A Pumping Algorithm for Ergodic Mean Payoff Stochastic Games with Perfect Information *†

Endre Boros‡ Khaled Elbassioni§ Vladimir Gurvich¶ Kazuhisa Makino‖

Abstract

In this paper, we consider two-person zero-sum stochastic mean payoff games with perfect information, or BWR-games, given by a digraph $G = (V = V_B \cup V_W \cup V_R, E)$, with local rewards $r : E \to \mathbb{R}$, and three types of vertices: black V_B, white V_W, and random V_R. The game is played by two players, White and Black: When the play is at a white (black) vertex v, White (Black) selects an outgoing arc (v, u). When the play is at a random vertex v, a vertex u is picked with the given probability $p(v, u)$. In all cases, Black pays White the value $r(v, u)$. The play continues forever, and White aims to maximize (Black aims to minimize) the limiting mean (that is, average) payoff. It was recently shown in [BEGM09] that BWR-games are polynomially equivalent with the classical Gillette games, which include many well-known subclasses, such as cyclic games, simple stochastic games (SSGs), stochastic parity games, and Markov decision processes. In this paper, we give a new algorithm for solving BWR-games in the *ergodic case*, that is when the game's value does not depend on the initial position. Our algorithm solves a BWR-game by reducing it, using a potential transformation, to a canonical form in which the optimal strategies of both players and the value for every initial position are obvious, since a locally optimal move in it is optimal in the whole game. We show that this algorithm is pseudo-polynomial when the number of random nodes is constant. We also provide an almost matching lower bound on its running time and show that this bound holds for a wider class of algorithms. Let us add that the general (non-ergodic) case is at least as hard as SSGs, for which no pseudo-polynomial algorithm is known.

Keywords: mean payoff games, local reward, Gillette model, perfect information, potential, stochastic games

1 Introduction

1.1 BWR-games

We consider two-person zero-sum stochastic games with perfect information and mean payoff: Let $G = (V, E)$ be a digraph whose vertex-set V is partitioned into three subsets $V = V_B \cup V_W \cup V_R$ that correspond to black, white, and random positions, controlled respectively, by two players, *Black* - the *minimizer* and *White* - the *maximizer*, and by nature. We also fix a *local reward* function $r : E \to \mathbb{R}$, and probabilities $p(v, u)$ for all arcs (v, u) going out of $v \in V_R$. Vertices $v \in V$ and arcs $e \in E$ are called *positions* and *moves*, respectively. In a personal position $v \in V_W$ or $v \in V_B$ the

*This research was partially supported by DIMACS, Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, and by the Scientific Grant-in-Aid from Ministry of Education, Science, Sports and Culture of Japan, and Center for Algorithmic Game Theory, University of Aarhus. Part of this research was done at the Mathematisches Forschungsintitüt Oberwolfach during a stay withing the Research in Pairs Program from March 7 to March 20, 2010.

†A preliminary version of this paper has appeared in *Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization (IPCO 2010).*

‡RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003; (boros@rutcor.rutgers.edu)

§Max-Planck-Institute for Informatics; Stuhlsatzenhausweg 85, 66123, Saarbruecken, Germany (elbassio@mpi-sb.mpg.de)

¶RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003; (gurvich@rutcor.rutgers.edu)

‖Graduate School of Information Science and Technology, University of Tokyo, Tokyo, 113-8656, Japan; (makino@mist.i.u-tokyo.ac.jp)

corresponding player White or Black selects an arc (v, u), while in a random position $v \in V_R$ a move (v, u) is chosen with the given probability $p(v, u)$. In all cases Black pays White the reward $r(v, u)$.

From a given initial position $v_0 \in V$ the game produces an infinite walk (called a *play*). White's objective is to maximize the *limiting mean payoff*

$$c = \liminf_{n \to \infty} \frac{\sum_{i=0}^{n} b_i}{n+1}, \tag{1}$$

where b_i is the reward incurred at step i of the play, while the objective of Black is the opposite, that is, to minimize $\limsup_{n\to\infty} \frac{\sum_{i=0}^{n} b_i}{n+1}$.

For this model it was shown in [BEGM09] that a *saddle point* exists in *pure positional uniformly optimal* strategies. Here "pure" means that the choice of a move (v, u) in a personal position $v \in V_B \cup V_W$ is deterministic; "positional" means that this choice depends solely on v, not on previous positions or moves; finally, "uniformly optimal" means that it does not depend on the initial position v_0, either.

This class of the *BWR-games* was introduced in [GKK88]; see also [CH08]. It was recently shown in [BEGM09] that the BWR-games and classical Gillette games [Gil57] are polynomially equivalent. The special case when there are no random positions, $V_R = \emptyset$, is known as *cyclic*, or *mean payoff*, or *BW-games*. They were introduced for the complete bipartite digraphs in [Mou76b, Mou76a], for all (not necessarily complete) bipartite digraphs in [EM79], and for arbitrary digraphs in [GKK88]. A more special case was considered extensively in the literature under the name of *parity games* [BV01a, BV01b, CJH04, Hal07, Jur98, JPZ06], and later generalized also to include random nodes in [CH08]. A BWR-game is reduced to a *minimum mean cycle problem* in case $V_W = V_R = \emptyset$, see, for example [Kar78]. If one of the sets V_B or V_W is empty, we obtain a *Markov decision process*; see, for example, [MO70]. Finally, if both are empty $V_B = V_W = \emptyset$, we get a *weighted Markov chain*[1].

It was noted in [BG09] that "parlor games", like Backgammon (and even Chess) can be solved in pure positional uniformly optimal strategies, based on their BWR-model.

In the special case of a BWR-game, when all rewards are zero except at a single node t called the terminal, at which there is a self-loop with reward 1, we obtain the so-called *simple stochastic games* (SSGs), introduced by Condon [Con92, Con93] and considered in several papers [GH08, Hal07]. In these games, the objective of White is to maximize the probability of reaching the terminal, while Black wants to minimize this probability. Recently, it was shown that Gillette games (and hence BWR-games by [BEGM09]) are equivalent to SSGs under polynomial-time reductions [AM09]. Thus, by recent results of Björklund, Vorobyov [BV05], and Halman [Hal07], all these games can be solved in randomized strongly subexponential time $2^{O(\sqrt{n_d \log n_d})}$, where $n_d = |V_B| + |V_W|$ is the number of *deterministic* vertices. Let us note that several pseudo-polynomial and subexponential algorithms exists for BW-games [GKK88, KL93, ZP96, Pis99, BV01a, BV01b, BSV04, BV07, Hal07, Vor08]; see also [DG06] for a policy iteration method, and [JPZ06] for parity games.

Besides their many applications (see e.g. [Lit96, Jur00]), all these games are of interest to Complexity Theory: Karzanov and Lebedev [KL93] (see also [ZP96]) proved that the decision problem "whether the value of a BW-game is positive" is in the intersection of NP and co-NP. Yet, no polynomial algorithm is known for these games, see e.g., the recent survey by Vorobyov [Vor08]. A similar complexity claim can be shown to hold for SSGs and BWR-games, see [AM09, BEGM09].

While there are numerous pseudo-polynomial algorithms known for the BW-case, it is a challenging open question whether a pseudo-polynomial algorithm exists for SSGs or BWR-games.

In fact, developing an efficient algorithm for stochastic games with perfect information was mentioned as an open problem in the survey [RF91]. Our results can be viewed as a partial solution of this problem for the case when the number of random nodes $|V_R|$ is fixed.

Let us also note that our "pumping" technique can be extended to ergodic stochastic games with imperfect information [BEGM10].

[1] Let us remark that the concept of ergodicity is different in the Game and Markov Chain theories. In particular, a game may be ergodic even if the Markov chain generated by some optimal strategies has multiple absorbing classes; compare [KS63] and Definition 1.

1.2 Potential transformations and canonical forms

Given a BWR-game, we consider *potential transformations* $x : V \to \mathbb{R}$, assigning a real-value $x(v)$ to each vertex $v \in V$, and transforming the local reward on each arc (v, u) to $r_x(v, u) = r(v, u) + x(v) - x(u)$. It is known that for BW-games there exists a potential transformation such that, in the obtained game the locally optimal strategies are globally optimal, and hence, the value and optimal strategies become obvious [GKK88]. This result was extended for the more general class of BWR-games in [BEGM09]: in the transformed game, the equilibrium value $\mu(v) = \mu_x(v)$ is given simply by the maximum local reward for $v \in V_W$, the minimum local reward for $v \in V_B$, and the average local reward for $v \in V_R$. In this case we say that the transformed game is in *canonical* form.

It is not clear how the algorithm given in [GKK88] for the BW-case can be generalized to BWR-case. On the other hand, the methods in [BEGM09] are similar to those of Gillette [Gil57]; see also Liggett and Lippman [LL69]: First, they analyze the so-called *discounted* version, in which the payoff is discounted by a factor β^i at step i, giving the effective payoff: $a_\beta = (1 - \beta)\sum_{i=0}^{\infty} \beta^i b_i$, and then they proceed to the limit as the *discount factor* $\beta \in [0, 1)$ tends to 1. While such an approach is sufficient to prove the existence of a canonical form, it does not provide an algorithm to compute the potentials, since the corresponding limits can be infinite. In this paper, we give such an algorithm that does *not* go through the discounted case. Our method computes an optimal potential transformation in case the game is *ergodic*, that is, when the optimal values do not depend on the initial position. If the game is not ergodic then our algorithm terminates with a proof of non-ergodicity, by exhibiting at least two vertices with provably distinct values. Unfortunately, our approach cannot be applied recursively in this case. This is unpleasant but not a complete surprise, since this case is at least as hard as SSGs, for which no pseudo-polynomial algorithm is known.

Theorem 1 *Consider a BWR-game with k random nodes, a total of n vertices, and integer rewards in the range of size R, and assume that all probabilities are rational with common denominator W. Then there is an algorithm that runs in time $(2^k nW)^{O(k)} R \log R$ and either brings the game by a potential transformation to canonical form, or proves that it is non-ergodic.*

Let us remark that the ergodic case is frequent in applications. For instance, it is the case when $G = (V_W \cup V_B \cup V_R, E)$ is a complete tripartite digraph (where $p(v, u) > 0$ for all $v \in V_R$ and $(v, u) \in E$); see Section 3 for more general sufficient conditions.

Theorem 1 states that our algorithm is pseudo-polynomial if the number of random nodes is fixed. As far as we know, this is the first algorithm with such a guarantee, in comparison, for example, to strategy improvement methods [BSV04, HK66, VJ00], for which exponential lower bounds are known [Fri09]. It is worth mentioning that the algorithm of [HK66] also works only for the ergodic case. However, the concept of ergodicity in [HK66] requires that for any pair of strategies of the players, the obtained Markov chain is ergodic (in the sense of [KS63]). In contrast, our condition of ergodicity is much weaker and requires only that the value does not depend on the initial position. Such concept is very close to the one defined in [Fed80]. However, unlike [Fed80] (and most other algorithms in the literature; see [RF91]), we do not make use of the approximation by discounted games. In particular, our proof of existence of the solution does not rely on the use of the classical Hardy-Littlewood theorem [HL31]. We are only ware of two more algorithms that do not make use of limit transitions $\beta \to 1$: one is the algorithm by Hoffman and Karp [HK66] mentioned above, and the other is the algorithm suggested by Vrieze, Tijs, Raghavan, and Filar [VTRF83]. For the first algorithm convergence, but not finiteness, was proved. The latter algorithm works without any additional restriction for the class of so-called *switching controller games*, which is wider than the class of BWR-games. For this algorithm finiteness was proved in [VTRF83], yet no bounds on the running time are known.

We are not aware of any previous results bounding the running time of an algorithm for a class of BWR-games in terms of the number of random nodes, except for [GH08] which shows that simple stochastic games on k random nodes can be solved in time $O(k!(|V||E|+L))$, where L is the maximum bit length of a transition probability. It is worth remarking here that even though BWR-games are polynomially reducible to simple stochastic games, under this reduction the number of random nodes k becomes a polynomial in n, even if the original BWR-game has constantly many random nodes.

In particular, the result in [GH08] does not imply a bound similar to that of Theorem 1 for general BWR-games.

One should also contrast the bound in Theorem 1 with the subexponential bounds in [Hal07]: roughly, the algorithm of Theorem 1 will be more efficient if $|V_R|$ is $o((|V_W| + |V_B|)^{\frac{1}{4}})$ (assuming that W and R are polynomials in n). Moreover, our algorithm could be practically much faster since it can stop much earlier than its estimated worst-case running time (unlike the subexponential algorithms [Hal07], or those based on dynamic programming [ZP96]). In fact, our algorithm can be used to approximate the value of an ergodic BWR-game within a prescribed additive error ε, and as our preliminary experiments indicate, on a random game of up to $15,000$ nodes, for $\varepsilon = 0.001$, the algorithm takes no more than a few hundred iterations, even if the maximum reward is very large (see Table 1 in Appendix E for details). One more desirable property of this algorithm is that it is of the *certifying* type (see e.g. [KMMS03]), in the sense that, given an optimal pair of strategies, the vector of potentials provided by the algorithm can be used to verify optimality in *linear* time (otherwise verifying optimality requires solving two linear programs).

1.3 Overview of the techniques

Our algorithm for proving Theorem 1 is quite simple. Starting from zero potentials, and depending on the current locally optimal rewards (maximum for White, minimum for Black, and average for Random), the algorithm keeps selecting a subset of nodes and reducing their potentials by some value, until either the locally optimal rewards at different nodes become sufficiently close to each other, or a proof of non-ergodicity is obtained in the form of a certain partition of the nodes. The upper bound on the running time consists of three technical parts. The first one is to show that if the number of iterations becomes too large, then there is a large enough potential gap to ensure non-ergodicity. In the second part, we show that the range of potentials can be kept sufficiently small throughout the algorithm, namely $\|x^*\|_\infty \leq nRk(2W)^k$, and hence the range of the transformed rewards does not explode. The third part concerns the required accuracy. We show that it is enough in our algorithm to get the value of the game within an accuracy of

$$\varepsilon = \frac{1}{2^{2k(k+1)} k^{2k} n^{2(k+1)} W^{2k+2}}, \tag{2}$$

in order to guarantee that it is equal to the exact value. As far as we know, such a bound in terms of k is new, and it could be of independent interest. For the special case of Markov decision processes (when V_B or V_W is empty), the potentials mentioned in the theorem correspond to the *dual* variables in the standard linear programming formulation; see e.g. [MO70][2]. We also show the lower bound $W^{\Omega(k)}$ on the running time of the algorithm of Theorem 1 by constructing a series of examples of the problem, with only random nodes (that is, Markov chains).

The paper is organized as follows. In the next section, we formally define BWR-games, canonical forms, and state some useful propositions. In Section 3, we give a sufficient condition for the ergodicity of a BWR-game, which will be used as one possible stopping criterion in our algorithm. We give the algorithm in Section 4.1, and prove it converges in Section 4.2. In Section 5, we show that this convergence proof can, in fact, be turned into a quantitative statement giving a more precise bounds stated in Theorem 1. The last section gives a lower bound example for the algorithm. Two illustrative examples are given in Appendix D. The first one illustrates the existence of a saddle point in pure strategies, while the second one shows, among other things, that local optimality does not imply global optimality, if the game is not in canonical form.

2 Preliminaries

2.1 BWR-games

A BWR-game is defined by the quadruple $\mathcal{G} = (G, P, v_0, r)$, where $G = (V = V_W \cup V_B \cup V_R, E)$ is a digraph that may have loops and multiple arcs, but no terminal vertices[3], i.e., vertices of out-degree

[2]In fact, one can use Theorem 1 to derive the dual LP-formulation for Markov decision processes.

[3]This assumption is without loss of generality since otherwise one can add a loop to each terminal vertex.

0; P is the set of probability distributions for all $v \in V_R$ specifying the probability $p(v,u)$ of a move from v to u; $v_0 \in V$ is an initial position from which the play starts; and $r : E \to \mathbb{R}$ is a local reward function. We assume that $\sum_{u \mid (v,u)\in E} p(v,u) = 1 \quad \forall v \in V_R$. For convenience we will assume that $p(v,u) > 0$ whenever $(v,u) \in E$ and $v \in V_R$, and set $p(v,u) = 0$ for $(v,u) \notin E$.

Standardly, we define a strategy $s_W \in S_W$ (respectively, $s_B \in S_B$) as a mapping that assigns a move $(v,u) \in E$ to each position $v \in V_W$ (respectively, $v \in V_B$). A pair of strategies $s = (s_W, s_B)$ is called a *situation*. Given a BWR-game $\mathcal{G} = (G, P, v_0, r)$ and situation $s = (s_B, s_W)$, we obtain a (weighted) Markov chain $\mathcal{G}_s = (G, P_s, v_0, r)$ with transition matrix P_s in the obvious way:

$$p_s(v,u) = \begin{cases} 1 & \text{if } (v \in V_W \text{ and } u = s_W(v)) \text{ or } (v \in V_B \text{ and } u = s_B(v)); \\ 0 & \text{if } (v \in V_W \text{ and } u \neq s_W(v)) \text{ or } (v \in V_B \text{ and } u \neq s_B(v)); \\ p(v,u) & \text{if } v \in V_R. \end{cases}$$

In the obtained Markov chain $\mathcal{G}_s = (G, P_s, v_0, r)$, we define the limiting (mean) effective payoff $c_s(v_0)$ as

$$c_s(v_0) = \sum_{v\in V} p^*(v) \sum_u p_s(v,u) r(v,u), \tag{3}$$

where $p^* : V \to [0,1]$ is the limiting distribution for $\mathcal{G}_s$ starting from v_0 (see Appendix A for details). Doing this for all possible strategies of Black and White, we obtain a matrix game $C_{v_0} : S_W \times S_B \to \mathbb{R}$, with entries $C_{v_0}(s_W, s_B)$ defined by (3).

2.2 Solvability and ergodicity

It is known that every such game has a saddle point in pure strategies [Gil57, LL69]. Moreover, there are optimal strategies (s_W^*, s_B^*) that do not depend on the starting position v_0, so-called uniformly optimal strategies. In contrast, the value of the game $\mu(v_0) = C_{v_0}(s_W^*, s_B^*)$ may depend on v_0.

The triplet $\mathcal{G} = (G, P, r)$ is called a *un-initialized BWR-game*, and $\mathcal{G}$ is called *ergodic* if the value $\mu(v_0)$ of each corresponding BWR-game (G, P, v_0, r) is the same for all initial positions $v_0 \in V$.

2.3 Potential transforms

Given a BWR-game $\mathcal{G} = (G, P, v_0, r)$, let us introduce a mapping $x : V \to \mathbb{R}$, whose values $x(v)$ will be called *potentials*, and define the transformed reward function $r_x : E \to \mathbb{R}$ as:

$$r_x(v,u) = r(v,u) + x(v) - x(u), \quad \text{where } (v,u) \in E. \tag{4}$$

It is not difficult to verify that the two normal form matrices C_x and C, of the obtained game $\mathcal{G}_x$ and the original game $\mathcal{G}$, are equal (see [BEGM09]). In particular, their optimal (pure positional) strategies coincide, and the values also coincide: $\mu_x(v_0) = \mu(v_0)$.

2.4 Ergodic canonical form

Given a BWR-game $\mathcal{G} = (G, P, r)$, and a potential transformation x, let us define a mapping $m_x : V \to \mathbb{R}$ as follows:

$$m_x(v) = \begin{cases} \max(r_x(v,u) \mid u : (v,u) \in E) & \text{for } v \in V_W, \\ \min(r_x(v,u) \mid u : (v,u) \in E) & \text{for } v \in V_W, \\ \text{mean}(r_x(v,u) \mid u : (v,u) \in E) = \sum_{u \mid (v,u)\in E} r_x(v,u)\, p(v,u) & \text{for } v \in V_R. \end{cases} \tag{5}$$

In the sequel, we shall simply use m and r, whenever it is clear from the context what x is. A move $(v,u) \in E$ in a position $v \in V_W$ (respectively, $v \in V_B$) is called *locally optimal* if it realizes the maximum (respectively, minimum) in (5). A strategy s_W of White (respectively, s_B of Black) is called *locally optimal* if it chooses a locally optimal move $(v,u) \in E$ in every position $v \in V_W$ (respectively, $v \in V_B$).

Definition 1 *We say that a BWR-game $\mathcal{G}$ is in* ergodic canonical form *if function (5) is constant: $m(v) \equiv m$ for some number m.*

Canonical forms were defined for BW-games in [GKK88], and extended to BWR-games in [BEGM09]. It was shown in [GKK88] that there always exists a potential transformation x such that the optimal local rewards $m_x(v)$ in a BW-game are equal to the game's value $\mu(v)$ at each vertex v. This result was extended in [BEGM09] to the BWR-case. The main purpose of this paper is to provide an algorithm for finding such a potential transformation in the ergodic case.

Proposition 1 *If in a BWR-game the function $m(v) = m$ for all $v \in V$, then (i) every locally optimal strategy is optimal and (ii) the game is ergodic: $m = \mu(v_0)$ is its value for every initial position $v_0 \in V$.*

Proof Indeed, if White (Black) applies a locally optimal strategy then after every own move (s)he will get (pay) m, while for each move of the opponent the local reward will be at least (at most) m, and finally, for each random position the expected local reward is m. Thus, every locally optimal strategy of a player is optimal. Furthermore, if both players choose their optimal strategies then the expected local reward b_i equals m for every step i. Hence, the value of the game $\lim_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} b_i$ equals m, too. □

3 Sufficient conditions for ergodicity of BWR-games

A digraph $G = (V = V_W \cup V_B \cup V_R, E)$ is called *ergodic* if any un-initialized BWR-game $\mathcal{G} = (G, P, r)$ on G is ergodic, that is, the values of the games $\mathcal{G} = (G, P, v_0, r)$ do not depend on v_0. We will give a simple characterization of ergodic digraphs, which, obviously, provides a sufficient condition for ergodicity of the BWR-games.

In addition to partition $\Pi_p : V = V_W \cup V_B \cup V_R$, let us consider one more partition $\Pi_r : V = V^W \cup V^B \cup V^R$ with the following properties:

(i) Sets V^W and V^B are not empty (while V^R might be empty).

(ii) There is no arc $(v, u) \in E$ such that ($v \in (V_W \cup V_R) \cap V^B$ and $u \notin V^B$) or, vice versa, ($v \in (V_B \cup V_R) \cap V^W$ and $u \notin V^W$). In other words, White cannot leave V^B, Black cannot leave V^W, and there are no random moves from $V^W \cup V^B$.

(iii) For each $v \in V_W \cap V^W$ (respectively, $v \in V_B \cap V^B$) there is a move $(v, u) \in E$ such that $u \in V^W$ (respectively, $u \in V^B$). In other words, White (Black) cannot be forced to leave V^W (respectively, V^B).

In particular, the properties above imply that the induced subgraphs $G[V^W]$ and $G[V^B]$ have no terminal vertex.

A partition $\Pi_r : V = V^W \cup V^B \cup V^R$ satisfying (i), (ii), and (iii) will be called a *contra-ergodic* partition for digraph $G = (V_W \cup V_B \cup V_R, E)$.

Theorem 2 *A digraph G is ergodic iff it has no contra-ergodic partition.*

Proof *"Only if part"*. Let $\Pi_r : V = V^W \cup V^B \cup V^R$ be a contra-ergodic partition of G. Let us assign arbitrary positive probabilities to random moves such that $\sum_{u \mid (v,u)\in E} p(v, u) = 1$ for all $v \in V_R$. We still have to assign a local reward $r(v, u)$ to each move $(v, u) \in E$. Let us define $r(v, u) = 1$ whenever $v, u \in V^W$, $r(v, u) = -1$ whenever $v, u \in V^B$, and $r(v, u) = 0$ otherwise. Clearly, if the initial position is in V^W (respectively, in V^B) then the value of the obtained game is 1 (respectively, -1). Hence, the corresponding un-initialized game is not ergodic.

"If part". Given an un-initialized and non-ergodic BWR-game $\mathcal{G} = (G, P, r)$, let $\mu(v)$ denote the value of the corresponding initialized game $\mathcal{G} = (G, P, v, r)$ for each initial position $v \in V$. The obtained function $\mu(v)$ is not constant, since $\mathcal{G}$ is not ergodic. Let μ_W and μ_B denote the maximum and minimum values, respectively. Then, let us set $V^W = \{v \in V \mid \mu(v) = \mu_W\}$, $V^B = \{v \in V \mid \mu(v) = \mu_B\}$, and $V^R = V \setminus (V^W \cup V^B)$. It is not difficult to verify that the obtained partition $\Pi_r : V = V^W \cup V^B \cup V^R$ is contra-ergodic. □

The "only if part" can be strengthened as follows:

Definition 2 *A* contra-ergodic decomposition *of $\mathcal{G}$ is a contra-ergodic partition $\Pi_r : V = V^W \cup V^B \cup V^R$ such that $m(v) > m(u)$ for every $v \in V^W$ and $u \in V^B$.*

Proposition 2 *Given a BWR-game $\mathcal{G}$ whose graph has a contra-ergodic partition, if $m(v) > m(u)$ for every $v \in V^W, u \in V^B$ then $\mu(v) > \mu(u)$ for every $v \in V^W, u \in V^B$. In particular, $\mathcal{G}$ is not ergodic.*

Proof Let us choose a number μ such that $m(v) > \mu > m(u)$ for every $v \in V^W$ and $u \in V^B$; it exists, because set V of positions is finite. Obviously, properties (i), (ii), and (iii) imply that White (Black) can guarantee more (less) that μ for every initial position $v \in V^W$ (respectively, $v \in V^B$). Hence, $\mu(v) > \mu > \mu(u)$ for every $v \in V^W$ and $u \in V^B$. □

For example, no contra-ergodic partition can exist if $G = (V_W \cup V_B \cup V_R, E)$ is a complete tripartite digraph; see, for instance, the example in Appendix D, where each part consists of only one vertex.

Remark 1 *A similar criterion was recently obtained [BEGM10] for general stochastic games, not necessarily with perfect information.*

4 Pumping algorithm for the ergodic BWR-games

4.1 Description of the algorithm

Given a BWR-game $\mathcal{G} = (G, P, r)$, let us compute $m(v)$ for all $v \in V$ using (5). Throughout, we will denote by $[m] \stackrel{\text{def}}{=} [m^-, m^+]$ and $[r] \stackrel{\text{def}}{=} [r^-, r^+]$ the range of functions m and r, respectively, and let $M = m^+ - m^-$ and $R = r^+ - r^-$. Given potentials $x : V \to \mathbb{R}$, we denote by m_x the function m in (5) in which r is replaced by the transformed reward r_x. Given a subset $I \subseteq [m]$, let $V(I) = \{v \in V \mid m(v) \in I\} \subseteq V$. In the following algorithm, set I will always be a closed or semi-closed interval within $[m]$.

Let $m^- = t_0 < t_1 < t_2 < t_3 < t_4 = m^+$ be given thresholds. We will successively apply potential transforms $x : V \to \mathbb{R}$ in such a way that no vertex ever leaves the whole interval $[t_0, t_4]$, and no vertex ever leaves the central $[t_1, t_3]$; in other words, $V_x[t_0, t_4] = V[t_0, t_4]$ and $V_x[t_1, t_3] \supseteq V[t_1, t_3]$ for all considered transforms x, where $V_x(I) = \{v \in V \mid m_x(v) \in I\}$.

Let us initialize potentials $x(v) = 0$ for all $v \in V$, and fix

$$t_i = m_x^- + \frac{i}{4}M_x, \quad i = 0, 1, 2, 3, 4, \quad \text{where } M_x = m_x^+ - m_x^-. \tag{6}$$

Then, let us reduce all potentials of $V_x[t_2, t_4]$ by a maximum constant δ such that no vertex leaves the closed interval $[t_1, t_3]$. It can happen that $\delta = +\infty$; in this case, by definition 2, we get a contra-ergodic decomposition such that $V^R = \emptyset$; see line 8 of the procedure.

The above procedure is repeated until one of the sets $V_x[t_0, t_1)$ or $V_x(t_3, t_4]$ becomes empty.

It is clear that δ can be computed in linear time: it is the maximum value δ such that $m_x^\delta(v) \geq t_1$ for all $v \in V_x[t_2, t_4]$ and $m_x^\delta(v) \leq t_3$ for all $v \in V_x[t_0, t_2)$, where $m_x^\delta(v)$ is the new value of $m_x(v)$ after all potentials in $V_x[t_2, t_4]$ have been reduce by δ (see Equations (13) in Appendix B for the exact formula). Note that the change in $m_x(u)$ value is never more than δ.

It is important to note that $\delta \geq M_x/4$. Indeed, vertices from $[t_2, t_4]$ can only go down, while vertices from $[t_0, t_2)$ can only go up. Each of them must traverse a distance of at least $M_x/4$ before it can reach the border of the interval $[t_1, t_3]$. Moreover, if after some iteration one of the sets $V_x[t_0, t_1)$ or $V_x(t_3, t_4]$ becomes empty then the range of m_x is reduced at least by 25%.

Procedure PUMP($\mathcal{G}, \varepsilon$) below tries to reduce any BWR-game $\mathcal{G}$ by a potential transformation x into one in which $M_x \leq \varepsilon$. Two subroutines are used in the procedure. REDUCE-POTENTIALS($\mathcal{G}, x$) replaces the current potential x with another potential with a sufficiently small norm (cf. Lemma 4 below). This reduction is needed since without it the potentials and, hence, the transformed local rewards too, can grow exponentially. The second routine FIND-PARTITION($\mathcal{G}, x$) uses the current potential vector x to construct a contra-ergodic decomposition of $\mathcal{G}$ (cf. line 19 of the algorithm below). We will prove in Lemma 3 that if the number of pumping iterations performed is large enough:

$$N = \frac{8n^2 R_x}{M_x \theta^k} + 1, \tag{7}$$

where $R_x = r_x^+ - r_x^-$, $\theta = \min\{p(v,u) \ : \ (v,u) \in E\}$, and k is the number of random nodes, and yet the range of m_x is not reduced, then we will be able to find a contra-ergodic decomposition.

In Section 4.2, we will first argue that the algorithm terminates in finite time if the considered BWR-game is ergodic. In the following section, this will be turned into a quantitative argument with the precise bound on the running time. Yet, in Section 6, we will show that this time can be exponential already for R-games.

4.2 Proof of finiteness for the ergodic case

To simplify notation, let us assume without loss of generality that range of m is $[0,1]$, that is, $t_i = \frac{i}{4}$, for $i = 0,\dots,4$, and that the initial potential $x^0 = 0$. Suppose that during N iterations no new vertex enters the interval $[\frac{1}{4}, \frac{3}{4}]$. Then, $-x(v) \geq N/4$ for each $v \in V(\frac{3}{4}, 1]$, since these vertices "were pumped" N times, and $x(v) \equiv 0$ for each $v \in V[0, \frac{1}{4})$, since these vertices "were not pumped" at all. We will show that if N is sufficiently large then the considered game is not ergodic.

Consider the infinite sequence of iterations $i = 0, 1, \dots$, and denote by $V^B \subseteq V$ (respectively, by $V^W \subseteq V$) the set of vertices that were pumped just finitely many times (respectively, always but finitely many times); in other words, $m_x(v) \in [\frac{1}{2}, 1]$ if $v \in V^W$ (respectively, $m_x(v) \in [0, \frac{1}{2})$ if $v \in V^B$) for all but finitely many iterations.

Proposition 3 *If V^W and V^B are both non-empty, then Partition $\Pi_r : V = V^W \cup V^B \cup V^R$, where $V^R = V \setminus (V^W \cup V^B)$, is a contra-ergodic decomposition.*

Proof First, let us check properties (i), (ii), (iii) of Section 3. Since (i) is our assumption and (iii) follows from (i), (ii), and the fact that the graph has no terminal vertices, it remains to prove (ii). It is clear that there exist no random or white (respectively, black) moves from V^B (respectively, V^W), because by defintion of V^B and V^W, the potential difference between any vertex of V^B and any vertex of $V \setminus V^B$ (respectively, any vertex of V^W and any vertex of $V \setminus V^W$) is infinite.

Finally, it also from the same definitions that after sufficiently many iterations $m_x(v) > \frac{1}{2}$ for all $v \in V^W$, while $m_x(v) \leq \frac{1}{2}$ for all $v \in V^B$. Hence the claim follows by Proposition 2. □

In other words, our algorithm is finite for the ergodic BWR-games. Below we shall give an upper bound for the number of iterations a vertex can "oscillate" in $[0, \frac{1}{4})$ or $(\frac{3}{4}, 1]$ before it finally enters $[\frac{1}{4}, \frac{3}{4}]$ (to stay there forever).

Remark 2 *Let us note that for general stochastic games (with imperfect information), the value might not exist [Gil57]. Nevertheless, the pumping algorithm can be extended [BEGM10] for the ergodic case. This algorithm either (i) certifies that the game is ε-ergodic, that is, it finds a potential transformation x such that all local values $m_x(v)$ are within an interval of length ϵ, or (ii) presents a contra-ergodic partition as in Section 3; in particular, we obtain two positions U and v whose values differ by more than ϵ, $|\mu(v) - \mu(v)| > \epsilon$.*

5 Running time analysis

Consider the execution of the algorithm on a given BWR-game. We define a *phase* to be a set of iterations during which the range of m_x, defined with respect to the current potential x, is not reduced by a constant factor of what it was at the beginning of the phase, i.e., neither of the sets $V_x[t_0, t_1)$ and $V_x(t_3, t_4]$ becomes empty (cf. line 12 of the procedure). Note that the number of iterations in each phase is at most N defined by (7). Lemma 3 states that if N iterations are performed in a phase then the game is not ergodic. Lemma 10 bounds the total number of phases and estimates the overall running time.

Algorithm 1 PUMP($\mathcal{G}, \varepsilon$)

Input: A BWR-game $\mathcal{G} = (G = (V, E), P, r)$ and a desired accuracy ε

Output: Either a potential $x : V \to \mathbb{R}$ s.t. $|m_x(v) - m_x(u)| \le \varepsilon$ for all $u, v \in V$, or a contra-ergodic decomposition

1: let $x^0(v) := x(v) := 0$ for all $v \in V$; $i := 1$
2: let $t_0, t_1, \ldots, t_4$, and N be as defined by (6) and (7)
3: **while** $i \le N$ **do**
4:   **if** $M_x \le \varepsilon$ **then**
5:     **return** x and **halt**
6:   **end if**
7:   $\delta := \max\{\delta' |\ m_x^{\delta'}(v) \ge t_1$ for all $v \in V_{x^0}[t_2, t_4]$ and $m_x^{\delta'}(v) \le t_3$ for all $v \in V_{x^0}[t_0, t_2)\}$
8:   **if** $\delta = \infty$ **then**
9:     **return** the contra-ergodic partition ($V^W = V_{x^0}[t_0, t_2)$, $V^B = V_{x^0}[t_2, t_4]$, $V^R = \emptyset$)
10:   **end if**
11:   $x(v) := x(v) - \delta$ for all $v \in V_{x^0}[t_2, t_4]$
12:   **if** $V_{x^0}[t_0, t_1) = \emptyset$ or $V_{x^0}(t_3, t_4] = \emptyset$ **then**
13:     $x := x^0$:=REDUCE-POTENTIALS($\mathcal{G}, x$); $i := 1$
14:     recompute the thresholds $t_0, t_1, \ldots, t_4$ and N using (6) and (7)
15:   **else**
16:     $i := i + 1$;
17:   **end if**
18: **end while**
19: (V^W, V^B, V^R):=FIND-PARTITION($\mathcal{G}, x$)
20: **return** the contra-ergodic partition (V^W, V^B, V^R)

5.1 Finding a contra-ergodic decomposition: FIND-PARTITION($\mathcal{G}, x$)

We assume throughout this section that we are inside phase h of the algorithm, which started with a potential x^h, and that $M_x > \frac{3}{4} M_{x^h}$ in all iterations of the phase, and hence we proceed to step 19. For convenience, we will write $(\cdot)_{x^h}$ as $(\cdot)_h$, where $(\cdot)$ could be m, r, r^+, etc (e.g., $m_h^- = m_{x^h}^-$, $m_h^+ = m_{x^h}^+$). For simplicity, we assume that the phase starts with local reward function $r = r_h$ and hence [4] $x^h = 0$. Given a potential vector x, we use the following notation:

$$\mathrm{EXT}_x = \{(v, u) \in E \ :\ v \in V_B \cup V_W \text{ and } r_x(v, u) = m_x(v)\}, \quad \Delta_x = \min\{x(v) : v \in V\}.$$

Let $t_l \le 0$ be the largest value satisfying the following conditions:

(i) there are no arcs $(v, u) \in E$ with $v \in V_W \cup V_R$, $x(v) \ge t_l$ and $x(u) < t_l$;

(ii) there are no arcs $(v, u) \in \mathrm{EXT}_x$ with $v \in V_B$, $x(v) \ge t_l$ and $x(u) < t_l$.

Let $X = \{v \in V \ :\ x(v) \ge t_l\}$. In words, X is the set of nodes with potential as close to 0 as possible, such that no white or random node in X has an arc crossing to $V \setminus X$, and no black node has an extremal arc crossing to $V \setminus X$. Similarly, define $t_u \ge \Delta_x$ to be the smallest value satisfying the following conditions:

(iii) there are no arcs $(v, u) \in E$ with $v \in V_B \cup V_R$, $x(v) \le t_u$ and $x(u) > t_u$;

(iv) there are no arcs $(v, u) \in \mathrm{EXT}_x$ with $v \in V_W$, $x(v) \le t_u$ and $x(u) > t_u$,

and let $Y = \{v \in V \ :\ x(v) \le t_u\}$. Note that the sets X and Y can be computed in $O(|V| \log |V| + |E|)$ time.

Lemma 1 *It holds that* $\max\{-t_l, t_u - \Delta_x\} \le n R_h \left(\frac{1}{\theta}\right)^k$.

[4] in particular, note that $r_x(v, u)$ and $m_x(v)$ are used, for simplicity of notation, to actually mean $r_{x+x^h}(v, u)$ and $m_{x+x^h}(v)$, respectively.

To prove Lemma 1, we need the following lemma.

Lemma 2 *Consider any two nodes $u, v \in E$ and let x be the current potential. Then*

$$x(u) \geq \begin{cases} x(v) - (m_h^+ - r_h^-) & \text{if either } (v \in V_W \text{ and } (v,u) \in E) \text{ or } (v \in V_B \text{ and } (v,u) \in \mathrm{EXT}_x) \\ \frac{x(v)-(m_h^+-r_h^-)}{\theta} & \text{if } v \in V_R \text{ and } (v,u) \in E, \end{cases}$$

and

$$x(u) \leq \begin{cases} x(v) + r_h^+ - m_h^- & \text{if either } (v \in V_B \text{ and } (v,u) \in E) \text{ or } (v \in V_W \text{ and } (v,u) \in \mathrm{EXT}_x) \\ \frac{x(v)+r_h^+-m_h^--(1-\theta)\Delta_x}{\theta} & \text{if } v \in V_R \text{ and } (v,u) \in E. \end{cases}$$

Proof We only consider the case for $v \in V_R$, as the other claims are obvious from the definitions. For the first claim, assume that $x(v) \geq x(u)$, since otherwise there is nothing to prove. Then from $m_x(v) \leq m_h^+$, it follows that

$$\begin{aligned} m_h^+ - r_h^- &\geq m_h^+ - \sum_{u'} p(v,u') r_h(v,u') \\ &\geq p(v,u)(x(v)-x(u)) + \sum_{u' \neq u} p(v,u')(x(v)-x(u')) \\ &\geq \theta(x(v)-x(u)) + x(v)(1-\theta), \end{aligned}$$

from which our claim follows. The other claim can be proved by a similar argument (by replacing $x(\cdot)$ by $\Delta_x - x(\cdot)$ and $m_h^+ - r_h^-$ by $r_h^+ - m_h^-$). □

Proof of Lemma 1. By definition of X, for every node $v \in X$ there must exist (not necessarily distinct) nodes $v_0, v_1, \ldots, v_{2j} = v \in X$, $j \leq |X|$, such that $x(v_0) = 0$, and for $i = 1, 2, \ldots, j$, $x(v_{2i}) \geq x(v_{2i-1})$, and either $((v_{2i-2}, v_{2i-1}) \in E$ and $v_{2i-2} \in V_W \cup V_R)$ or $((v_{2i-2}, v_{2i-1}) \in \mathrm{EXT}_x$ and $v_{2i-2} \in V_B)$. Among the even-numbered nodes, let $v_{2i_1-2}, \ldots, v_{2i_l-2}$ be the ones belonging to V_R, and assume withut loss of generality that $l > 0$ and $i_1 < i_2 < \cdots < i_l$. Using Lemma 2, we obtain the following inequality by a telescoping sum:

$$x(v_{2i_{q+1}-2}) \geq x(v_{2i_q-1}) - (i_{q+1} - i_q - 1)(m_h^+ - r_h^-), \text{ for } q = 1, \ldots, l, \tag{8}$$

and $x(v_{2i_1-2}) \geq -(i_1 - 1)(m_h^+ - r_h^-)$.

Now applying Lemma 2 to the pair $v_{2i_q-2} \in V_R$ and v_{2i_q-1}, for $q = 1, \ldots, l-1$, and using (8) we obtain:

$$x_{q+1} \geq \frac{x_q}{\theta} - \left(\frac{1}{\theta} + i_{q+1} - i_q - 1\right)(m_h^+ - r_h^-), \quad x_1 \geq -(i_1 - 1)(m_h^+ - r_h^-), \tag{9}$$

where we write, for convenience, $x_q = x(v_{2i_q-2})$, for $q = 1, \ldots, l$. Iterating, we obtain:

$$x_l \geq -\left(\frac{i_1 - 1}{\theta^{l-1}} + \sum_{q=2}^{l} \frac{1}{\theta^{l-q}}\left(\frac{1}{\theta} + i_q - i_{q-1} - 1\right)\right)(m_h^+ - r_h^-).$$

Combining this with the inequality $x(v) \geq \frac{x_l}{\theta} - (\frac{1}{\theta} + j - i_l)(m_h^+ - r_h^-)$ and using $\theta < 1$, we get

$$x(v) \geq -\frac{j}{\theta^l}(m_h^+ - r_h^-) \geq -\frac{1}{\theta^k}|X|(m_h^+ - r_h^-).$$

Similarly, one can prove for any $v \in Y$ that $x(v) \leq \Delta_x + \frac{1}{\theta^k}|Y|(r_h^+ - m_h^-)$, and the lemma follows. □

The correctness of the algorithm follows from the following lemma.

Lemma 3 *Suppose that pumping is performed for $N_h \geq 2nT_h + 1$ iterations, where $T_h = \frac{4nR_h}{M_h\theta^k}$, and neither the set $V_h[t_0, t_1)$ nor $V_h(t_3, t_4]$ becomes empty. Let $V^B = X$ and $V^W = Y$ be the sets constructed as above, and $V^R = V \setminus (X \cup Y)$. Then $V^W \cup V^B \cup V^R$ is a contra-ergodic decomposition.*

Proof As noted above (see (13)) we pump in each iteration by $\delta \geq \frac{M_h}{4}$. Furthermore, our formula for δ implies that once a vertex enters the region $V_h[t_1, t_3]$, it will never leave the region. In particular, there are vertices $v_0 \in X \cap V_h[t_0, t_1)$ and $v_n \in Y \cap V_h(t_3, t_4]$ with $x(v_0) = 0$ and $x(v_n) = \Delta_x$.

For a vertex $v \in V$, let $N(v)$ denote the number of times the vertex was pumped. Then $N(v_0) = 0$ and $N(v_n) = N_h$.

We claim that $N(v) \leq T_h$ for any $v \in X$, and $N(v) \geq N_h - T_h$ for all $v \in Y$ (i.e., every vertex in X was *not* pumped in *all* steps but at most T, and every vertex in Y was pumped in *all* steps but at most T_h). Indeed, if $v \in X$ (respectively, $v \in Y$) was pumped greater than (respectively, less than) T_h times then $x(v) - x(v_0) \leq -\frac{nR_h}{\theta^k}$ (respectively, $x(v_n) - x(v) \leq -\frac{nR_h}{\theta^k}$), in contradiction to Lemma 1.

Since $N_h > 2T_h$, it follows that $X \cap Y = \emptyset$. Furthermore, among the first $2nT_h + 1$ iterations, in at most nT_h iterations some vertex $v \in X$ was pumped, and in at most nT_h iterations some vertex in Y was not pumped. Thus, there must exist an iteration at which every vertex $v \in X$ was not pumped and every vertex $v \in Y$ was pumped. At that particular iteration, we must have $X \subseteq V_h[t_0, t_2)$ and $Y \subseteq V_h[t_2, t_4]$, and hence $m_x(v) < t_2$ for every $v \in X$ and $m_x(v) \geq t_2$ for every $v \in Y$. By the way the sets X and Y were constructed and from (13), we can easily see that X and Y will continue to have this property till the end of the N_h iterations, and hence they induce a contra-ergodic partition. The lemma follows. □

5.2 Potential reduction: REDUCE-POTENTIALS($\mathcal{G}, x$)

One problem that arises during the pumping procedure is that the potentials can increase exponentially in the number of phases, making our bounds on the number of iterations per phase also exponential in n. For the BW-case Pisaruk [Pis99] solved this problem by giving a procedure that reduces the range of the potentials after each round, while keeping all its desired properties needed for the running time analysis. (For convenience, we give a variant of this procedure in Appendix C.)

Pisaruk's potential reduction procedure can be thought of as a combinatorial procedure for finding an *extreme point* of a polyhedron, given a point in it. Indeed, given a BWR-game and a potential x, let us assume without loss of generality, by shifting the potentials if necessary, that $x \geq 0$, and let $E' = \{(v, u) \in E \ : \ r_x(v, u) \in [m_x^-, m_x^+], \ v \in V_B \cup V_W\}$, where r is the *original* local reward function. Then the following polyhedron

$$\Gamma_x = \left\{ x' \in \mathbb{R}^V \;\middle|\; \begin{array}{ll} m_x^- \leq r(v,u) + x'(v) - x'(u) \leq m_x^+, & \forall (v,u) \in E' \\ r(v,u) + x'(v) - x'(u) \leq m_x^+, & \forall v \in V_W, \ (v,u) \in E \setminus E' \\ m_x^- \leq r(v,u) + x'(v) - x'(u), & \forall v \in V_B, \ (v,u) \in E \setminus E' \\ m_x^- \leq \sum_{u \in V} p(v,u)(r(v,u) + x'(v) - x'(u)) \leq m_x^+, & \forall v \in V_R \\ x'(v) \geq 0 & \forall v \in V \end{array} \right\}.$$

is non-empty, since $x \in \Gamma_x$. Moreover, Γ_x is pointed, and hence, it must have an extreme point. Let us remark that given a feasible point x, an extreme point can be computed in $O(n^2|E|)$ time (see, e.g., [Sch03]).

Lemma 4 *Consider a BWR-game in which all rewards are integral with range $R = r^+ - r^-$, and probabilities $p(v, u)$ are rational with common denominator W, and let $k = |V_R|$. Then any extreme point x^* of Γ_x satisfies $\|x^*\|_\infty \leq nRk(2W)^k$.*

Proof Consider such an extreme point x^*. Then x^* is uniquely determined by a system of n linearly independent equations chosen from the given inequalities. Thus there exist subsets $V' \subseteq V$, $V_R' \subseteq V_R$ and $E'' \subseteq E$ such that $|V'| + |V_R'| + |E''| = n$, x^* is the unique solution of the subsystem $x'(v) = 0$ for all $v \in V'$, $x'(v) - x'(u) = m_x^* - r(v, u)$ for $(v, u) \in E''$, and $x'(v) - \sum_{u \in V} p(v, u)x'(u) = m_x^* - \sum_{u \in V} p(v, u)r(v, u)$ for $v \in V_R'$, where m_x^* stands for either m_x^- or m_x^+.

Note that all variables $x'(v)$ must appear in this subsystem, and that the underlying undirected graph of the digraph $G' = (V, E'')$ must be a forest (otherwise the subsystem does not uniquely fix x^*, or it is not linearly independent).

Consider first the case $V_R = \emptyset$. For $i \geq 0$, let V_i be the set of vertices of V at (undirected) distance i from V' (observe that i is finite for every vertex). Then we claim by induction on i that $x^*(v) \leq i\gamma$ for all $v \in V_i$, where $\gamma = \max\{m_x^+ - r^-, r^+ - m_x^-\}$. This is trivially true for $i = 0$. So let us assume that it is also true for some $i > 0$. For any $v \in V_{i+1}$, there must exist either an arc (v, u) or an arc (u, v) where $u \in V_{i-1}$. In the former case, we have $x^*(v) = x^*(u) + m_x^* - r(v, u) \leq i\gamma + m_x^+ - r^- \leq (i + 1)\gamma$. In the latter case, we have $x^*(v) = x^*(u) - (m_x^* - r(u, v)) \leq i\gamma + r^+ - m_x^- \leq (i + 1)\gamma$.

Now suppose that $|V_R| > 0$. For each connected component D_l in the forest G', let us fix a node u_l from V' if $D_l \cap V' \neq \emptyset$; otherwise, v_l is chosen arbitrarily. For every node $v \in D_l$ let $\mathcal{P}_v$ be a (not necessary directed) path from v to v_l. Thus, we can write $x'(v)$ uniquely as

$$x'(v) = x'(v_l) + \ell_{v,1} m_x^+ + \ell_{v,2} m_x^- + \sum_{(u',u'') \in \mathcal{P}_v} \ell_{v,u',u''} r(u', u''), \tag{10}$$

for some $\ell_{v,1}, \ell_{v,2} \in \mathbb{Z}$, and $\ell_{v,u',u''} \in \{-1, 1\}$. Thus if $x^*(v_l) = 0$ for some component D_l, then by a similar argument as above, $x^*(v) \leq \gamma |D_l|$ for every $v \in D_l$.

Note that, up to this point, we have used all equations corresponding to arcs in G' and to vertices in V'. The remaining set of $|V_R'|$ equations should uniquely determine the values of the variables in any component which has no node in V'. Substituting the values of $x'(v)$ from (10), for the nodes in any such component, we end-up with a linearly independent system on $k' = |V_R'|$ variables $Ax = b$, where A is a $k' \times k'$ matrix in which eatch entry is at most 1 in absolute value and the sum of each row is at most 2 in absolute value, and $\|b\|_\infty \leq n(R + M_x) \leq 2nR$.

The rest of the argument follows (in a standard way) by Cramer's rule. Indeed, the value of each component in the solution is given by Δ'/Δ, where Δ is the determinant of A and Δ' is the determinant of a matrix obtained by replacing one column of A by b. We upper bound Δ' by $k'\|b\|_\infty \Delta_{max}$, where Δ_{max} is the maximum absolute value of a subdeterminant of A of size $k' - 1$. To bound Δ_{max}, let us consider such a subdeterminant with rows $a_1, \ldots, a_{k'-1}$, and use Hadamard's inequality:

$$\Delta' \leq \prod_{i=1}^{k'-1} \|a_i\| \leq 2^{k'-1},$$

since $\|a_i\|_1 \leq 2$, for all i. To lower bound Δ, we note that $W^{k'}\Delta$ is a non-zero integer, and hence has absolute value at least 1. Combining the above inequalities, the lemma follows. □

Note that any point $x' \in \Gamma_x$ satisfies $M_{x'} \subseteq M_x$, and hence replacing x by x^* does not increase the range of m_x.

5.3 An upper bound on the required accuracy

Consider a BWR-game in which all local rewards are integral in the range $[r^-, r^+]$ with $R = r^+ - r^-$, and all probabilities $p(v, u)$, for $(v, u) \in E$ and $v \in V_R$, are rational numbers with least common denominator W. Fix an arbitrary situation s and a starting vertex w, and let respectively P_s^* and $p_s^* = e_w P_s^*$ be the limiting transition matrix and distribution corresponding to s. Using the notation in Appendix A, we let C_i, $i \in J$ be the absorbing classes and $T = V \setminus (\cup_{i \in J} C_i)$ be the set of transient nodes. Let $k_i = |V_R \cap C_i|$ for $i \in J$, and $k_T = |V_R \cap T|$. By definition $k = \sum_{i \in J} k_i + k_T$. For $i \in J$, let $y_i \in [0, 1]^T$ be the absorbing probability vector into class C_i.

Lemma 5 *For any $i \in J$ and $v \in T$, $y_i(v)$ are rational with denominator $\rho_{iv} W^{k_T} \leq (2W)^{k_T}$, where $\rho_{iv} \in \mathbb{Z}$.*

Proof Consider the system of equations defining y_i: $(I - P')y_i = p_i'$ for some $i \in J$, where $P' = P_s[T; T]$ and $p_i' = P_s[T; C_i]e$ (see Appendix A for details). As in the proof of Lemma 4, the idea is to eliminate the variables corresponding to black and white nodes and get a system only on random nodes. Let $E' = \{(v, u) \in E \ : \ v \in V_B \cup V_W$ and $P'(v, u) = 1\}$ and $G' = (V, E')$. Then by linear independence of the system (recall (L3)), G' is a forest. For a node $v \in V_B \cup V_W$, the equation is $y_i(v) = y_i(u) + p_i'(v)$. In particular, if for each connected component D_l in the forest G', we fix a node v_l arbitrarily, we can express $y_i(v)$, for every node $v \in D_l$, uniquely as

$y_i(v) = y_i(v_l) + \sum_{(u,u')\in\mathcal{P}_v} \ell_{v,u,u'} p'_i(u)$, for some $\ell_{v,u,u'} \in \{-1,1\}$, where $\mathcal{P}_v$ is the path between v to v_l.

Substituting these values of $y_i(v)$ in the remaining $k' \leq k_T$ equations $y_i(v) = \sum_u P'(v,u)y_i(u) + p'_i(v)$ for $v \in V_R$, we end-up with a linearly independent system on $k' = |V'|$ variables: $Ax = b$, where A is a $k' \times k'$ matrix in which each entry a_{vu} is the rational α_{vu}/W with $\alpha_{vu} \in \mathbb{Z}$ and $|\alpha_{vu}| \leq W$ and each row sums up to at most 2 in absolute value, and $b(v) = \beta_v/W$ with $\beta_v \in \mathbb{Z}$ and $|b(v)| \leq 2n$.

The rest of the argument follows as in Lemma 4. The value of each component in the solution is given by Δ'/Δ, where Δ is the determinant of A and Δ' is the determinant of a matrix obtained by replacing one column of A by b. The lemma follows by observing that $\Delta \leq 2^{k'}$ and that both $W^{k'}\Delta$, and $W^{k'}\Delta'$ are integral. □

For a situation s (that is a pair of strategies), let $\kappa(s)$ be the number of absorbing classes with only black and white nodes (i.e., directed cycles with no random node).

Lemma 6 *The numbers* $\{p^*_s(v) \; : \; v \in V\}$ *are rational with common denominator at most* $2^{k(k+\kappa(s))}k^k n^{k+\kappa(s)} W^{k+1}$.

Proof Let $J' (\subseteq J)$ be the set of indices i such that $C_i \cap V_R = \emptyset$. Note by definition that $\kappa(s) = |J'|$, and $|J \setminus J'| \leq k$. Consider any absorbing class C_i. Let $E' = \{(v,u) \in E \; : \; v,u \in C_i \text{ and } P_s(v,u) > 0\}$ and $G' = (V,E')$. Assume first that $i \in J'$. Then $p^*_s(v) = y_i(w)/|C_i|$ for all $v \in C_i$, where w is the starting vertex and $y_i(w)$ is the absorbing probability of w into C_i (so $y_i(w) = 1$ if $w \in C_i$). It follows from Lemma 5 that $\{p^*_s(v) \; : \; v \in C_i\}$ are rational with common denominator $M_i = \rho_i W^{k_T} \leq 2^{k_T}|C_i|W^{k_T}$, where $\rho_i \in \mathbb{Z}$.

Now consider the case $i \in J \setminus J'$. For $v \in C_i$, let $\mathcal{R}(v)$ be the set of vertices that can reach v by a directed path in G', all whose internal nodes (if any) are in $V_B \cup V_W$. Note that $\mathcal{R}(v) \cap V_R \neq \emptyset$ for each $v \in (V_B \cup V_W) \cap C_i$. Otherwise, there is a node $v \in (V_B \cup V_W) \cap C_i$ such that no random node in C_i can reach v, and hence the whole component C_i must be a cycle of black and white nodes only.

Consider the system of equations in (L5) in Appendix A defining $p^*_s(v)$:

$$\pi(v) = \sum_{u \in (V_B \cup V_W) \cap C_i} \pi(u) + \sum_{u \in V_R \cap C_i} p(u,v)\pi(u),$$

for $v \in C_i$, and $\sum_{v \in C_i} \pi(v) = y_i(w)$. Eliminating the variables $\pi(v)$ for $v \in (V_B \cup V_W) \cap C_i$, we end-up with a system on only random nodes $v \in V_R \cap C_i$: $\pi(v) = \sum_{u \in \mathcal{R}(v) \cap V_R} p'(u,v)\pi(u)$, where $p'(u,v) = p(u,v) + \sum_{u' \in \mathcal{R}(v) \cap (V_B \cup V_W)} p(u,u')$ (note that $\sum_{u \in V_R \cap C_i} p'(v,u) = 1$ for all $v \in V_R \cap C_i$). Similarly, we can reduce the normalization equation to $\sum_{v \in V_R \cap C_i}(1 + \sum_{u \in V_B \cup V_W} p'(v,u))\pi(v) = y_i(w)$. This gives a system on $k_i = |C_i \cap V_R|$ variables of the form $(I - P')\pi = 0$, $b\pi = y_i(w)$, where the matrix P' and the vector b have rational entries with common denominator W, each column of P' sums up to at most 1, and $\|b\|_1 \leq k_i|C_i|$. Any non-zero component $p^*_s(v)$ in the solution of this system takes the form $y_i(w)\frac{\Delta_0}{\sum_{i=1}^{k_i} b_i \Delta_i}$, where $\Delta_0, \Delta_1, \ldots, \Delta_{k_i}$ are subdeterminants of $I - P'$ of rank $k_i - 1$. Using $\Delta_i \leq 2^{k_i - 1}$ and Lemma 5, we get that all $p^*_s(v)$, for $v \in V_R \cap C_i$ are rational with common denominator $\rho'_i W^{k_i + k_T} \leq 2^{k_i + k_T - 1} k_i |C_i| W^{k_i + k_T}$, where $\rho'_i \in \mathbb{Z}$.

After solving this system, we can get the value of $p^*_s(v)$, for $v \in (V_W \cap V_B) \cap C_i$ from the equation: $\pi(v) = \sum_{u \in \mathcal{R}(v) \cap V_R} p'(u,v)\pi(u)$. In summery, we get rational numbers with common denominator $M_i = \rho_i W^{k_i + k_T + 1} \leq 2^{k_i + k_T - 1} k_i |C_i| W^{k_i + k_T + 1}$, where $\rho_i \in \mathbb{Z}$.

It follows that all components of p^*_s are rational with common denominator at most:

$$\prod_{i \in J} \rho_i W^{k+1} \leq 2^{k(k+\kappa(s))} k^k n^{k+\kappa(s)} W^{k+1}. \tag{11}$$

□

The following example shows that the bound in Lemma 6 cannot be made, in general, to depend exponentially only on the number of random nodes, if we consider an *arbitrary* situation s. Consider a graph with one random node w and κ disjoint directed cycles $C_1, \ldots, C_\kappa$ of sizes $n_i = |C_i|$ which are *relatively prime*. There is an arc from w to one arbitrary node in each cycle with transition probability $1/\kappa$, and the total reward of each cycle is 1. Then the limiting distribution starting from w is $p_s^*(w) = 0$ and $p_s^*(v) = 1/n_i$ and the value at node w is given by $\frac{1}{\kappa}\sum_{i=1}^{\kappa}\frac{1}{n_i}$. This has a denominator superpolynomial in $n = \sum_{i=1}^{\kappa} n_i + 1$.

In view of this example, in order to get stronger bounds, one needs to consider special situations, namely, those that can be returned by our pumping algorithm. Let s be such a situation. Then for every $v \in V_B \cup V_W$, the arc $(v, s(v))$ selected by s is extremal, i.e., $m_x(v) = r_x(v, s(v))$ given the current potential x. We call such a situation *extremal* with respect to x.

Lemma 7 *Suppose that s is an extremal situation with respect to a potential vector x, such that C and C' are absorbing classes with only black and white nodes (i.e., $C, C' \subseteq V_B \cup V_W$ and hence directed cycles). If $M_x < \frac{1}{n^2}$, then $\frac{r(C)}{|C|} = \frac{r(C')}{|C'|}$, where $r(C)$ and $r(C')$ are the local rewards of the cycles C and C', respectively (that is, $r(C) = \sum_{v\in C} r(v, s(v))$).*

Proof Since s is extremal, we have $r(C) = r_x(C) = \sum_{v\in C} m_x(v) \in [m_x^-|C|, m_x^+|C|]$, and hence $r(C)/|C| \in [m_x^-, m_x^+]$. Similarly, $r(C')/|C'| \in [m_x^-, m_x^+]$. Thus, if $r(C)/|C| \neq r(C')/|C'|$, we must have

$$\frac{1}{n^2} \leq \left|\frac{|C'|r(C) - |C|r(C')}{|C||C'|}\right| = \left|\frac{r(C)}{|C|} - \frac{r(C')}{|C'|}\right| \leq m_x^+ - m_x^- < \frac{1}{n^2},$$

which is a contradiction. □

Lemma 8 *Let s be an extremal situation with respect to some potential x such that $M_x < \frac{1}{n^2}$. Then c_s (the expected limiting payoff corresponding to situation s) is rational with denominator at most $2^{k(k+1)}k^k n^{k+1} W^{k+2}$.*

Proof Let $C_1, \ldots, C_{\kappa(s)}$ be the absorbing classes with no random nodes, in the Markov chain defined by s, and let $V' \subseteq V \setminus (\cup_{i=1}^{\kappa(s)} C_i)$ be the set of nodes in the remaining absorbing classes. Since s is extremal, Lemma 7 implies that either $\kappa(s) \leq 1$ or $r(C_i)/|C_i|$ is some constant γ for all $i = 1, \ldots, \kappa(s)$. In the former case, $c(s) = p_s^* Q_s r = \sum_{v\in V} p_s^*(v)\sum_u p(v,u)r(v,u)$ is rational with denominator at most $2^{k(k+1)}k^k n^{k+1} W^{k+2}$ by Lemma 6, since $\sum_u p(v,u)r(v,u)$ brings in another $1/W$ factor. Here Q_s denotes the vertex-arc transition matrix with respect to s.

In the latter case, we get

$$c(s) = \sum_{i=1}^{\kappa(s)} y_i(w)\frac{r(C_i)}{|C_i|} + \sum_{v\in V'} p_s^*(v)\sum_u p(v,u)r(v,u) = \gamma\sum_{i=1}^{\kappa(s)} y_i(w) + \sum_{v\in V'} p_s^*(v)\sum_u p(v,u)r(v,u),$$

which by a similar argument as in the proof of Lemma 6 yields a denominator as stated above. □

The next lemma shows that it is enough in our pumping algorithm to take

$$\varepsilon = \frac{1}{2^{2k(k+1)}k^{2k}n^{2(k+1)}W^{2(k+2)}}. \tag{12}$$

Lemma 9 *Assume that a given BWR-game is ergodic, and let ε be as defined in (12). Consider two situations s and s' such that s is optimal, s' is extremal with respect to some potential x for which $M_x \leq \varepsilon$. Then $c_s = c_{s'}$.*

Proof Since both s an s' are extremal with respect to some potentials, and $\varepsilon < \frac{1}{n^2}$, Lemma 8 implies that both $c(s)$ and $c(s')$ are rational with denominator at most $d = 2^{k(k+1)}k^k n^{k+1} W^{k+2}$. It follows that $|c(s) - c(s')|$ is rational with denominator less than d^2, and hence $c(s) \neq c(s')$ would imply that $|c(s) - c(s')| > \frac{1}{d^2} = \varepsilon$, a contradiction. □

5.4 Proof of Theorem 1

Consider a BWR-game $\mathcal{G} = (G = (V, E), P, r)$ with $|V| = n$ vertices and k random nodes. Assume r to be integral in the range of size R and all transition probabilities are rational with common denominator W. From Lemmas 3 and 4, we can conclude the following bound.

Lemma 10 *Procedure PUMP($\mathcal{G}, \varepsilon$) terminates in $O(n^2|E|R(nk(2W)^k\frac{1}{\varepsilon} + \log R)$ time.*

Proof We note the following:

1. By (7), the number of iterations per phase h is at most $N_h = \frac{8n^2R_h}{M_h\theta^k} + 1$.
2. Each iteration requires $O(|E|)$ time, and the end of a phase we need additional $O(n^2|E|)$ time (which is required for REDUCE-POTENTIALS).
3. By Lemma 4, for any $(v, u) \in E$, we have $r_x(v, u) = r(v, u)+x(v)-x(u) \leq r(v, u)+2nk(2W)^kR$, and similarly, $r_x(v, u) \geq r(v, u) - 2nk(2W)^kR$. In particular, $R_h \leq (1 + 4nk(2W)^k)R$ at the beginning of each phase h in the procedure.

Since $M_h \leq \frac{3}{4}M_{h-1}$ for $h = 1, 2, \ldots$, the maximum number of such phases until we reach the required accuracy is at most $H = \log_{4/3}\left(\frac{M_0}{\varepsilon}\right)$. Putting all the above together, we get that the total running time is at most

$$(1 + 4nk(2W)^k)R\sum_{h=0}^{H}\frac{8n^2|E|}{M_h} + O(n^2|E|)H.$$

Noting that $M_0 \leq R$ and $M_H \geq \varepsilon$, the lemma follows. □

Theorem 1 follows by setting ε sufficiently small:

Corollary 1 *When procedure PUMP($\mathcal{G}, \varepsilon$) is run with ε as in (2), it either outputs a potential vector x such $m_x(v)$ is constant for all $v \in V$, or finds a contra-ergodic partition. The total running time is $(2^knW)^{O(k)}R\log R$.*

Proof The correctness of the algorithm follows from Lemmas 3 and 9. So we need to argue only only about the running time. As noted in the previous section, we need only to go down to an accuracy of $m_h^+ - m_h^- \leq \varepsilon$ defined as in (12) to get the exact solution. The statement then follows from Lemma 10 after substituting ε as in (12). □

6 Lower bound

We show now that the execution time of the algorithm, in the worst case, can be exponential in the number of random nodes k, already for weighted Markov chains, that is, for R-games. Consider the following example. Let $G = (V, E)$ be a digraph on $k = 2l + 1$ vertices $u_l, \ldots, u_1, u_0 = v_0, v_1, \ldots, v_l$, and with the following set of arcs:

$$E = \{(u_l, u_l), (v_l, v_l)\} \cup \{(u_{i-1}, u_i), (u_i, u_{i-1}), (v_{i-1}, v_i), (v_i, v_{i-1}) : \ i = 1, \ldots, l\}.$$

Let $W \geq 1$ be an integer. All nodes are random with the following transition probabilities: $p(u_l, u_l) = p(v_l, v_l) = 1 - \frac{1}{W+1}$, $p(u_0, u_1) = p(u_0, v_1) = \frac{1}{2}$, $p(u_{i-1}, u_i) = p(v_{i-1}, v_i) = 1 - \frac{1}{W+1}$, for $i = 2, \ldots, l$, and $p(u_i, u_{i-1}) = p(v_i, v_{i-1}) = \frac{1}{W+1}$, for $i = 1, \ldots, l$. The local rewards are zero on every arc, except for $r(u_l, u_l) = -r(v_l, v_l) = 1$ (See Figure 1 for $l = 4$). Clearly this Markov chain consists of a single recurrent class, and it is easy to verify that the limiting distribution p^* is as follows:

$$p^*(u_0) \quad = \quad \frac{W-1}{(W+1)W^l - 2}, \ p^*(u_i) = p^*(v_i) = \frac{W^{i-1}(W^2-1)}{2((W+1)W^l - 2)} \text{ for } i = 1, \ldots, l.$$

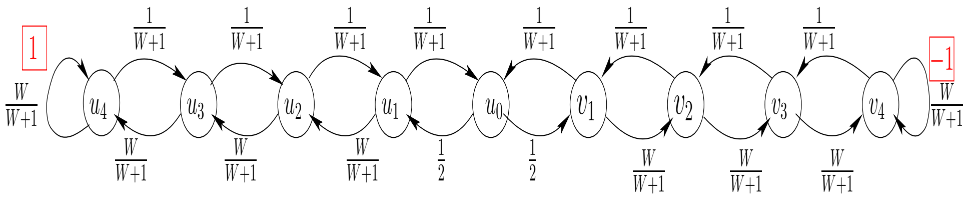

Figure 1: An exponential time example.

The optimal expected reward at each vertex is

$$\mu(u_i) = \mu(v_i) = -1 \cdot (1 - \frac{1}{W+1})p^*(u_l) + 1 \cdot (1 - \frac{1}{W+1})p^*(u_l) = 0,$$

for $i = 0, \ldots, l$. Up to a shift, there is a unique set of potentials x that transform the Markov chain into the canonical form, and they satisfy the following system of equations:

$$\begin{aligned}
0 &= -\frac{1}{2}\Delta_1 - \frac{1}{2}\Delta'_1, \\
0 &= -(1 - \frac{1}{W+1})\Delta_{i+1} + \frac{1}{W+1}\Delta_i, \text{ for } i = 1, \ldots, k-1 \\
0 &= -(1 - \frac{1}{W+1})\Delta'_{i+1} + \frac{1}{W+1}\Delta'_i, \text{ for } i = 1, \ldots, k-1 \\
0 &= -(1 - \frac{1}{W+1}) + \frac{1}{W+1}\Delta_l, \\
0 &= 1 - \frac{1}{W+1} + \frac{1}{W+1}\Delta'_l,
\end{aligned}$$

where $\Delta_i = x(u_i) - x(u_{i-1})$ and $\Delta'_i = x(v_i) - x(v_{i-1})$; by solving the system, we get $\Delta_i = -\Delta'_i = W^{k-i+1}$, for $i = 1, \ldots, l$.

Lower bound on pumping algorithms. Any pumping algorithm that starts with 0 potentials and modifies the potentials in each iteration by at most γ will not have a number of iterations less than $\frac{W^{l-1}}{2\gamma}$ on the above example. In particular, the algorithm in Section 4 has $\gamma \leq 1/\min\{p(v,u) : (v,u) \in E,\ p(v,u) \neq 0\}$, which is $\Omega(W)$ in our example. We conclude that the running time of the algorithm is $\Omega(W^{l-2}) = W^{\Omega(k)}$ on this example.

References

[AM09] D. Andersson and P. B. Miltersen. The complexity of solving stochastic games on graphs. In *ISAAC*, pages 112–121, 2009.

[BEGM09] E. Boros, K. Elbassioni, V. Gurvich, and K. Makino. Every stochastic game with perfect information admits a canonical form. RRR-09-2009, RUTCOR, Rutgers University, 2009.

[BEGM10] E. Boros, K. Elbassioni, V. Gurvich, and K. Makino. A potential reduction algorithm for limit average min-payoff ergodic stochastic games. RRR-05-2010, RUTCOR, Rutgers University, 2010.

[BG09] E. Boros and V. Gurvich. Why chess and back gammon can be solved in pure positional uniformly optimal strategies? RRR-21-2009, RUTCOR, Rutgers University, 2009.

[BSV04] H. Björklund, S. Sandberg, and S. Vorobyov. A combinatorial strongly sub-exponential strategy improvement algorithm for mean payoff games. DIMACS Technical Report 2004-05, DIMACS, Rutgers University, 2004.

[BV01a] E. Beffara and S. Vorobyov. Adapting Gurvich-Karzanov-Khachiyan's algorithm for parity games: Implementation and experimentation. Technical Report 2001-020, Department of Information Technology, Uppsala University, available at: https://www.it.uu.se/research/reports/#2001, 2001.

[BV01b] E. Beffara and S. Vorobyov. Is randomized Gurvich-Karzanov-Khachiyan's algorithm for parity games polynomial? Technical Report 2001-025, Department of Information Technology, Uppsala University, available at: https://www.it.uu.se/research/reports/#2001, 2001.

[BV05] H. Björklund and S. Vorobyov. Combinatorial structure and randomized subexponential algorithms for infinite games. *Theoretical Computer Science*, 349(3):347–360, 2005.

[BV07] H. Björklund and S. Vorobyov. A combinatorial strongly sub-exponential strategy improvement algorithm for mean payoff games. *Discrete Applied Mathematics*, 155(2):210–229, 2007.

[CH08] K. Chatterjee and T. A. Henzinger. Reduction of stochastic parity to stochastic mean-payoff games. *Inf. Process. Lett.*, 106(1):1–7, 2008.

[CJH04] K. Chatterjee, M. Jurdziński, and T. A. Henzinger. Quantitative stochastic parity games. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 121–130, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[Con92] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.

[Con93] A. Condon. An algorithm for simple stochastic games. In *Advances in computational complexity theory, volume 13 of DIMACS series in discrete mathematics and theoretical computer science*, 1993.

[DG06] V. Dhingra and S. Gaubert. How to solve large scale deterministic games with mean payoff by policy iteration. In *valuetools '06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, page 12, New York, NY, USA, 2006. ACM.

[EM79] A. Eherenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109–113, 1979.

[Fed80] A. Federgruen. Successive approximation methods in undiscounted stochastic games. *Operations. Research*, 1:794–810, 1980.

[Fri09] O. Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. *Logic in Computer Science, Symposium on*, 0:145–156, 2009.

[GH08] H. Gimbert and F. Horn. Simple stochastic games with few random vertices are easy to solve. In *FoSSaCS*, pages 5–19, 2008.

[Gil57] D. Gillette. Stochastic games with zero stop probabilities. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contribution to the Theory of Games III, in Annals of Mathematics Studies*, volume 39, pages 179–187. Princeton University Press, 1957.

[GKK88] V. Gurvich, A. Karzanov, and L. Khachiyan. Cyclic games and an algorithm to find minimax cycle means in directed graphs. *USSR Computational Mathematics and Mathematical Physics*, 28:85–91, 1988.

[Hal07] N. Halman. Simple stochastic games, parity games, mean payoff games and discounted payoff games are all LP-type problems. *Algorithmica*, 49(1):37–50, 2007.

[HK66] A. J. Hoffman and R. M. Karp. On nonterminating stochastic games. *Management Science, Series A*, 12(5):359–370, 1966.

[HL31] G. H. Hardy and J. E. Littlewood. Notes on the theory of series (xvi): two tauberian theorems. *J. of London Mathematical Society*, 6:281–286, 1931.

[JPZ06] M. Jurdziński, M. Paterson, and U. Zwick. A deterministic subexponential algorithm for solving parity games. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 117–123, New York, NY, USA, 2006. ACM.

[Jur98] M. Jurdziński. Deciding the winner in parity games is in UP ∩ co-UP. *Inf. Process. Lett.*, 68(3):119–124, 1998.

[Jur00] M. Jurdziński. *Games for Verification: Algorithmic Issues*. PhD thesis, Faculty of Science, University of Aarhus, USA, 2000.

[Kar78] R. M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Math.*, 23:309–311, 1978.

[KL93] A. V. Karzanov and V. N. Lebedev. Cyclical games with prohibition. *Mathematical Programming*, 60:277–293, 1993.

[KMMS03] D. Kratsch, R. M. McConnell, K. Mehlhorn, and J. P. Spinrad. Certifying algorithms for recognizing interval graphs and permutation graphs. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 158–167, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[KS63] J. G. Kemeny and J. L. Snell. *Finite Markov chains*. Springer-Verlag, New York, 1963.

[Lit96] M. L. Littman. *Algorithm for sequential decision making, CS-96-09*. PhD thesis, Dept. of Computer Science, Brown Univ., USA, 1996.

[LL69] T. M. Liggett and S. A. Lippman. Stochastic games with perfect information and time-average payoff. *SIAM Review*, 4:604–607, 1969.

[MO70] H. Mine and S. Osaki. *Markovian decision process*. American Elsevier Publishing Co., New York, 1970.

[Mou76a] H. Moulin. Extension of two person zero sum games. *Journal of Mathematical Analysis and Application*, 5(2):490–507, 1976.

[Mou76b] H. Moulin. Prolongement des jeux à deux joueurs de somme nulle. *Bull. Soc. Math. France, Memoire*, 45, 1976.

[Pis99] N. N. Pisaruk. Mean cost cyclical games. *Mathematics of Operations Research*, 24(4):817–828, 1999.

[RF91] T. E. S. Raghavan and J. A. Filar. Algorithms for stochastic games a survey. *Mathematical Methods of Operations Research*, 35(6):437–472, 1991.

[Sch03] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency, Algorithms and Combinatorics*, volume 24. Springer, New York, 2003.

[VJ00] J. Vöge and M. Jurdzinski. A discrete strategy improvement algorithm for solving parity games. In *CAV '00: Proceedings of the 12th International Conference on Computer Aided Verification*, pages 202–215, London, UK, 2000. Springer-Verlag.

[Vor08] S. Vorobyov. Cyclic games and linear programming. *Discrete Applied Mathematics*, 156(11):2195–2231, 2008.

[VTRF83] O. J. Vrieze, S. H. Tijs, T. E. S. Raghavan, and J. A. Filar. A finite algorithm for the switching control stochastic game. *OR Spectrum*, 5(1):15–24, 1983.

[ZP96] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1-2):343–359, 1996.

Appendix A: Preliminaries on Markov chains

Effective limiting payoffs in weighted Markov chains

When $V_B = V_W = \emptyset$, and hence $V = V_R$ consists only of random nodes, we obtain a *weighted Markov chain*. In this case $P : V \times V \to [0,1]$ is the *transition matrix* whose entry $p(v,u)$ is the probability of transition from v to u in one move, for every pair of positions $v, u \in V$. Then, it is obvious and well-known that for every integral $i \geq 0$ matrix $P^i : V \times V \to [0,1]$ (the i-th power of P) is the i-move transition matrix, whose entry $p_i(v,u)$ is the probability of transition from v to u in exactly i moves, for every $v, u \in V$.

Let $q_i(v,u)$ be the probability that arc $(v,u) \in E$ will be the $(i+1)$-st move, given the original distribution $p_0 = e_{v_0}$, where $i = 0, 1, 2, \ldots$ and e_{v_0} is the n-dimensional unit vector with 1 in position v_0, and denote by $q_i : E \to [0,1]$ the corresponding probabilistic $|E|$-vector. For convenience, we introduce $|V| \times |E|$ vertex-arc transition matrix $Q : V \times E \to [0,1]$ whose entry $q(\ell, (v,u))$ is equal to $p(v,u)$ if $\ell = v$ and 0 otherwise, for every $\ell \in V$ and $(v,u) \in E$. Then, it is clear that $q_i = p_0 P^i Q$.

Let r be the $|E|$-dimensional vector of local rewards, and b_i denote the expected reward for the $(i+1)$-st move; $i = 0, 1, 2, \ldots$, i.e., $b_i = \sum_{(v,u)\in E} q_i(v,u) r(v,u) = p_0 P^i Q r$. Then the *effective payoff* of the weighted Markov chain is defined to be the average expected reward on the limit, i.e., $c = \lim_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} b_i$. It is well-known (see e.g. [MO70]) that this is equal to $c = p_0 P^* Q r$, where P^* is the *limit Markov matrix* (see next subsection).

Limiting distribution of Markov chains

Let $(G = (V,E), P)$ be a Markov chain, and let $C_1, \ldots, C_k \subseteq V$ be the vertex sets of the strongly connected components (classes) of G. For $i \neq j$, let us (standardly) write $C_i \prec C_j$ if there is an arc $(v,u) \in E$ such that $v \in C_i$ and $u \in C_j$. The components C_i, such that there is no C_j with $C_i \prec C_j$ are called the *absorbing* (or *recurrent*) classes, while the other components are called *transient* or *non-recurrent*. Let $J = \{i : \ C_i \text{ is absorbing}\}$, $A = \cup_{i\in J} C_i$, and $T = V \setminus A$. For $X, Y \subseteq V$, a matrix $H \subseteq \mathbb{R}^{V\times V}$, a vector $h \subseteq \mathbb{R}^V$, we denote by $H[X;Y]$ the submatrix of H induced by X as rows and Y as columns, and by $h[X]$ the subvector of h induced by X. Let $I = I[V;V]$ be the $|V| \times |V|$ identity matrix, $e = e[V]$ be the vector of all ones of dimension $|V|$. For simplicity, we drop the indices of $I[\cdot,\cdot]$ and $e[\cdot]$, when they are understood from the context. Then $P[C_i;C_j] = 0$ if $C_j \prec C_i$, and hence in particular, $P[C_i;C_i]e = e$ for all $i \in J$, while $P[T,T]e$ has at least one component of value strictly less than 1.

The following are well-known facts about P^i and the limiting distribution $p_w = e_w P^*$, when the initial distribution is the wth unit vector e_w of dimension $|V|$ (see, e.g., [KS63]):

(L1) $p_w[A] > 0$ and $p_w[T] = 0$;

(L2) $\lim_{i\to\infty} P^i[V;T] = 0$;

(L3) $\text{rank}(I - P[C_i;C_i]) = |C_i| - 1$ for all $i \in J$, $\text{rank}(I - P[T;T]) = |T|$, and $(I - P[T;T])^{-1} = \sum_{i=0}^{\infty} P[T;T]^i$;

(L4) the absorption probabilities $y_i \in [0,1]^V$ into a class C_i, $i \in J$, are given by the unique solution of the linear system: $(I - P[T;T])y_i[T] = P[T;C_i]e$, $y_i[C_i] = e$ and $y_i[C_j] = 0$ for $j \in J$ with $j \neq i$;

(L5) the limiting distribution $p_w \in [0,1]^V$ is given by the unique solution of the linear system: $p_w[C_i](I - P[C_i;C_i]) = 0$, $p_w[C_i]e = y_i(w)$, for all $i \in J$, and $p_w[T] = 0$.

Appendix B: Formula for $m_x^\delta(\cdot)$

Recall that in every iteration of the algorithm in Section 4.1, we reduce the potentials of all nodes in $V_x[t_2, t_4]$ by δ. For a vertex v, the new value of $m_x(v)$ is given by the following formula:

$$
\begin{array}{lll}
\max\left\{\max\limits_{(v,u)\in E, u\in V_x[t_2,t_4]}\{r_x(v,u)\}, \max\limits_{(v,u)\in E, u\in V_x[t_0,t_2)}\{r_x(v,u)\}-\delta\right\} & \text{for} & v\in V_W\cap V_x[t_2,t_4],\\
\min\left\{\min\limits_{(v,u)\in E, u\in V_x[t_2,t_4]}\{r_x(v,u)\}, \min\limits_{(v,u)\in E, u\in V_x[t_0,t_2)}\{r_x(v,u)\}-\delta\right\} & \text{for} & v\in V_B\cap V_x[t_2,t_4],\\
\sum\limits_{(v,u)\in E, u\in V} p(v,u)r_x(v,u)-\delta \sum\limits_{(v,u)\in E, u\in V_x[t_0,t_2)} p(v,u) & \text{for} & v\in V_R\cap V_x[t_2,t_4],\\
\max\left\{\max\limits_{(v,u)\in E, u\in V_x[t_2,t_4]}\{r_x(v,u)\}+\delta, \max\limits_{(v,u)\in E, u\in V_x[t_0,t_2)}\{r_x(v,u)\}\right\} & \text{for} & v\in V_W\cap V_x[t_0,t_2),\\
\min\left\{\min\limits_{(v,u)\in E, u\in V_x[t_2,t_4]}\{r_x(v,u)\}+\delta, \min\limits_{(v,u)\in E, u\in V_x[t_0,t_2)}\{r_x(v,u)\}\right\} & \text{for} & v\in V_B\cap V_x[t_0,t_2),\\
\sum\limits_{(v,u)\in E, u\in V} p(v,u)r_x(v,u)+\delta \sum\limits_{(v,u)\in E, u\in V_x[t_2,t_4]} p(v,u) & \text{for} & v\in V_R\cap V_x[t_0,t_2).
\end{array}
\tag{13}
$$

Appendix C: Potential reduction for BW-games

For completeness, we give here a version of the potential reduction procedure of Pisaruk [Pis99] for BW-games.

Algorithm 2 REDUCE-BW($\mathcal{G}, x$)

Input: A BW game $\mathcal{G} = (G = (V,E), P, r)$ and a current set of potentials $x : V \to \mathbb{R}$
Output: A reduced potential x'

```
X := emptyset; x' := x
while X ≠ V do
    ε1 := min{m_x^+ − r_x'(v,u) : v ∈ X, u ∈ V \ X, (v,u) ∈ E, r_x'(v,u) ≤ m_x^+}
    ε2 := min{r_x'(v,u) − m_x^− : v ∈ V \ X, u ∈ X, (v,u) ∈ E, r_x'(v,u) ≥ m_x^−}
    ε3 := min{x'(v) : v ∈ V \ X}
    x'(v) := x'(v) − min{ε1, ε2, ε3} for all v ∈ V \ X
    while ∃v ∈ V \ X such that x'(v) ≤ ((m_x^+ − m_x^−)/2)|X| do
        X := X ∪ {v}
    end while
end while
return x'
```

Lemma 11 *Let $\mathcal{G} = (G, r)$ be a BW-game and x be a given potential vector. Then in $O(n|E|)$ time, procedure REDUCE($\mathcal{G}, x$) returns another potential vector x' such that*

(a) $0 \le x'(v) \le \frac{m^+ - m^-}{2}(n-1)$ *for all* $v \in V$*;*

(b) $[m_{x'}] \subseteq [m_x]$.

Proof After the first iteration of the outer while loop, we are guaranteed that the new potential x' satisfies $x'(v) \ge 0$, and it will remain so for the rest of the iterations. In particular, condition (a) will be satisfied when the procedure terminates (provided it does terminate) as follows from the way the set X is updated. Condition (b) is maintained throughout the procedure by our choice of $\epsilon = \min\{\epsilon_1, \epsilon_2, \epsilon_3\}$. So it remains to show that the procedure terminates in at most n steps. Consider any iteration of the outer while loop, where the potential is x' at the beginning of the iteration, and note that any $v \in X$ and $u \in V \setminus X$ have a potential difference of $x'(v) - x'(u) < -\frac{m_x^+ - m_x^-}{2}$. Thus, if $v \in X$, $u \in V \setminus X$ (respectively, $v \in V \setminus X$, $u \in X$), $(v,u) \in E$, and $r_{x'}(v,u) \le m_x^+$ (respectively, $r_{x'}(v,u) \ge m_x^-$) then $r_{x'}(v,u) = r_x(v,u) + x'(v) - x'(u) < \frac{m_x^+ + m_x^-}{2} \le m_x^+$ (respectively,

$r_{x'}(v,u) > \frac{m_x^+ + m_x^-}{2} \geq m_x^-$). This implies that $\epsilon_1 > \frac{m_x^+ + m_x^-}{2}$ and $\epsilon_2 > \frac{m_x^+ + m_x^-}{2}$. Also $\epsilon_3 > 0$ (except possibly for the first iteration). Thus $\epsilon > 0$. When $\epsilon = \epsilon_3$, then with respect to the new potentials x'' (obtained by updating x': $x''(v) = x'(v) - \epsilon$ for all $v \in V \setminus X$), there is a vertex v having $x''(v) = 0$ which will be added to X. On the other hand, when $\epsilon = \epsilon_1$ (respectively, $\epsilon = \epsilon_2$), then there is an edge $(v,u) \in E$ such that $v \in X$, $u \in V \setminus X$ and $\epsilon_1 = m_x^+ - r_{x'}(v,u)$ (respectively, $v \in V \setminus X$, $u \in X$ and $\epsilon_2 = r_{x'}(v,u) - m_x^-$), i.e., $m_x^+ = r_{x''}(v,u) = r_x(v,u) + x''(v) - x''(u) \leq m_x^+ + x''(v) - x''(u)$ (respectively, $m_x^- = r_{x''}(v,u) = r_x(v,u) + x''(v) - x''(u) \geq m_x^- + x''(v) - x''(u)$), from which follows $x''(u) \leq x''(v) \leq \frac{m_x^+ - m_x^-}{2}(|X| - 1)$ (respectively, $x''(v) \leq x''(u) \leq \frac{m_x^+ - m_x^-}{2}(|X| - 1)$). (Note that we used the fact that $r_x(v,u) \leq r_{x'}(v,u)$ for all $(v,u) \in E$ such that $v \in X$ and $u \in V \setminus X$, and $r_x(v,u) \geq r_{x'}(v,u)$ for all $(v,u) \in E$ such that $v \in V \setminus X$ and $u \in X$.) Thus u (respectively, v) will be added to X. □

Appendix D: Examples

Example 1: Solvability in pure positional strategies

Let us consider a graph in which all three classes, White, Black, and Random positions, are not empty. A minimal such graph $G = (V, E)$ contains three vertices $V = \{0, 1, 2\}$, where $V_R = \{0\}$, $V_W = \{1\}$, and $V_B = \{2\}$. Furthermore, let E consists of six arcs $[i, j)$ for all $i, j \in \{0, 1, 2\}$ such that $i \neq j$. Thus, G is a complete tripartite graph without loops and multiple edges. For all local costs and probabilities, the obtained game has a value (in pure positional strategies).

Yet, even for this simple example the reduction to canonical form would demand a long case analysis. Instead, let us consider the normal form. Each player, White and Black, has only two strategies: $s_W^1 = (1,2)$, $s_W^2 = (1,0)$ for White and $s_B^1 = (2,1)$, $s_B^2 = (2,0)$ for Black. We will consider the corresponding 2×2 matrix, with entries $a_{1,1}$, $a_{1,2}$, $a_{2,1}$, $a_{2,2}$, and show that it has a saddle point in pure strategies for all possible parameters of the game.

As usual, we denote by $r(i,j)$ the cost of the move (i,j) and by $p_1 = p_{0,1}$ and $p_2 = p_{0,2}$ the probabilities to move from 0 to 1 and to 2, respectively; assuming that $p_1 > 0, p_2 > 0$ and $p_1 + p_2 = 1$.

Let us consider all four situations. It is easy to see that (s_W^1, s_B^1) results in a simple cycle on vertices 1 and 2, and hence $a_{1,1} = (1/2)(r(1,2) + r(2,1))$.

Situation (s_W^1, s_B^2) results in a Markov chain with the limit distribution:

$$P_1 = p_1/(2+p_1), \quad P_0 = P_2 = 1/(2+p_1) \quad \text{and hence,}$$
$$a_{1,2} = (p_1/(2+p_1))(r(0,1) + r(1,2)) + (p_2/(2+p_1))r(0,2) + (1/(2+p_1))r(2,0).$$

By symmetry, (s_W^2, s_B^1) results in a Markov chain with the limit distribution:

$$P_2 = p_2/(2+p_2), \quad P_0 = P_1 = 1/(2+p_2) \quad \text{and hence,}$$
$$a_{2,1} = (p_2/(2+p_2))(r(0,2) + r(2,1)) + (p_1/(2+p_2))r(0,1) + (1/(2+p_2))r(1,0).$$

Finally, (s_W^2, s_B^2) results in

$$a_{2,2} = (1/2)[p_1(r(0,1) + r(1,0))) + p_2(r(0,2) + r(2,0))].$$

Let us remark that all four limit distributions do not depend on the initial position, in agreement with the ergodicity of G.

Multiplying all entries by $2(2+p_1)(2+p_2)$ we obtain:

$$a'_{1,1} = (2+p_1)(2+p_2)(r(1,2) + r(2,1));$$
$$a'_{1,2} = 2(2+p_2)[p_1(r(0,1) + r(1,2)) + p_2 r(0,2) + r(2,0)];$$
$$a'_{2,1} = 2(2+p_1)[p_2(r(0,2) + r(2,1)) + p_1 r(0,1) + r(1,0)];$$
$$a'_{2,2} = (2+p_1)(2+p_2)[p_1(r(0,1) + r(1,0)) + p_2(r(0,2) + r(2,0))].$$

We claim that this matrix game has a saddle point in pure strategies for all values $r(i,j)$ and p_1, p_2 such that $p_1 > 0, p_2 > 0$, and $p_1 + p_2 = 1$. It is well-known that a 2×2 matrix game has no saddle point in pure strategies if and only if each entry of one diagonal is strictly larger than each entry of the other diagonal. Let us assume indirectly that $\max\{a_{1,1}, a_{2,2}\} < \min\{a_{1,2}, a_{2,1}\}$, that is,

$$a_{1,1} < a_{1,2} \;,\; a_{1,1} < a_{2,1} \;,\; a_{2,2} < a_{1,2} \;, \text{ and } \; a_{2,2} < a_{2,1}.$$

Substituting four entries $a_{i,j}$ we can rewrite this system as follows:

$$2p_1r(0,1) + 2p_2r(0,2) + 2r(2,0) - (2-p_1)r(1,2) - (2+p_1)r(2,1) > 0,$$
$$2p_2r(0,2) + 2p_1r(0,1) + 2r(1,0) - (2-p_2)r(2,1) - (2+p_2)r(1,2) > 0,$$
$$-p_1^2r(0,1) - p_1p_2r(0,2) + (2-(2+p_1)p_2)r(2,0) + 2p_1r(1,2) - (2+p_1)p_1r(1,0) > 0$$
$$-p_2^2r(0,2) - p_2p_1r(0,1) + (2-(2+p_2)p_1)r(1,0) + 2p_2r(2,1) - (2+p_2)p_2r(2,0) > 0$$

If we assume that $\min\{a_{1,1}, a_{2,2}\} > \max\{a_{1,2}, a_{2,1}\}$ then we obtain the same four inequalities multiplied by -1. In both cases we get a contradiction, since their linear combination with the strictly positive coefficients: $p_1p_2^2$, $p_2p_1^2$, p_2, and p_1 results in 0 in the left hand side.

Example 2: Local and global optimality

Let us consider a graph with 2 nodes: a white node u and a black node v. There are two loops (u,u) and (v,v) with rewards -1 and 1 respectively, and two more arcs that form a cycle $(u,v),(v,u)$ with rewards, $-R$ and R, respectively, where R is a large positive integer. This very small example with no random nodes illustrates many important points. First, the locally attractive moves (u,u) and (v,v) are not optimal, since the optimal strategy for both players is to follow the cycle of average reward 0. Second, the range of m (which is $[-1,1]$ in this case) can be much smaller than the range of R, and yet, the arcs with very large local rewards are not eligible for elimination. Third, our pumping algorithm (as well as the simplex-like algorithm of [GKK88]) will finish in $O(\log R)$ iterations on this example, while the dynamic programming approach of [ZP96] takes $O(R)$ iterations.

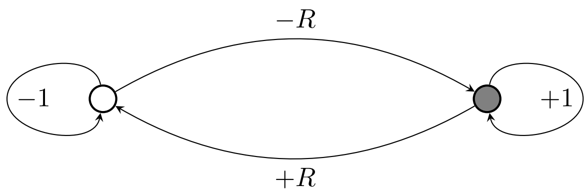

Figure 2: Example 2

Appendix E: Experimental results

We performed a number of experiments testing our algorithm on random tripartite graphs, with random rewards and probabilities. We generated probabilities that are multiples of $\frac{1}{n}$, where $n = |V|$, but let the maximum absolute local reward be as large as 10^8. Our results are summarized in the following table.

	$R = 10^3$		$R = 10^6$		$R = 10^8$	
n	$\bar{N}$	$\hat{N}$	$\bar{N}$	$\hat{N}$	$\bar{N}$	$\hat{N}$
600	91	338	142	810	144	496
1200	91	385	152	566	172	545
1800	110	643	122	428	162	806
2400	104	318	159	508	178	767
3000	118	338	155	551	162	544
3600	113	397	138	541	180	576
4200	114	283	153	569	213	693
4800	126	681	164	680	194	702
5400	124	328	169	586	246	5284
6000	118	290	170	620	210	845
6600	138	535	173	565	202	718
7200	125	442	161	664	181	750
7800	143	860	170	588	188	604
8400	115	296	185	541	200	717
9000	121	368	171	543	202	1071
9600	128	361	171	532	201	624
10200	122	523	173	639	166	566
10800	142	532	172	815	211	759
11400	131	778	155	666	219	928
12000	141	587	198	2107	194	831
12600	159	918	180	664	187	747
13200	137	642	173	752	210	638
13800	116	355	172	701	198	768
14400	125	804	156	715	228	809
15000	139	511	165	632	227	1593

Table 1: Experimental results showing the average number of iterations $\bar{N}$ on random graphs with $n \in \{300, \ldots, 15000\}$ nodes, $|V_W| = |V_B| = |V_R| = \frac{n}{3}$, and maximum reward $R \in \{10^3, 10^6, 10^8\}$. $\hat{N}$ is the maximum number of iterations encountered in all experiments with the given selection of parameters.