On the ESL algorithm for solving energy games

Antonio Casares\textsuperscript{1} and Pierre Ohlmann\textsuperscript{2}
\textsuperscript{1} LaBRI, Université de Bordeaux, France
\textsuperscript{2} IRIF, Université de Paris, France

Abstract. We propose a variant of an algorithm introduced by Schewe and also studied by Luttenberger for solving parity or mean-payoff games. We present it over energy games and call our variant the ESL algorithm (for Energy-Schewe-Luttenberger). We find that using potential reductions as introduced by Gurvich, Karzanov and Khachiyan allows for a simple and elegant presentation of the algorithm, which repeatedly applies a natural generalisation of Dijkstra’s algorithm to the two-player setting due to Khachiyan, Gurvich and Zhao.

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

1 Introduction

Mean-payoff and energy games. In the games under study, two players, Min and Max, take turns in moving a token over a sinkless finite directed graph whose edges are labelled by (potentially negative) integers, interpreted as payoffs. In a mean-payoff game, the players aim to optimise (minimising or maximising, respectively) the average payoff in the long run. When playing an energy game, Min and Max optimise the supremum of the cumulative sum of values which takes values in $[0, +\infty]$.

These two games are determined \cite{17}; for each initial vertex $v$, there is a value $x$ such that starting from $v$, the minimiser can ensure an outcome $\leq x$ whereas the maximiser can ensure at least $x$. They are moreover positional determined \cite{8,3} which means that the players can achieve the optimal value even when restricted to strategies that do 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. As a consequence of positional determinacy, the energy value of a vertex is finite if and only if its mean-payoff value is non-positive \cite{4}, and therefore solving an energy game also solves the so called threshold problem for mean-payoff games. In fact, all state of the art algorithms \cite{4,7,2} for the threshold problem – further discussed below – actually go through solving the energy game. The best algorithms for the more general problems of computing the exact values or synthesising optimal strategies in the mean-payoff game also rely on solving many auxiliary energy games \cite{6}.
Fig. 1. Example of a game; circles and squares represent vertices which respectively belong to Min and Max. Mean-payoff values from left to right are $-2, -2, -\frac{1}{2}, -\frac{1}{2}, 1$ and 1, and mean-payoff-optimal positional strategies for both players are identified in bold. Energy values are $0, 2, 9, 0, \infty$ and $\infty$, and energy-optimal strategies are given by arrows with double heads.

Positional strategies achieving positive or non-positive mean-payoff values can be checked in polynomial time, and therefore the threshold problem belongs to NP \cap coNP. Despite numerous efforts, no polynomial algorithm is known. Mean-payoff games are known [22] to be more general than parity games [9,20] which enjoy a similar complexity status but were recently shown to be solvable in quasipolynomial time [5]. It is however unlikely that algorithms for solving parity games in polynomial time generalise to mean-payoff games [10].

**Schewe’s algorithm.** Schewe presented in [23] a strategy improvement algorithm for solving parity games. Schewe points out that his framework can be adapted to the more general case of mean-payoff games; one can actually see it as a switching policy in the combinatorial strategy improvement framework proposed by Björklund and Vorobiov [2]. Luttenberger [15] later formulated the same algorithm as one iterating over non-deterministic strategies: over such strategies, it can be rephrased as iterative applications of the natural “all-switch” policy.

At that time, the connection between mean-payoff and energy games – established only later in [3] and then simplified in [4] – was not well understood. In particular, the algorithms above, following [2], introduce an additional sink vertex, called retreat vertex, and restrict the iteration to so called admissible strategies, which can be understood as those guaranteeing a finite energy. Actually, Björklund and Vorobiov ask in their conclusion whether one can avoid the need for a retreat vertex and admissible strategies; a positive answer to this question is provided by the energy valuation (details are provided in the second author’s PhD thesis [21]).

**Our contribution.** We propose a variant of Schewe’s algorithm; while the main ideas are the same, the presentation as well as the precise execution of the algorithm differ. In particular, we do not require introducing a retreat vertex,
or restrict to a subclass of strategies. We also do not require the vocabulary from strategy improvements. Overall, we believe that our version is conceptually simpler and allows to appreciate the core algorithmic idea in a new light.

The algorithm is presented as one iterating succession of potential reductions, as introduced by Gurvich, Karzanov and Khachiyan [13], until obtaining a trivial game. Each iteration solves a game over only non-negative weights, which is done in $O(m + n \log n)$ operations using a simple extension of Dijkstra’s algorithm to the two-player setting at hands, due to Khachiyan, Gurvich and Zhao [14].

**Related work.** It is worth noting that Schewe’s algorithm is a key component in the LTL-synthesis tool STRIX [19,16], which has won all editions of the main annual synthesis competition SYNTCOMP. The algorithm was also ported to the GPU by Meyer and Luttenberger [18]. We also believe that there are similarities to be understood between the algorithm under study and the quasi-dominion approach of Benerecetti, Dell’Erba and Mogavero [1].

**Outline.** The first section introduces all necessary concepts and recalls the relationship between mean-payoff and energy games. The second section presents the ESL algorithm, and the third one proves its correctness and termination. We then conclude and discuss future work.

## 2 Preliminaries

In this preliminary section, we introduce mean-payoff and energy games, and potential reductions.

**Mean-payoff and energy games.** In this paper, a game is a tuple $G = (G, w, V_{\text{Min}}, V_{\text{Max}})$, where $G = (V, E)$ is a finite directed graph with no sink, $w : E \to \mathbb{Z}$ is a labelling of its edges by integer weights, and $V_{\text{Min}}, V_{\text{Max}}$ is a partition of $V$. We always use $n, m$ and $W$ respectively for $|V|, |E|$ and $\max_{e \in E} |w(e)|$; we say that vertices in $V_{\text{Min}}$ belong to Min while those in $V_{\text{Max}}$ belong to Max. We now fix a game $G = (G, w, V_{\text{Min}}, V_{\text{Max}})$.

A path is a (possibly empty, possibly infinite) sequence of edges $\pi = e_0 e_1 \ldots$ with matching endpoints: if $e_{i+1} = v_{i+1} v_{i+2}$ is defined then its first component $v_{i+1}$ matches the second component of $e_i$. For convenience, we often write $v_0 \to v_1 \to \ldots$ for the path $e_0 e_1 \cdots = (v_0 v_1) (v_1 v_2) \cdots$. Given a finite or infinite path $\pi = e_0 e_1 \ldots$, we let $w(\pi) = w(e_0) w(e_1) \ldots$ denote the sequence of weights appearing on $\pi$. The sum of a finite path $\pi$ is the sum of the weights appearing on it, we denote it by $\text{sum}(\pi)$.

Given a finite or infinite path $\pi = e_0 e_1 \cdots = v_0 \to v_1 \to \ldots$ and an integer $k \geq 0$, we let $\pi_{\leq k} = e_0 e_1 \cdots e_{k-1} = v_0 \to v_1 \to \cdots \to v_k$, and we let $\pi_{< k} = \pi_{< k+1}$. Note that $\pi_{< 0}$ is the empty path, and that $\pi_{< k}$ has length $k$ in general: it belongs to $E_k$. We say that $\pi$ starts in $v_0$, and when it is finite and of length $k$ that it ends in $v_k$. By convention, the empty path starts and ends in all vertices. A cycle is a finite path which starts and ends in the same vertex. A finite path
A valuation is a map \( \text{val} : \mathbb{Z}_\omega \to \mathbb{R}_\infty \) which assigns a potentially infinite real number to infinite sequences of weights. The three valuations which are studied in this paper are the mean-payoff, energy, and positive-energy valuations, given by

- Mean-payoff:
  \[
  \text{MP}(w_0w_1 \ldots) = \limsup_k \frac{1}{k} \sum_{i=0}^{k-1} w_i \in \mathbb{R},
  \]

- Energy:
  \[
  \text{En}(w_0w_1 \ldots) = \sup_k \sum_{i=0}^{k-1} w_i \in \mathbb{N}_\infty,
  \]

- Positive-energy:
  \[
  \text{En}^+(w_0w_1 \ldots) = \sum_{i=0}^{k_{\text{neg}}-1} w_i \in \mathbb{N}_\infty,
  \]

where \( k_{\text{neg}} = \min\{k \mid w_k < 0\} \in \mathbb{N}_\infty \) is the first index of a negative weight. For technical convenience, and only in the context of energy games, we will also consider games in which weights are potentially (positively) infinite. We thus extend the definitions of \( \text{En} \) and \( \text{En}^+ \) 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 three valuations are illustrated on a given weight profile 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 \( \pi = e_0e_1 \ldots = v_0 \to v_1 \to \ldots \) is consistent with \( \sigma \) if whenever \( e_i = v_iv_{i+1} \) is defined and such that \( v_i \in V_{\text{Min}} \), it holds that \( e_i = \sigma(v_i) \). We write in this case \( \pi \models \sigma \). Strategies for Max are defined similarly and written \( \tau : V_{\text{Max}} \to E \). Paths consistent with Max strategies are defined analogously and also denoted by \( \pi \models \tau \). In common vocabulary, the theorem below states that the three valuations are determined with positional strategies over finite games. It is well known for \( \text{MP} \) and \( \text{En} \) and easy to prove for \( \text{En}^+ \) (Lemma 3 provides an algorithmic proof of

---

**Fig. 2.** The three valuations over a given weight profile. 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 \( \leq 0 \), and both \( \text{En} \) and \( \text{En}^+ \) are finite.
this fact). We remark that positional determinacy also holds for the two energy valuations $\text{En}$ and $\text{En}^+$ over games where we allow for infinite weights.

**Theorem 1 ([8,3]).** For each $\text{val} \in \{\text{MP}, \text{En}, \text{En}^+\}$, there exist strategies $\sigma_0$ for Min and $\tau_0$ for Max such that for all $v \in V$ we have

$$\sup_{\pi|\pi=\sigma_0} \text{val}(w(\pi)) = \inf_{\sigma|\pi=\sigma} \sup_{\pi|\pi=\tau} \text{val}(w(\pi)) = \inf_{\tau|\pi=\tau} \inf_{\pi|\pi=\sigma} \text{val}(w(\pi)),$$

where $\sigma, \tau$ and $\pi$ 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 $\text{val}_{G}(v) \in \mathbb{R}^\infty$; the strategies $\sigma_0$ and $\tau_0$ 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 [4].

**Corollary 1 ([4]).** For all $v \in V$ it holds that

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

Therefore computing $\text{En}$ values of the games is harder than the threshold problem. As explained in the introduction, all state-of-the-art algorithms for the threshold problem actually compute $\text{En}$ values. This shifts our focus from mean-payoff to energy games.

**Potential reductions.** We fix a game $G = (G = (V, E), w, V_{\text{Min}}, V_{\text{Max}})$. A potential is a map $\phi : V \to \mathbb{N}^\infty$. Potentials are partially ordered coordinatewise. Given an edge $vv' \in E$, we define its $\phi$-modified weight to be

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

The $\phi$-modified game $G_{\phi}$ is simply the game $(G, w_{\phi}, V_{\text{Min}}, V_{\text{Max}})$; informally, all weights are replaced by the modified weights. Note that the underlying graph does not change, in particular paths in $G$ and $G_{\phi}$ are the same. Note that any edge outgoing a vertex $v$ with potential $\phi(v) = \infty$ has weight $\infty$ in the modified game, therefore $v$ has $\text{En}$ and $\text{En}^+$-values $\infty$ in $G_{\phi}$.

Observe that for a finite path $\pi = v_0 \to v_1 \to \ldots \to v_k$ which visits only vertices with finite potential, its sum in $G_{\phi}$ is given by

$$\text{sum}_{\phi}(v) = \text{sum}(\pi) - \phi(v_0) + \phi(v_k).$$

We let 0 denote the constant zero potential; note that $G_0 = G$. For convenience, we use $\text{En}_{\phi}$ to denote $\text{En}_{G_{\phi}}$ (we remark that $\text{En}_0 = \text{En}_{G}$). We will omit the subscript whenever the game or potential under consideration is clear from the context.
Moving from \( G \) to \( G_\phi \) for a given potential \( \phi \) is called a potential reduction; these were introduced by Gallai [12] for studying network related problems such as shortest-paths problems. In the context of mean-payoff or energy games, they were introduced in [13] and later rediscovered numerous times. The result below describes the effect of potential reductions over energy values.

**Theorem 2.** If \( \phi \) satisfies \( \phi \leq E_{n0} \), then it holds that \( E_{n0} = \phi + E_{n\phi} \) over \( V \).

A full proof is given in Appendix A for completeness, and an illustration is given in Figure 3. We say that a potential \( \phi \) is sound if it satisfies the hypothesis of Theorem 2, \( \phi \leq E_{n0} \).

![Fig. 3. An illustration of the second item in Theorem 2. For the three vertices on the right, energy values in both games are \( \infty \).](image)

Observe that \((G_\phi)_{\phi'} = G_{\phi + \phi'}\): sequential applications of potential reductions correspond to reducing with respect to the sum of the potentials. The following is easily derived as a consequence of Theorem 2.

**Lemma 1.** If \( \phi \) is sound for \( G \), and \( \phi' \) is sound for \( G_{\phi} \), then \( \phi + \phi' \) is sound for \( G \).

**Simple games.** 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 2.** Let \( G = (G, V, V_{\text{Min}}, V_{\text{Max}}) \) be an arbitrary game. The game \( G' = (G, V, (n + 1)w - 1, V_{\text{Min}}, V_{\text{Max}}) \) is simple and has the same vertices of positive mean-payoff values as \( G \).

Note moreover that simplicity is preserved by potential transformations, since sums of weights over cycles are left unchanged.
3 The ESL algorithm

3.1 Presentation of the algorithm

The ESL algorithm is based on successively applying sound potential reductions \( \phi_0, \ldots, \phi_j \) until a game \( G' \) is reached where energy-values are either 0 or \( \infty \). Thanks to Theorem 2 and Lemma 1, we then have \( E_{G} = E_{G'} + \phi_0 + \cdots + \phi_j \); in particular, a vertex has finite energy (or non-positive mean-payoff) in \( G \) if and only if it has energy 0 in \( G' \), and in this case its energy in \( G \) is given by \( \phi_0(v) + \cdots + \phi_j(v) \).

The potential reductions computed by each iteration of the ESL algorithm are precisely the \( E_{G}^{+} \)-values in the game. Intuitively, the players optimise the (non-negative) sum of the weights seen before the first negative weight. Since \( E_{G}^{+} \leq E_{G} \) in general, the potential \( E_{G}^{+} : V \to \mathbb{Z}^\infty \) is indeed sound, as required by our approach. The fact that it can be computed efficiently follows from the fact that only non-negative weights are considered, and therefore a straightforward two-player extension of Dijkstra’s algorithm, due to\(^3\) Khachiyan, Gurvich and Zhao [14] can be applied. A similar subroutine was also given by Schewe [23], whereas Luttenberger [15] uses an adaptation of the algorithm of Bellman-Ford which is less efficient.

Lemma 3 (Based on [14]). Over simple games, the \( E_{G}^{+} \)-values can be computed in \( O(m + n \log n) \) operations.

Proof. We start by determining in linear time \( O(m) \) the set \( N \) of vertices from which Min can force to immediately visit an edge of negative weight; these have \( E_{G}^{+} \)-value 0. We will successively update a set \( F \) containing the set of vertices over which \( E_{G}^{+} \) is currently known. We initialise this set \( F \) to \( N \). Note that all remaining Min vertices have only non-negative outgoing edges, and all remaining Max vertices have (at least) a non-negative outgoing edge.

We then iterate the two following steps illustrated in Figure 4. (A complexity analysis is given below.)

1. If there is a Max vertex \( v \notin F \) all of whose non-negative outgoing edges \( vv' \) lead to \( F \), set \( E_{G}^{+}(v) \) to be the maximal \( w(vv') + E_{G}^{+}(v') \), add \( v \) to \( F \), and go back to 1.
2. Otherwise, let \( vv' \) be an edge from \( V_{\text{Min}} \setminus F \) to \( F \) (it is necessarily positive) minimising \( w(vv') + E_{G}^{+}(v') \); set \( E_{G}^{+}(v) = w(vv') + E_{G}^{+}(v') \), add \( v \) to \( F \) and go back to 1. If there is no such edge, terminate.

After the iteration has terminated, there remains to deal with \( F^c \), which is the set of vertices from which Max can ensure to visit only non-negative edges forever. Since the arena is assumed to be simple (no simple cycle has weight zero) it holds that \( E_{G}^{+} \) is \( \infty \) over \( F \), and we are done\(^4\).

\(^{3}\) This corresponds to Theorem 1 in [14], case (i) with blocking systems \( B_2 \).

\(^{4}\) If the arena is not simple, one must additionally solve a Büchi game, and the complexity of the iteration is increased. We believe that this increased cost can be amortised overall, but give no details for this claim.
Fig. 4. The game version of Dijkstra’s algorithm; blue edges are negative and red ones are non-negative. If there is a vertex such as \( v \) (it belongs to \( \text{Max} \) and all edges pointing out of \( F \) are \( < 0 \)), one may set the value of \( v \). Otherwise, set the value of a \( \text{Min} \) vertex minimising \( w(vv') + \text{En}^+(v') \) over edges going to \( F \); if there is no such edge, terminate (\( \text{Max} \) can force seeing \( \geq 0 \) edges forever).

As is standard, by storing the number of edges outgoing from \( \text{Max} \) vertices in \( F^c \) to \( F \), step 1 induces only a total linear runtime \( O(m) \). For step 2, one should store, for each \( v \in V_{\text{Max}} \setminus F \), the edge towards \( F \) minimising \( w(vv') + \text{En}^+(v') \) in a priority queue. Using a Fibonacci heap as was first suggested by Fredman and Tarjan [11] for Dijkstra’s algorithm lowers the complexity from \( O(m \log n) \) to \( O(m + n \log n) \).

3.2 Termination and correctness

Termination. To prove that the ESL algorithm terminates in finitely many steps, we rely on a simple lemma which states that the set of vertices from which \( \text{Min} \) can ensure to immediately see a negative weight (and thus stop the game) can only shrink throughout the iteration.

Lemma 4. Let \( G' = G_{\text{En}_0^+} \), let \( N \) and \( N' \) be the sets of vertices from which \( \text{Min} \) can ensure to immediately visit a negative weight, respectively in \( G \) and \( G' \). We have \( N' \subseteq N \).

Proof. We show that \( N^c \subseteq N'^c \); let \( v \in N^c \). If \( \text{En}_G^+(v) = \infty \) then \( v \) has only outgoing edges of infinite weight in \( G' \) thus is cannot belong to \( N' \); we assume otherwise.

- Assume \( v \in V_{\text{Max}} \). Let \( \tau \) be an \( \text{En}_G^+ \)-optimal strategy in \( G \), and let \( \tau(v) = vv' \in E \). Since \( v \notin N \) we have \( w(vv') \geq 0 \) and \( \text{En}_G^+(v) = w(vv') + \text{En}_G^+(v') \).
  Hence we have \( w_{\text{En}_G^+}(vv') = w(vv') + \text{En}_G^+(v') - \text{En}_G^+(v) = 0 \geq 0 \) so \( v \notin N' \).
- Assume now that \( v \in V_{\text{Min}} \). We have for all \( vv' \in E \) that \( w(vv') \geq 0 \) hence \( \text{En}_G^-(v) \leq w(vv') + \text{En}_G^+(v') \), and thus \( w_{\text{En}_G^+}(vv') \geq 0 \), so \( v \notin N' \). \( \Box \)

We now let \( G_0 = G \) denote the initial game, and for each \( j \geq 0 \) we let \( \phi_j = \text{En}_{G_j}^+ \) and \( G_{j+1} = (G_j)_{\phi_j} \) be the game obtained after \( j \) iterations of the ESL algorithm. We also let \( \Phi_j = \phi_0 + \cdots + \phi_{j-1} \); it holds that \( G_j = (G_0)_{\Phi_j} \).
This lemma directly gives (with obvious notations) \( N_0 \supseteq N_1 \supseteq \ldots \) and therefore vertices \( v' \) in \( N_j \) satisfy \( \Phi_j(v') = 0 \). Now if \( v \) is a vertex such that \( \phi_j(v) = \text{En}^+_{\Phi_j}(v) \) is finite, then by definition there is a simple path \( \pi = v_0 \rightarrow \ldots \rightarrow v_k \) in \( \mathcal{G} \) from \( v \) to some \( v' \in N_j \) whose \( \Phi_j \)-modified sum is \( \sum_{\pi} \phi_j(\pi) = \phi_j(v) \). This rewrites as

\[
\phi_j(v) = -\Phi_j(v) + \Phi_j(v') + \sum_{i=0}^{k-1} w(\pi_i) \leq nW,
\]

and thus \( \Phi_j(v) \leq nW \). Stated differently, finite values remain \( \leq nW \), which guarantees termination in at most \( O(n^2N) \) iterations.

Using an implementation similar to the BCDGR algorithm [4], where the set of vertices with positive \( \text{En}^+ \) (as well as relevant internal data structures) is stored from an iteration to the next, one may lower the global complexity upper bound to \( O((m+n\log n)nW) \), at least for simple arenas. Such an implementation was already suggested by Schewe [23], but no complexity bound is given for mean-payoff games\(^5\).

We give a full example over a game of size 15 in Figure 5.

**Correctness.** The iteration terminates after \( j \) steps if \( \Phi_{j+1} = \Phi_j \). We now state and prove correctness of the ESL algorithm.

**Lemma 5.** If \( \Phi_{j+1} = \Phi_j \) then \( \text{En}_j \) takes values in \( \{0, \infty\} \) over \( V \).

**Proof.** Note that vertices such that \( \Phi_j(v) = \infty \) have only outgoing edges of weight \( \infty \) in \( \mathcal{G}_j \) and therefore they have \( \text{En}^+ \)-value \( \infty \). Hence, the hypothesis rewrites as \( \Phi_j(v) < \infty \implies \text{En}^+_{\Phi_j}(v) = 0 \); we let \( F \) be the set of vertices \( v \) with \( \Phi_j(v) < \infty \). This implies that all Min vertices in \( F \) have a non-positive outgoing edge in \( \mathcal{G}_j \) towards \( F \), and all Max vertices in \( F \) have all their outgoing edges non-positive and towards \( F \), hence the result. \( \square \)

### 4 Conclusion

We have presented the ESL algorithm using potential reductions. This allows to reduce to several iterations over non-negative weights, each of which can be treated efficiently using Dijkstra’s algorithm. In particular, presenting the algorithm does not require introducing a retreat vertex, or using vocabulary from strategy improvements. We believe that this new presentation sheds a lot of clarity on this important algorithmic idea.

\(^5\) Schewe’s presentation is done over parity games; no complexity bound is given for mean-payoff games, which are only mentioned in a footnote. This is quite unfortunate, since his algorithm would have improved over the state of the art at that time (and also suggests the fruitful link with energy games, which was exploited by [4] only a few years later).
Fig. 5. A complete execution of the ESL algorithm. In each iteration, we indicate the $E_n^+$-values of each vertex. Optimal strategies are indicated with bold arrows. Vertices from which Min can force to immediately see a negative weight are coloured in blue, and those with a strictly positive $E_n^+$-value in red.

We end the paper with a possible extension of these ideas. One may also compute, in the very same fashion, the $E_n^-$-values of the game, where $E_n^- : \mathbb{Z}^\omega \rightarrow [-\infty, 0]$ is given by $E_n^- (w_0 w_1 \ldots) = \sum_{i=0}^{k_{pos}-1} w_i$, with $k_{pos} = \min\{k \mid w_k > 0\}$. Iterating $E_n^-$ potential transformations gives rise to a dual algorithm, which of course terminates with similar complexity.

We have observed empirically that alternately iterating $E_n^+$ and $E_n^-$ potential transformation leads to a terminating algorithm over any instance. Moreover, it achieves even fewer iterations than the (asymmetric) ESL algorithm, and especially so over parity games, for which we have witnessed a significant gain over large random instances.

However, we have not been able to derive its termination using the currently available tools. Could one prove termination of the (symmetric) alternating algorithm? Could we hope for a subexponential combinatorial upper bound?

---

6 This requires working with potentials with values in $\mathbb{Z} \cup \{-\infty, \infty\}$, which is a formality.
Acknowledgments. We are grateful to Alexander Kozachinskyi for pointing out to us several important references. We also thank Thomas Colcombet, Nathanaël Fijalkow and Olivier Serre for fruitful discussions.

References


A Proof of Theorem 2

This short appendix is devoted to the proof of Theorem 2. We use the following result.

**Lemma 6.** Let $\sigma_0$ be an $E_0$-optimal Min strategy in $G$ and $\pi = v_0 \to v_1 \to \ldots \to v_k$ be a finite path consistent with $\sigma_0$ such that $E_0(v_k) < \infty$. Then we have $\sum(\pi) \leq E_0(v_0) - E_0(v_k)$.

**Proof.** Let $\pi'$ be an infinite path from $v_k$ consistent with $\sigma_0$ and such that $E_0(v_k) = \sum(w(\pi'))$. Then $\pi\pi'$ is consistent with $\sigma_0$ thus $E_0(v_0) \geq \sum(w(\pi'))$ by optimality. We thus obtain

\[
E_0(v_0) \geq \sum(w(\pi\pi')) = \sup_{k' \geq 0}(\sum((\pi\pi')_{<k'})) \\
\geq \sup_{k' \geq k}(\sum((\pi\pi')_{<k'})) \\
= \sum(\pi) + \sup_{k' \geq 0}\sum(\pi'_{<k'}) \\
= \sum(\pi) + \sum(w(\pi')) = \sum(\pi) + E_0(v_k).
\]

\[ \Box \]

We may now derive the wanted result.

**Proof (Proof of Theorem 2).** Let $\phi : V \to \mathbb{N}^\infty$ be a potential such that $\phi \leq E_0$; we aim to prove that $E_0 = \phi + E_\phi$ over $V$. Consider first a vertex $v$ with $E_0(v) = \infty$, fix an optimal Max strategy $\tau_0$ in $G$ and an infinite path $\pi = e_0e_1 \cdots = v_0 \to v_1 \to \ldots$ consistent with $\tau_0$ from $v$; by definition we have $\sum(w(\pi)) = \sup_k \sum_{i=0}^{k-1} w(e_i) = \infty$. We claim that $\sum(w_\phi(\pi)) = \infty$ which proves the wanted equality over $v$ (both terms are infinite).

- If for some $i$, $w(e_i) = \infty$ then $w_\phi(e_i) = \infty$ which implies the result.
- If for some $i$, $\phi(v_i) = \infty$ then again we have $w_\phi(e_i) = \infty$.
- Otherwise, we have for all $k$

\[
\sum_\phi(\pi_{<k}) = \phi(v_k) - \phi(v_0) + \sum(\pi_{<k}),
\]

and therefore $\sup_k \sum_\phi(\pi_{<k}) = \sup_k \sum(\pi_{<k}) = \infty$, the wanted result.

We now consider a vertex $v$ such that $E_0(v) < \infty$. Consider an $E$-optimal Min strategy $\sigma_0 : V_{\text{Min}} \to E$ in $G$ and let $\pi = v_0 \to v_1 \to \ldots$ be an infinite path consistent with $\sigma_0$ starting from $v_0 = v$. Note that for any $k \geq 0$, $v_k$ has finite energy value, and thus we obtain thanks to Lemma 6 and the hypothesis $\phi \leq E_0$ that

\[
\sum_\phi(\pi_{<k}) = \sum(\pi_{<k}) + \phi(v_k) - \phi(v_0) \\
\leq E_0(v_0) - E_0(v_k) + \phi(v_k) - \phi(v_0) \leq E_0(v_0) - \phi(v_0).
\]
hence \( \text{En}_\phi(v_0) = \sup_{\pi \models \sigma_0} \sup_{k \geq 0} \sum_{\phi < k} \phi < k \leq \text{En}_0(v_0) - \phi(v_0) \).

For the other inequality, consider an optimal Min strategy \( \sigma_\phi \) in \( G_\phi \), and let \( \pi \) be an infinite path from \( v_0 = v \) consistent with \( \sigma_\phi \). By applying Lemma 6 in \( G_\phi \) we now get

\[
\sum_{\pi < k} = \sum_{\phi < k} \phi < k - \phi(v_k) + \phi(v_0) \leq \text{En}_\phi(v_0) - \text{En}_\phi(v_k) + \phi(v_0) \geq 0, \geq 0 \leq \text{En}_\phi(v_0) + \phi(v_0),
\]

and the wanted result follows by taking a supremum. \( \square \)