The rest of the section is devoted to a proof of Theorem 2.

Termination. Termination of Algorithm 1 is ensured thanks to lines 6 and 7: the function Φ strictly increases in each non-terminating iteration, and it only takes values in $[0, nW] \cup \{\infty\}$, hence the bound n^2W.

Correctness. We now state the key technical theorem enabling our framework. It describes the effect of sound potential reductions over energy values, allowing to combine them. From it, we easily derive compositionality of sound potentials.

Theorem 3 (Update of energy values). *If φ is sound then $\mathrm{En}_{\mathcal{G}} = \varphi + \mathrm{En}_{\mathcal{G}_\varphi}$.*

Corollary 2 (Compositionality). *If φ is sound for $\mathcal{G}$ and φ' is sound for $\mathcal{G}_\varphi$ then $\varphi + \varphi'$ is sound for $\mathcal{G} + \mathcal{G}'$.*

We are now ready to prove Theorem 2.

[6] Formally, a class of pairs $(\mathcal{G}', \mathcal{G})$, with $\mathcal{G}' \subseteq \mathcal{G}$.

[7] Any sound potential is $\mathcal{S}_{\mathrm{Trap}}$-sound for the class of subgames $(\mathcal{G}', \mathcal{G})$ such that $\mathcal{G}'$ is a Min-trap in $\mathcal{G}$ (as $\mathrm{En}_{\mathcal{G}'} \leq \mathrm{En}_{\mathcal{G}}|_{\mathcal{G}'}$ in that case).

Proof (Informal proof for Theorem 2). Let $\mathcal{G}_i$, $\varphi_i = \Psi(\mathcal{G}_i)$ and $\Phi_i = \varphi_1 + \cdots + \varphi_{i-1}$ denote the game, potential and cumulative sum at the i-th iteration of the algorithm. Since Ψ is sound, φ_i is sound for $\mathcal{G}_i$ for all i. Thus it follows from an easy induction and compositionality that for all i, Φ_i is sound for $\mathcal{G}$. In particular, for the maximal i, Theorem 3 gives $\Phi_i + \mathrm{En}_{\mathcal{G}_i} = \mathrm{En}_{\mathcal{G}}$, but moreover since $\varphi_i = 0$ we get by completeness that $\mathrm{En}_{\mathcal{G}_i} = 0$ which concludes. □

3.3 Asymmetry and alternating fast value iteration

Fast value iteration are based on successive underapproximations of the energy valuation En, which is inherently asymmetric. However, the initial problem (solving mean-payoff games) is itself symmetric, which calls for the design of more symmetrical solutions, a recurring theme in the literature [21,22,34,36].

Dual algorithm computing Max*-values.* Let $\overline{\mathcal{G}}$ be the game obtained by swapping V_{Min} and V_{Max} and relabelling the weights by $\overline{w} = -w$. The two games are essentially equivalent, for instance the mean-payoff values in $\overline{\mathcal{G}}$ and $\mathcal{G}$ are opposite. However asymmetric algorithms such as value iterations behave differently over each game; this is useful for instance if one wants to compute Max strategies in $\mathcal{G}$, which are output by running value iterations in the dual. But this still does not provide a symmetric solution.

Alternating fast value iteration. We now consider *alternating versions* of the algorithm, by working with potentials in $\varphi : V \to \mathbb{Z} \cup \{\pm\infty\}$. The algorithm applies potential reductions corresponding to Ψ and its dualized version $\overline{\Psi}$ *on the same game* in an alternating fashion, until all vertices are sent to $+\infty$ or $-\infty$. Naturally, when a vertex is set to $+\infty$ or $-\infty$, the adequate attractor is computed and removed from the game.

Assuming the potential assigner Ψ is sound, since sound potential reductions do not alter winning regions, the algorithm is correct and Min's winning region is the preimage of $-\infty$ by the final potential. Termination, however, is not easily guaranteed. Interestingly, we observe experimentally that, for some potential assigners, this alternating algorithm always terminates, and it is even remarkably fast (see Section 5). We leave as an interesting open problem to determine for which potential assigners (if any) this algorithm terminates (see conclusion).

4 Instances of fast value iteration

We have already shown (Example 1) how SVI instantiates in our framework. In this section, we introduce further potential assigners to capture known efficient algorithms for energy games, and prove their soundness. This provides a streamlined and unified presentation of (versions of) the algorithms OSI [33] and QDPM [2,3] (Section 4.1), namely the positive path iteration algorithm (PPI). We also propose a dynamic variant DPPI, corresponding to a potential assigner generating potentials with provably larger values. At the end of the section we also discuss the GKK algorithm [19], and then provide formal comparisons between the four algorithms stated in our framework.

In all cases, we find that the algorithms are easier to explain over simple games, which we will assume without loss of generality (see Lemma 1); note also that simplicity is preserved by potential reductions.

4.1 The positive path iteration algorithm

We now study the fast value iteration algorithm corresponding to the potential assigner $\mathit{\Psi}_{\mathrm{En}^+}(\mathcal{G}) = \mathrm{En}^+_{\mathcal{G}}$. We call it the *Positive path iteration algorithm (PPI)*. It is immediate to check that the potential assigner $\mathit{\Psi}_{\mathrm{En}^+}$ is sound and complete (see Example 2), so Theorem 2 applies, directly giving correctness of PPI. Moreover, we can in this case simplify the algorithm by removing lines 6-7 in Algorithm 1, because $\mathit{\Psi}_{\mathrm{En}^+}$ is ∞-attracting.

Proposition 1. *The potential assigner $\mathit{\Psi}_{\mathrm{En}^+}$ is ∞-attracting.*

We let $N_{\mathcal{G}}$ denote the set of vertices from which Min can ensure to immediately see a negative vertex: $v \in V_{\mathrm{Max}}$ (resp. V_{Min}) belongs to N if and only if all outgoing edges (resp. some outgoing edge) have weight < 0. Note that computing En^+-values in $\mathcal{G}$ corresponds to solving a variant of the energy game which stops whenever N is reached. It turns out that this problem is (efficiently) tractable, thanks to two-player game extensions of Dijkstra's algorithm. In fact, two seemingly distinct algorithms are known, corresponding to OSI [33] and QPDM [2]. Remarkably, Khachiyan, Gurvich and Zhao [23] solved the same problem[8] earlier and in a different context (with an algorithm similar to Schewe's).

Two algorithms for computing En^+*.* We now describe the two algorithms, respectively extracted from [33] and [2]. We first introduce some notation. For a subset $F \subseteq V$, a vertex $v \in F$ and an edge vv', we define $\mathsf{esc}_F(vv') = w(vv')$ if $v' \notin F$, and $\mathsf{esc}_F(vv') = \infty$, if $v' \in F$. We define the escape value of a vertex as:

$$\mathsf{esc}_F(v) = \begin{cases} \min\{\mathsf{esc}_F(vv') \mid w(vv') \geq 0\}, & \text{if } v \in V_{\mathrm{Min}}, \\ \max\{\mathsf{esc}_F(vv') \mid w(vv') \geq 0\}, & \text{if } v \in V_{\mathrm{Max}}. \end{cases}$$

We will only consider subsets $F \subseteq {N_{\mathcal{G}}}^{\mathsf{c}}$, so Max vertices have a non-negative outgoing edge and Min vertices have only non-negative outgoing edges, in particular $\mathsf{esc}_F(v) \geq 0$. It can be seen as the minimal weight that Min can force to see while leaving F immediately from v, or ∞ is she cannot force to leave F in one step, assuming Max is constrained to playing non-negative edges. We further let $F^{<\infty}$ denote the set of vertices with finite esc_F, and $F^{<\infty}_{\mathrm{Max}}$ and $F^{<\infty}_{\mathrm{Min}}$ their intersections with V_{Max} and V_{Min}. Last, for $v \in V$, the notation $\varphi_v(v) \leftarrow x$ indicates that φ_v is the potential defined by $\varphi_v(v) = x$ and $\varphi_v(v') = 0$ for $v' \neq v$.

[8] This corresponds to Theorem 1 in [23], case (i) with blocking systems $\mathcal{B}_2$.

Algorithm 2 Subprocedure in OSI

```
Input: Simple game $\mathcal{G}$
$\Phi \leftarrow 0$
$F \leftarrow V \setminus N_{\mathcal{G}}$
while $F^{<\infty} \neq \emptyset$ do
    if $F^{<\infty}_{\mathrm{Max}} \neq \emptyset$ then
        let $v \in F^{<\infty}_{\mathrm{Max}}$
        $\Phi(v) \leftarrow \max_{w(vv') \geq 0} w(vv') + \Phi(v')$
    else if $F^{<\infty}_{\mathrm{Min}} \neq \emptyset$ then
        let $v \in F^{<\infty}_{\mathrm{Min}}$ minimizing
                    $m = \min_{v' \in F} w(vv') + \Phi(v')$
        $\Phi(v) \leftarrow m$
    end if
end while
$\Phi_v(v) \leftarrow \infty$ for all $v \in F$
return $\Phi$
```

Algorithm 3 Subprocedure in QDPM

```
Input: Simple game $\mathcal{G}$
$\Phi = 0$
$F \leftarrow V \setminus N_{\mathcal{G}}$
while $F^{<\infty} \neq \emptyset$ do
    let $v \in \mathrm{argmin}_F \, \mathsf{esc}_F(v)$
    $\varphi_v(v) \leftarrow \mathsf{esc}_F(v)$
    $F \leftarrow F \setminus \{v\}$
    $\mathcal{G} \leftarrow \mathcal{G}_{\varphi_v}$
    $\Phi = \Phi + \varphi_v$
end while
$\Phi_v(v) \leftarrow \infty$ for all $v \in F$
return $\Phi$
```

Theorem 4 (Adapted from [23,33,2]). *Algorithms 2 and 3 both compute $\mathrm{En}^+_{\mathcal{G}}$, and both can be implemented to run in $O(m + n \log n)$ operations.*

4.2 A new fast value iteration algorithm

Drawing inspiration from Algorithms 2 and 3 above, we introduce another potential assigner, leading to a fast value iteration algorithm which we call *Dynamic positive path iteration (DPPI)*. Note that both algorithms above compute the Min attractor to $N_{\mathcal{G}}$ over non-negative edges, which corresponds exactly to the set of vertices with finite En^+, and obtain the values of En^+ by backtracking. We will also backtrack over the same attractor, and just as in Algorithm 3, we make potential updates on the fly. The difference is in the precise way in which we choose the vertices, which enables in our case that some of the potential updates may cause new edges to become positive, which will then be taken into account, sometimes leading to a potential $> \mathrm{En}^+$.

Lemma 2. *The potential assigner Ψ_{DPPI} is sound and complete for simple games.*

We see DPPI as a marginal improvement over PPI, but an improvement nonetheless, showing that the barrier imposed by PPI can be broken and motivating future work. Figure 3 provides a small game where DPPI performs fewer iterations than PPI, while Theorem **??** below proves that for any game $\mathcal{G}$, $\Psi_{\mathsf{DPPI}}(\mathcal{G}) \geq \Psi_{\mathrm{En}^+}(\mathcal{G})$ (see also the discussion above Theorem **??**).

Algorithm 4 Computation of the Ψ_{DPPI}-potential.

```
Input: Simple game $\mathcal{G}$
$F \leftarrow V \setminus N_{\mathcal{G}}$
while $F^{<\infty} \neq \emptyset$ do
    if there is $v \in (\operatorname{argmin} \mathsf{esc}_F(v)) \cap V_{\mathrm{Min}}$ then $\varphi_v(v) \leftarrow \mathsf{esc}_F(v)$
    else let $v \in \operatorname{argmax}_{v \in F^{<\infty}_{\mathrm{Max}}} \mathsf{esc}_F(v)$ and $\varphi_v(v) \leftarrow \mathsf{esc}_F(v)$
    end if
    $F \leftarrow F \setminus \{v\}$
    $\mathcal{G} \leftarrow \mathcal{G}_{\varphi_v}$
end while
$\varphi_v(v) \leftarrow \infty$ for all $v \in F$
return $\sum_{v \in V} \varphi_v$
```

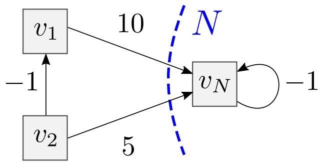


Fig. 3. A game $\mathcal{G}$ (all vertices belong to Max) where DPPI performs a single iteration, as $\Psi_{\mathsf{DPPI}}(\mathcal{G}) = [v_1 \to 10; v_2 \to 9; v_N \to 0] = \mathrm{En}_{\mathcal{G}}$. In contrast, PPI requires two iterations since $\mathrm{En}^+ = [v_1 \to 10; v_2 \to 5; v_N \to 0]$.

4.3 The GKK algorithm

We include a short discussion about the GKK algorithm; a more detailed modern exposition, including state-of-the-art upper bounds and comparison with the related approach of Dorman et al. [12], was proposed by Ohlmann [31].

The *GKK algorithm* is the Ψ_{GKK}-fast value iteration where Ψ_{GKK} is the potential assigner defined as follows.

Let V_- be the set of vertices from which Min can ensure that a negative edge is seen before the first positive edge. (Note that V_- coincides with $(\mathrm{En}^+_{\mathcal{G}})^{-1}(0)$.) Likewise, let V_+ denote the set of vertices from which Max can ensure seeing a positive edge before a negative one; and observe that in a simple game, V_+ is the complement of V_-. Consider the maximal value w_+ such that from any vertex of V_+ Max can ensure to add up to w_+ before a negative weight is seen (alternatively, w_+ is the smallest nonzero value of $\mathrm{En}^+_{\mathcal{G}}$) ; and dually for w_-. Note that, if from any vertex in V_+, Max can ensure to remain in V_+ while seeing positive vertices, then $w_+ = \infty$. Clearly $w_+ \leq \mathrm{En}^+ \leq \mathrm{En}$ over V_+. We define $\Psi_{\mathsf{GKK}}(\mathcal{G})(v)$ to be $\min(w_+, -w_-)$ if $v \in P$ and 0 otherwise. Soundness follows from the inequality above, and completeness is easy to prove. Moreover, Ψ_{GKK} is ∞-attracting.

The potential Ψ_{GKK} has a remarkable symmetric property: the assigned potentials are the same over $\mathcal{G}$ and over its dual $\overline{\mathcal{G}}$: $\Psi_{\mathsf{GKK}} = \overline{\Psi_{\mathsf{GKK}}}$.[9] In particular, the algorithm and its alternating version coincide.

[9] This was first observed by Ohlmann [31] leading to an improved upper bound.

4.4 Comparing fast value iteration algorithms

We now propose formal comparisons between the above potential assigners. Intuitively, in order to minimise the number of iterations of a fast value iteration algorithm, we should seek for potentials assigning large values to vertices, so that a "big step" is produced in each iteration of the algorithm. In this sense, if $\Psi \leq \Psi'$, the Ψ'-FVI algorithm is expected to perform better. A priori, the sequence of games produced by the two algorithms will diverge, impeding formal comparisons on the number of iterations. However, for *monotone* potential assigners, we can also compare the number of iterations of the induced FVI algorithms.

Lemma 3. *For every game $\mathcal{G}$,*

$$\Psi_{\mathrm{First}^+}(\mathcal{G}) \leq \Psi_{\mathrm{En}^+}(\mathcal{G}) \leq \Psi_{\mathsf{DPPI}}(\mathcal{G}), \text{ and } \Psi_{\mathsf{GKK}}(\mathcal{G}) \leq \Psi_{\mathrm{En}^+}(\mathcal{G})$$

Moreover, there are games making these inequalities strict. The potential assigners Ψ_{First^+} and Ψ_{GKK} are incomparable.

Let Ψ, Ψ' be two potential assigners. We say that Ψ' is *monotonically larger* than Ψ', noted $\Psi \leq_{\mathrm{mon}} \Psi'$ if, for all game and potentials $\varphi \leq \varphi'$, it holds

$$\varphi + \Psi(\mathcal{G}_\varphi) \leq \varphi' + \Psi'(\mathcal{G}_{\varphi'}).$$

We say that Ψ is *monotone* if $\Psi \leq_{\mathrm{mon}} \Psi$.

Lemma 4. *Let Ψ, Ψ' be sound, complete potential assigners, and assume $\Psi \leq_{\mathrm{mon}} \Psi'$. Then over any input game $\mathcal{G}$, the Ψ'-FVI algorithm terminates in less iterations than the Ψ-FVI algorithm.*

Lemma 5. *Let Ψ, Ψ_1, Ψ_2 be potential assigners. It holds:*

$$\Psi \leq_{\mathrm{mon}} \Psi_1 \leq \Psi_2 \implies \Psi \leq_{\mathrm{mon}} \Psi_2.$$

In particular, if Ψ is monotone and $\Psi \leq \Psi'$, then $\Psi \leq_{\mathrm{mon}} \Psi'$.

Lemma 6 ([7,19]). *The potential assigners Ψ_{First^+} and Ψ_{GKK} are monotone.*

Corollary 3. *The algorithms PPI and DPPI terminate in less iterations than either of SVI or GKK, over any input game.*

5 Experimental results

We focus on two distinct game-solving applications: energy game solving, which is the natural target for our algorithms, and parity game solving, which incurs a conversion cost to energy games but allows using established parity game benchmarks and comparison with other parity game solvers.

In both cases, the algorithms were implemented in Oink [10], a tool developed to provide a uniform framework for the comparison of parity game solvers.

After explaining the technical aspects of our implementation, and choices of algorithms and benchmarks, we discuss the most remarkable behaviours that can be observed in the experiments.

All the following experiments were carried on an Intel® Core™ i7-8700 CPU @ 3.20GHz paired with 16GiB of memory, each test being capped at 60 seconds and 10GiB of memory. Arithmetic operations over multiple precision integers are carried out using the GNU Multiple Precision Arithmetic library (GMP).

Set of algorithms. We compare our implementations of PPI, DPPI and their alternating versions (PPI-alt and DPPI-alt) to 4 other algorithms. Only one of them (QDPM) can be executed over general energy games, the other three are parity game solvers: QDPM from [3], and Zielonka's recursive algorithm (ZLK), Tangle learning (TL) and Recursive tangle learning (RTL) (winner of the last edition of SYNTCOMP) from [9,10].

We remark that we do not include comparisons with SVI [7], nor with GKK-DKZ [19,12], as these algorithms are known to be inefficient in practice [3] and incur in frequent timeouts. Also, we have not compared to an independent implementation of OSI, as we have not found one such implementation computing winning regions consistent with the rest of the algorithms.

5.1 Parity game solving

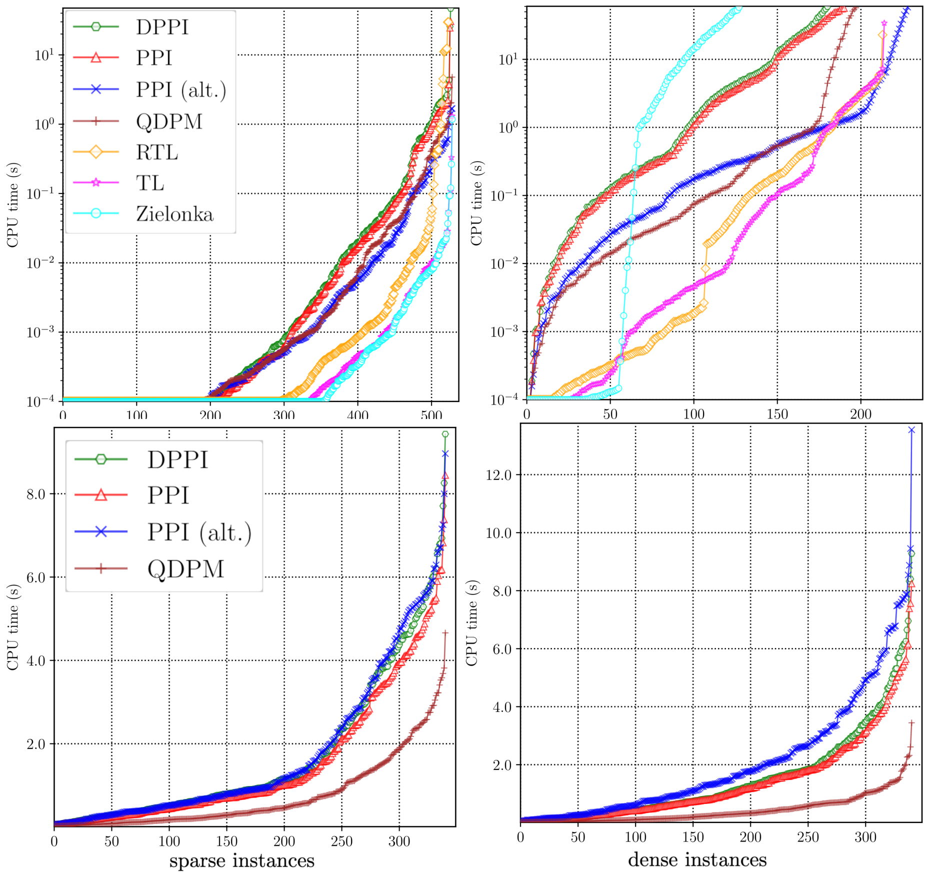
We rely on the yearly competition SYNTCOMP24 for our benchmarks, which has a competition track for parity game solvers. We subdivide the approximately 1000 benchmarks into two categories: synthetic games (crafted by researchers, usually with the intent of being hard for certain parity game solving approaches) and organic games (the natural counterpart of the synthetic games). We note that the synthetic games include the "two counter games" examples [11], in which TL and RTL show an exponential behaviour. It also contains the family of examples by Friedman [17], supposedly exponential for OSI.

As is usual in this settings we present the experimental results as a survival plot, which indicates how many tests are solved (x-axis) within a time limit (y-axis, time per test). In order to solve input parity games with energy games solvers, we first need to convert the parity game into an energy one. This costly step is included in the runtime of our algorithms as well as QDPM.

5.2 Energy game solving

We modified Oink so that it would accept negative weights and implemented a strategy-checker for energy games — this boils down to checking that, in the game restricted to the strategy, Max-winning strongly-connected components do not have infinite negative cycles, and symmetrically for Min.

We consider randomly generated bipartite graphs. The restriction to bipartite graphs is justified by the fact that, otherwise, the vast majority of vertices are part of winning cycles controlled by the same player, making the game (and its resolution) much easier. We separate instances that are sparse (the out-degree of each vertex is 2) or dense (the number of edges is $n^2/5$).



5.3 Conclusions of the experiments

In the light of the experiments above, we derive the following conclusions.

1. Overall, the fast value iteration framework captures several algorithms (PPI, PPI-alt, QDPM) that perform competitively in standard benchmarks of parity games. Despite being less efficient than leading parity game solvers, they are remarkably robust against hard instances, particularly PPI-alt.
2. The alternating version of PPI and DPPI, for which we were unable to prove termination in theory, always terminate. Moreover, over instances coming from parity games benchmarks, they are significantly faster than their asymmetric counterparts.
3. While DPPI was introduced as a theoretically enhanced version of PPI, there is no significant difference in the running time of these algorithms. In fact, DPPI tends to be slightly slower, due to the increased cost in the computation of the potential.

4. Although based on the same algorithmic ideas, QDPM consistently outperforms PPI, by almost an order of magnitude. This difference can be explained by two factors: (1) QDPM uses some smart implementation optimizations [3, Sect. 5], and (2) our implementation of PPI is tailored for (usual) edge-weighted games, whereas QDPM is implemented for vertex-labelled game (for which two weights outgoing a given vertex are always equal).

6 Conclusion and future work

We have presented a general framework to describe algorithms for energy games, capturing and providing simple descriptions and correctness proofs for many of the algorithms, including the top performing ones in practical scenarios. The fast value iteration framework raises numerous exciting questions; we outline some of them here.

New algorithms. The new framework provides a very easy way to propose new correct algorithms: it suffices to define a potential assigner which is sound, complete, and computable in polynomial time. We have isolated Ψ_{En^+} as a important potential assigner, implicitly used by the two fastest algorithms solving energy games, and presented the potential Ψ_{DPPI} which, while still being computable in polynomial time, is $\geq \Psi_{\mathrm{En}^+}$ in general.

Question 1. Does there exist a reasonable[10] potential assigner which is sound, complete, computable in polynomial time and $\geq \Psi_{\mathsf{DPPI}}$?

Alternating algorithms. Our framework also allows to design symmetric alternating algorithms, for which we are unable to prove termination using the currently available tools. Our empirical study shows that, in practice, these not only terminate, but are often considerably faster than their asymmetric counterparts.

Question 2. Do alternating fast value iterations terminate over simple games?

We stress the fact that the question is open for all sound and complete potential assigners (except for GKK, for which the alternating algorithm coincides with the normal one, see Section 4.3).

Lower bounds. Friedmann proposed notoriously involved constructions which provide exponentially many iterations for strategy improvement algorithms in [17]. Although it is claim (see [17, Sect. 4.6.2]) that these include OSI, our experiments show that PPI can solve these instances in linear time, and PPI-alt in a constant number of 2 iterations. Currently, we lack any family of examples in which PPI takes more than a linear number of iterations, although we expect that it should admit exponential lower bounds.

Question 3. Can one design superpolynomial lower bounds on the number of iterations for PPI? And (more challenging) for its alternating variant?

[10] A non-reasonable example meeting the requirements is $\Psi_{\mathsf{DPPI}} \circ \Psi_{\mathsf{DPPI}}$.

Randomized initialization. As remarked in Section 3.1, the weights of the cycles of $\mathcal{G}$ and $\mathcal{G}_\varphi$ coincide for any finite potential φ, so the threshold problem for the mean-payoff objective is equivalent over these games. Therefore, we can initialize a given game with an arbitrarily potential φ, and solve the "perturbed game". This directly provides a randomized version of any algorithm: add a random perturbation before execution. This idea is not novel, it was studied empirically by Beffara and Vorobyov [1] for the GKK algorithm; and lower bounds were later derived by Lebedev [24] for the same algorithm.

Question 4. Is the randomized variant of PPI subexponential? More generally, can we design a potential assigner whose associated randomized fast value iteration is subexponential?

Smooth analysis. An interesting parallel can be drawn with smooth analysis [35], which consider small perturbations of the input (randomized initialization is difference since we get an equivalent input). In fact, it was recently established that there is a strategy improvement algorithm for mean-payoff games that is polynomial in the sense of smooth analysis [25].

Question 5. Can the algorithm of [25] be recast as a fast value iteration?

References

1. Beffara, E., Vorobyov, S.: Is randomized gurvich-karzanov-khachiyan's algorithm for parity games polynomial? (2001)
2. Benerecetti, M., Dell'Erba, D., Mogavero, F.: Solving mean-payoff games via quasi dominions. In: TACAS. Lecture Notes in Computer Science, vol. 12079, pp. 289–306 (2020). https://doi.org/10.1007/978-3-030-45237-7_18
3. Benerecetti, M., Dell'Erba, D., Mogavero, F.: Solving mean-payoff games via quasi dominions. Inf. Comput. **297**, 105151 (2024). https://doi.org/10.1016/J.IC.2024.105151
4. Björklund, H., Sandberg, S., Vorobyov, S.G.: A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. In: MFCS. Lecture Notes in Computer Science, vol. 3153, pp. 673–685. Springer (2004). https://doi.org/10.1007/978-3-540-28629-5_52
5. Björklund, H., Vorobyov, S.G.: Combinatorial structure and randomized subexponential algorithms for infinite games. Theor. Comput. Sci. **349**(3), 347–360 (2005)
6. Bouyer, P., Fahrenberg, U., Larsen, K.G., Markey, N., Srba, J.: Infinite runs in weighted timed automata with energy constraints. In: FORMATS. Lecture Notes in Computer Science, vol. 5215, pp. 33–47. Springer (2008)
7. Brim, L., Chaloupka, J., Doyen, L., Gentilini, R., Raskin, J.: Faster algorithms for mean-payoff games. Formal Methods in System Design **38**(2), 97–118 (2011)
8. Calude, C.S., Jain, S., Khoussainov, B., Li, W., Stephan, F.: Deciding parity games in quasipolynomial time. In: STOC. pp. 252–263 (2017)
9. van Dijk, T.: Attracting tangles to solve parity games. In: CAV. Lecture Notes in Computer Science, vol. 10982, pp. 198–215. Springer (2018). https://doi.org/10.1007/978-3-319-96142-2_14, https://doi.org/10.1007/978-3-319-96142-2_14

10. van Dijk, T.: Oink: An implementation and evaluation of modern parity game solvers. In: TACAS. Lecture Notes in Computer Science, vol. 10805, pp. 291–308. Springer (2018). https://doi.org/10.1007/978-3-319-89960-2_16, https://doi.org/10.1007/978-3-319-89960-2_16
11. van Dijk, T.: A parity game tale of two counters. In: GandALF. EPTCS, vol. 305, pp. 107–122 (2019). https://doi.org/10.4204/EPTCS.305.8
12. Dorfman, D., Kaplan, H., Zwick, U.: A faster deterministic exponential time algorithm for energy games and mean payoff games. In: ICALP. pp. 114:1–114:14 (2019)
13. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. International Journal of Game Theory **109**(8), 109–113 (1979)
14. Emerson, E.A., Jutla, C.S.: Tree automata, μ-calculus and determinacy. In: FOCS. pp. 368–377. IEEE Computer Society (1991)
15. Fearnley, J.: Non-oblivious strategy improvement. In: LPAR. Lecture Notes in Computer Science, vol. 6355, pp. 212–230. Springer (2010). https://doi.org/10.1007/978-3-642-17511-4_13, https://doi.org/10.1007/978-3-642-17511-4_13
16. Fijalkow, N., Gawrychowski, P., Ohlmann, P.: Value iteration using universal graphs and the complexity of mean payoff games. In: MFCS. LIPIcs, vol. 170, pp. 34:1–34:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
17. Friedmann, O.: Exponential Lower Bounds for Solving Infinitary Payoff Games and Linear Programs. Ph.D. thesis, Ludwig Maximilians University Munich (2011), http://edoc.ub.uni-muenchen.de/13294/
18. Gallai, T.: Maximum-minimum sätze über graphen. Acta Math. Acad. Sci. Hung. (9), 395–434 (1958)
19. Gurvich, V.A., Karzanov, A.V., Khachiyan, L.G.: Cyclic games and an algorithm to find minimax cycle means in directed graphs. USSR Computational Mathematics and Mathematical Physics **28**, 85–91 (1988)
20. Jacobs, S., Perez, G.A., Abraham, R., Bruyere, V., Cadilhac, M., Colange, M., Delfosse, C., van Dijk, T., Duret-Lutz, A., Faymonville, P., Finkbeiner, B., Khalimov, A., Klein, F., Luttenberger, M., Meyer, K., Michaud, T., Pommellet, A., Renkin, F., Schlehuber-Caissier, P., Sakr, M., Sickert, S., Staquet, G., Tamines, C., Tentrup, L., Walker, A.: The reactive synthesis competition (SYNTCOMP): 2018-2021 (2022)
21. Jurdziński, M., Morvan, R., Ohlmann, P., Thejaswini, K.S.: A symmetric attractor-decomposition lifting algorithm for parity games. CoRR **abs/2010.08288** (2020), https://arxiv.org/abs/2010.08288
22. Jurdzinski, M., Morvan, R., Thejaswini, K.S.: Universal algorithms for parity games and nested fixpoints. In: Principles of Systems Design - Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 13660, pp. 252–271. Springer (2022). https://doi.org/10.1007/978-3-031-22337-2_12
23. Khachiyan, L., Gurvich, V., Zhao, J.: Extending Dijkstra's algorithm to maximize the shortest path by node-wise limited arc interdiction. In: CSR. Lecture Notes in Computer Science, vol. 3967, pp. 221–234. Springer (2006)
24. Lebedev, V.: Exponential examples of solving parity games. Computational Mathematics and Mathematical Physics **56**, 688–697 (2016)
25. Loff, B., Skomra, M.: Smoothed analysis of deterministic discounted and mean-payoff games. In: ICALP. LIPIcs, vol. 297, pp. 147:1–147:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2024). https://doi.org/10.4230/LIPICS.ICALP.2024.147

26. Luttenberger, M.: Strategy iteration using non-deterministic strategies for solving parity games. CoRR (2008), http://arxiv.org/abs/0806.2923
27. Luttenberger, M., Meyer, P.J., Sickert, S.: Practical synthesis of reactive systems from LTL specifications via parity games. Acta Informatica **57**(1-2), 3–36 (2020)
28. Meyer, P.J., Luttenberger, M.: Solving mean-payoff games on the GPU. In: ATVA. Lecture Notes in Computer Science, vol. 9938, pp. 262–267 (2016)
29. Meyer, P.J., Sickert, S., Luttenberger, M.: Strix: Explicit reactive synthesis strikes back! In: CAV. Lecture Notes in Computer Science, vol. 10981, pp. 578–586. Springer (2018)
30. Mostowski, A.W.: Games with forbidden positions. Tech. Rep. 78, University of Gdansk (1991)
31. Ohlmann, P.: The GKK algorithm is the fastest over simple mean-payoff games. In: CSR. Lecture Notes in Computer Science, vol. 13296, pp. 269–288. Springer (2022)
32. Puri, A.: Theory of Hybrid Systems and Discrete Event Systems. Ph.D. thesis, EECS Department, University of California, Berkeley (dec 1995)
33. Schewe, S.: An optimal strategy improvement algorithm for solving parity and payoff games. In: CSL. Lecture Notes in Computer Science, vol. 5213, pp. 369–384. Springer (2008)
34. Schewe, S., Trivedi, A., Varghese, T.: Symmetric strategy improvement. In: ICALP. Lecture Notes in Computer Science, vol. 9135, pp. 388–400. Springer (2015). https://doi.org/10.1007/978-3-662-47666-6_31, https://doi.org/10.1007/978-3-662-47666-6_31
35. Spielman, D.A., Teng, S.: Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. J. ACM **51**(3), 385–463 (2004). https://doi.org/10.1145/990308.990310
36. Thejaswini, K.S., Ohlmann, P., Jurdzinski, M.: A technique to speed up symmetric attractor-based algorithms for parity games. In: FSTTCS. LIPIcs, vol. 250, pp. 44:1–44:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPICS.FSTTCS.2022.44