Stochastic Mean Payoff Games: Smoothed Analysis and Approximation Schemes*

Endre Boros[1], Khaled Elbassioni[2], Mahmoud Fouz[3], Vladimir Gurvich[1], Kazuhisa Makino[4], and Bodo Manthey[5]

[1] RUTCOR, Rutgers University
[2] Max-Planck-Institut für Informatik, Saarbrücken, Germany
[3] Universität des Saarlandes, Fachrichtung Informatik, Germany
[4] University of Tokyo, Graduate School of Information Science and Technology
[5] University of Twente, Department of Applied Mathematics

Abstract. In this paper, we consider two-player zero-sum stochastic mean payoff games with perfect information modeled by a digraph with black, white, and random vertices. These *BWR-games* games are polynomially equivalent with the classical Gillette games, which include many well-known subclasses, such as cyclic games, simple stochastic games, stochastic parity games, and Markov decision processes. They can also be used to model parlor games such as Chess or Backgammon.
It is a long-standing open question if a polynomial algorithm exists that solves BWR-games. In fact, a pseudo-polynomial algorithm for these games with an arbitrary number of random nodes would already imply their polynomial solvability. Currently, only two classes are known to have such a pseudo-polynomial algorithm: BW-games (the case with *no* random nodes) and *ergodic* BWR-games (in which the game's value does not depend on the initial position) with constant number of random nodes. In this paper, we show that the existence of a pseudo-polynomial algorithm for BWR-games with constant number of random vertices implies smoothed polynomial complexity and the existence of absolute and relative polynomial-time approximation schemes. In particular, we obtain smoothed polynomial complexity and derive absolute and relative approximation schemes for BW-games and ergodic BWR-games (assuming a technical requirement about the probabilities at the random nodes).

1 Introduction

The rise of the Internet has led to an explosion in research in game theory: the mathematical modeling of competing agents in strategic situations. The central concept in such models is that of a *Nash equilibrium*, defining a state where

* The first author is grateful for the partial support of the National Science Foundation (CMMI-0856663, "Discrete Moment Problems and Applications"), and the first, second, fourth and fifth authors are thankful to the Mathematisches Forschungsinstitut Oberwolfach for providing a stimulating research environment with an RIP award in March 2010.

no agent gains an advantage by changing her current strategy; it serves as a prediction for the outcome of strategic situations in which selfish agents compete.

A fundamental result in game theory shows that if the agents can choose a *mixed strategy* (i.e., probability distributions of deterministic strategies), a Nash equilibrium is guaranteed to exist in finite games. Often, however, already *pure* (i.e., deterministic) strategies already lead to a Nash equilibrium. Still, the existence of Nash equilibria might be irrelevant in practice, since their computation would take too long. Thus, algorithmic aspects of game theory have gained a lot of interest. Following the dogma that only polynomial time algorithms are feasible in practice, it is desirable to show polynomial time complexity for the computation of Nash equilibria. On the other hand, in cases where such an efficient algorithm is not known to exist an approximate notion of Nash equilibria has been suggested, in which no agent can gain a substantial advantage by changing her current strategy. In this paper, we advocate another notion of tractability by considering the *smoothed complexity* of a well-known two-player stochastic game for which the existence of a polynomial algorithm is a long-standing open question. In contrast to the usual worst-case complexity, smoothed complexity analyzes the running time of algorithms on typical instances. By establishing smoothed polynomial complexity, we argue that the computation of a Nash equilibrium is feasible in all, but artificially constructed worst-case instances.

The model that we consider is *mean stochastic payoff games* or *BWR-games*: we are given a directed graph $G = (V, E)$ 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*, respectively. The arcs stand for *moves*. The black and white vertices are owned by two players: BLACK – the *minimizer* – owns the black vertices in V_B, and WHITE – the *maximizer* – owns the white vertices in V_W. The vertices in V_R are owned by nature. We have a local reward $r_e \in \mathbb{R}$ for each arc $e \in E$ and a probability p_{vu} for each arc (v, u) going out of $v \in V_R$. Starting from some vertex $v_0 \in V$, a token is moved along one arc e in every round of the game. If the token is on a black vertex, BLACK selects an outgoing arc e and moves the token along e. If the token is on a white vertex, WHITE selects an outgoing arc e. In a random position $v \in V_R$, a move $e = (v, u)$ is chosen according to the probabilities p_{vu} of the outgoing arcs of v. In all cases, BLACK pays WHITE the reward r_e on the selected arc e. A strategy of a player is a mapping that assign a move $(u, v) \in E$ to each position u he owns.

Starting from a given initial position $v_0 \in V$, the game produces for a pair of fixed strategies (i.e., one for each player) an infinite walk $\{v_0, v_1, v_2, \ldots\}$ (called a *play*). Let b_i denote the reward $r_{v_i v_{i+1}}$ received by WHITE in step $i \in \{0, 1, \ldots\}$. The *undiscounted limit average effective payoff* is defined as the *Cesàro average* $c = \liminf_{n\to\infty} \frac{\sum_{i=0}^{n} \mathbb{E}[b_i]}{n+1}$. WHITE's objective is to maximize c, while BLACK's objective is to minimize c. Every such game is known to have a pair of *uniformly optimal strategies* that result in a Nash equilibrium (called a *saddle point*) from *any* initial position [4, 8]. All optimal pairs of strategies, from a given initial position v result in a unique payoff $\mu(v)$, called the value of the game at v. An algorithm is said to solve the game if it computes an optimal pair of strategies.

BWR-games are an equivalent formulation [5] of the stochastic games with perfect information and mean payoff that were introduced in 1957 by Gillette [4]. They generalize many important problems. The special case of BWR-games without random vertices ($V_R = \emptyset$) is known as *cyclic* or *mean payoff* games (see, e.g., [5]); we call these *BW-games*. If one of the sets V_B or V_W is empty, we obtain a *Markov decision process* for which polynomial-time algorithms are known [9]. If both are empty ($V_B = V_W = \emptyset$), we get a *weighted Markov chain*.

Besides their many applications, all these games are of interest to complexity theory: Karzanov and Lebedev [7] 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 even in this special case, see, e.g., the recent survey by Vorobyov [14]. A similar complexity claim can be shown to hold for BWR-games. On the other hand, there exist algorithms (see, e.g., [5]) that solve BW-games in practice very fast. The situation for these games is thus comparable to linear programming before the seminal discovery of the ellipsoid method, where the problem was also known to lie in the intersection of NP and co-NP and where the simplex algorithm proved to be a fast algorithm in practice. Spielman and Teng [12, 13] introduced smoothed analysis to explain the practical performance of the simplex method. We further enforce this analogy by showing a smoothed polynomial complexity for a large class of BWR-games.

While there are numerous pseudo-polynomial algorithms known for the BW-case [5, 15], pseudo-polynomiality for BWR-games (with no restriction on the number of random nodes) is in fact equivalent to polynomiality [1]. Recently, a pseudo-polynomial algorithm was given in [3] for BWR-games with a *constant* number of random vertices and polynomial common denominator of transition probabilities, but under the assumption that the game is *ergodic*, i.e., the game value does not depend on the initial position. However, the existence of a similar algorithm for the non-ergodic or non-constant number of random vertices remains open, as the approach in [3] does not seem to generalize to these cases.

1.1 Our Results and Some Related Work

Approximation Schemes. The only result that we are aware of regarding approximation schemes is the observation made by Roth et al. [11] that the values of BW-games can be approximated within an *absolute* error of ε in polynomial-time, if all rewards are in the range $[-1, 1]$. This follows immediately from truncating the rewards and using any of the known pseudo-polynomial algorithms.

In this paper, we generalize this result to BWR-games in two directions. Throughout the paper, we write $\mathbb{G}$ for any class of digraphs $G = (V_B \cup V_W \cup V_R, E)$ that admit a pseudo-polynomial algorithm $\mathbb{A}$, i.e., $\mathbb{A}$ solves any BWR-game $\mathcal{G}$ on $G \in \mathbb{G}$, with integral rewards and rational transition probabilities, in time polynomial in n, D, and R, where $n = n(\mathcal{G})$ is the total number of vertices, $R = R(\mathcal{G})$ is the size of the range of the rewards, and $D = D(\mathcal{G})$ is the common denominator of the transition probabilities. E.g., digraphs without random vertices are known to belong to $\mathbb{G}$. The same holds for digraphs that have a constant number of random nodes and are *structurally ergodic* [6], i.e., for *any* set of rewards, all positions have the same game value. Note that the dependence on D is inherent in all known pseudo-polynomial algorithms for BWR-games.

Let $p_{\min} = p_{\min}(\mathcal{G})$ be the minimum positive transition probability in the game $\mathcal{G}$. Throughout the paper, we will assume that $k := |V_R|$ is constant.

Theorem 1. *For any $\varepsilon > 0$, there exists for each of the following two cases an algorithm that returns a pair of strategies that approximates the value of any BWR-game on $G \in \mathbb{G}$ from any starting position:*

i. With rewards in the interval $[-1, 1]$*, within an* absolute *error of ε, in time* $\text{poly}(n, \frac{1}{p_{\min}}, \frac{1}{\varepsilon})$.
ii. With non-negative integral rewards, within a relative *error of ε, in time* $\text{poly}(n, \log R, \frac{1}{p_{\min}}, \frac{1}{\varepsilon})$.

Our reduction in case (i), unlike case (ii), has the property that if the pseudo-polynomial algorithm returns *uniformly* optimal strategies, i.e., that are *independent* of the starting position, then so does the approximation scheme (in an approximate sense). With some more work, we can show that the same is also true in case (ii) of Theorem 1 for BW-games.

In deriving these approximation schemes from a pseudo-polynomial algorithm as defined above, we face two main technical challenges that distinguish the computation of approximate equilibria of BWR-games from similar standard techniques used in optimization: (i) the running time of the pseudo-polynomial algorithm depends polynomially both on the maximum reward and the common denominator D of the transition probabilities; thus to obtain a *fully polynomial-time approximation scheme* (FPTAS) with an absolute guarantee whose running time is independent of D, we need to truncate the probabilities and bound the change in the game value, which is a *non-linear* function of D, (ii) to obtain an FPTAS with a relative guarantee, one usually exploits a (trivial) lower/upper bound on the optimum value; this is not possible in the case of BWR-games, since the game value can be arbitrarily small; the situation becomes even more complicated, if we look for uniformly ε-optimal strategies, since we have to output one pair of strategies which guarantees ε-optimality from any starting position. In order to solve the first issue, we analyze the change in the game values and optimal strategies if the rewards or transition probabilities are changed. The second issue is solved through repeated applications of the pseudo-polynomial algorithm on a truncated game; after each such application we show that either the value of the game has already been approximated within the required accuracy, or the range of the rewards can be shrunk by a constant factor without changing the value of the game (Sections 3.2 and 3.3).Since BW-games and structurally ergodic BWR-games with constant k admit pseudo-polynomial algorithms, we obtain the following results.

Corollary 1. *There is an FPTAS that solves,*

i. within a relative error, in uniformly ε-optimal strategies, any BW-game with non-negative (rational) rewards;

ii. *within an absolute error, in uniformly ε-optimal strategies, any structurally ergodic BWR-game with rewards in* $[-1, 1]$ *and* $\frac{1}{p_{\min}} = \text{poly}(n)$*;*
iii. *within a relative error, in uniformly ε-optimal strategies, any structurally ergodic BWR-game with non-negative rational rewards and* $\frac{1}{p_{\min}} = \text{poly}(n)$.

Note that (i) strengthens the absolute FPTAS for BW-games [11], and (ii) and (iii) enlarge the class of games for which an FPTAS exists.

Smoothed Analysis for BWR-games. We further show that typical instances of digraphs that admit a pseudo-polynomial algorithm can be solved in polynomial time. Towards this end, we do a smoothed analysis using the one-step model introduced by Beier and Vöcking [2]: an adversary specifies a BWR-game $\mathcal{G}$ and for each arc a density function. These functions are bounded from above by a parameter ϕ. Then the rewards for all arcs are drawn independently according to their respective density functions. We prove that in this setting, independent of the actual choices of the adversary, the resulting game can be solved in polynomial time with high probability; there exists a polynomial $P(n, \phi, 1/\varepsilon)$ such that the probability that the algorithm exceeds a running-time of $P(n, \phi, 1/\varepsilon)$ is at most ε. This shows that such BWR-games with a constant number of random vertices have smoothed polynomial complexity.

Theorem 2. *There is an algorithm solving any BWR-game on any* $G \in \mathbb{G}$ *with rational transition probabilities and* $D = \text{poly}(n)$ *in smoothed polynomial time.*

Theorem 2 is similar to the result by Beier and Vöcking [2] who showed that a binary optimization problem defined by linear constraints and a linear objective function has smoothed polynomial complexity if it admits a pseudo-polynomial algorithm. Our proof of Theorem 2 has a similar structure like their analysis. However, in the case of BWR-games, the situation becomes more complicated: First, we have to deal with two conflicting objectives (of the two players). Second, the coefficients of the objective functions are not given explicitly. In consequence, our proof requires a novel isolation lemma that deals with two players who optimize the same objective function in two different directions. Furthermore, our procedure for certifying that the solution found is indeed the optimal solution is considerably more involved and requires careful rounding of the coefficients in order to certify optimality.

Corollary 2. *(i) BW-games and (ii) structurally ergodic BWR-games with* $D = \text{poly}(n)$ *can be solved in smoothed polynomial time.*

Let us remark finally that removing the assumption that k is constant in the above results remains a challenging open problem.

2 Preliminaries, Notation and Basic Properties

BWR-games and Markov Chains. A BWR-game is defined by a triple $\mathcal{G} = (G, P, r)$, 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, i.e., vertices of out-degree 0; $P \in [0,1]^E$ is the vector of probability distributions for all $v \in V_R$ specifying the probability p_{vu} of a move from v to u; $r \in \mathbb{R}^E$ is a local reward function. It is assumed that $\sum_{u:(v,u)\in E} p_{vu} = 1$ for all $v \in V_R$ and $p_{v,u} > 0$ whenever $(v,u) \in E$ and $v \in V_R$.

Standardly, we define a strategy $s_W \in S_W$ for WHITE as a mapping that assigns a move $(v,u) \in E$ to each position $v \in V_W$. For simplicity, we may write $s_W(v) = u$ for $s_W(v) = (v,u)$. Strategies $s_B \in S_B$ for BLACK are analogously defined. A pair of strategies $s = (s_W, s_B)$ is called a *situation*. Given a BWR-game $\mathcal{G} = (G, P, r)$ and a situation $s = (s_B, s_W)$, we obtain a weighted Markov chain $\mathcal{G}(s) = (G(s) = (V, E(s)), P(s), r)$ with transition matrix $P(s)$ defined by:

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

Here, $E(s) = \{e \in E \mid p_e(s) > 0\}$ is the set of arcs with positive probability. Given an initial position $v_0 \in V$ from which the play starts, we define the limiting (mean) effective payoff $c_{v_0}(s)$ in $\mathcal{G}(s)$ as $c_{v_0}(s) = \rho(s)^T r = \sum_{e\in E} \rho_e(s) r_e$, where $\rho(s) = \rho(s, v_0) \in [0,1]^E$ is the arc-limiting distribution for $\mathcal{G}(s)$ starting from v_0. This means that for $(v,u) \in E$, $\rho_{vu}(s) = \pi_v(s) p_{vu}(s)$, where $\pi \in [0,1]^V$ is the limiting distribution in the Markov chain $\mathcal{G}(s)$ starting from v_0. In what follows, we use $(\mathcal{G}, v_0)$ to denote the game starting from v_0. We write $\rho(s)$ for $\rho(s, v_0)$, when v_0 is clear from the context. For rewards $r : E \to \mathbb{R}$, let $r^- = \min_e r_e$ and $r^+ = \max_e r_e$. Let $[r] = [r^-, r^+]$ be the range of r. Let $R = R(\mathcal{G}) = r^+ - r^-$.

Strategies and Saddle Points. If we consider $c_{v_0}(s)$ for all possible situations, we obtain a matrix game $C_{v_0} : S_W \times S_B \to \mathbb{R}$, with entries $C_{v_0}(s_W, s_B) = c_{v_0}(s_W, s_B)$. Every such game has a Nash equilibrium in pure strategies [4, 8]; a corresponding pair of strategies is said to be *optimal*. Moreover, there exists optimal strategies (s_W^*, s_B^*) that do not depend on the starting position v_0; such strategies are called *uniformly optimal*. Although there might be several optimal strategies, it is easy to see that they all lead to the same value. We define this to be the value of the game and write $\mu_{v_0}(\mathcal{G}) := C_{v_0}(s_W^*, s_B^*)$ where (s_W^*, s_B^*) is any pair of optimal strategies. Note that $\mu_{v_0}(\mathcal{G})$ may depend on the starting node v_0.

3 Approximation Schemes

Given a BWR-game $\mathcal{G} = (G = (V,E), P, r)$, a constant $\varepsilon > 0$, and a starting position $v \in V$, an ε*-relative approximation* of the value of the game is determined by a situation (s_W^*, s_B^*) such that

$$\max_{s_W} \mu_v(\mathcal{G}(s_W, s_B^*)) \leq (1+\varepsilon)\mu_v(\mathcal{G}) \quad \text{and} \quad \min_{s_B} \mu_v(\mathcal{G}(s_W^*, s_B)) \geq (1-\varepsilon)\mu_v(\mathcal{G}). \quad (1)$$

An alternative to relative approximations is to look for an approximation with *absolute* error of ε. This is achieved by a situation (s_W^*, s_B^*) such that

$$\max_{s_W} \mu_v(\mathcal{G}(s_W, s_B^*)) \leq \mu_v(\mathcal{G}) + \varepsilon \quad \text{and} \quad \min_{s_B} \mu_v(\mathcal{G}(s_W^*, s_B)) \geq \mu_v(\mathcal{G}) - \varepsilon. \quad (2)$$

A situation (s_W^*, s_B^*) satisfying (1) (resp., (2)) is called relative (resp., absolute) ε-optimal. If the pair (s_W^*, s_B^*) is ε-optimal for any starting position, it is called *uniformly* ε-optimal.

3.1 Absolute Approximation

Let $G = (V, E)$ be a graph in $\mathbb{G}$ and $\mathcal{G} = (G, P, r)$ be a BWR-game on G. In this section, we assume that $r^- = -1$ and $r^+ = 1$, i.e., all rewards are from the interval $[-1, 1]$. We apply the pseudo-polynomial algorithm $\mathbb{A}$ on a truncated game $\tilde{\mathcal{G}} = (G = (V, E), \tilde{P}, \tilde{r})$ defined by rounding the rewards to the nearest integer multiple of $\varepsilon/4$ (denoted $\tilde{r} := \lfloor r \rceil_{\frac{\varepsilon}{4}}$), and truncating the vector of probabilities $(p_{vu} : u \in V)$ for each random node $v \in V_R$ as follows.

Lemma 1. *Let* $\alpha \in [0, 1]^n$ *with* $\|\alpha\|_1 = 1$. *Let* $B \in \mathbb{Z}^+$ *be an integer such that* $\min_{i:\alpha_i>0}\{\alpha_i\} > 2^{-B}$. *Then there exists* $\alpha' \in [0, 1]^n$ *such that (i)* $\|\alpha'\|_1 = 1$*; (ii) for all* $i = 1, \ldots, n$, $\alpha'_i = c_i/2^B$ *where* $c_i \in \mathbb{Z}^+$ *is an integer; (iii) for all* $i = 1, \ldots, n$, $\alpha'_i > 0$ *if and only* $\alpha_i > 0$, *and (iv)* $\|\alpha - \alpha'\|_\infty \le 2^{-B}$.

Lemma 2. *Let* $\mathbb{A}$ *be a pseudo-polynomial algorithm that solves, in (uniformly) optimal strategies, any BWR-game* $\mathcal{G} = (G, P, r)$ *with* $G \in \mathbb{G}$ *in time* $\tau(n, D, R)$. *Then for any* $\varepsilon > 0$, *there is an algorithm that solves, in (uniformly) absolute* ε*-optimal strategies, any BWR-game* $\mathcal{G} = (G, P, r)$ *with* $G \in \mathbb{G}$ *in time bounded by* $\tau(n, \frac{2^{2k+5} n^3 k^2}{\varepsilon p_{\min}^{2k}}, \frac{4}{\varepsilon})$, *where* $p_{\min} = p_{\min}(\mathcal{G})$.

3.2 Relative approximation

Let $G = (V, E)$ be a graph in $\mathbb{G}$ and $\mathcal{G} = (G, P, r)$ be a BWR-game on G with non-negative rational rewards (i.e., $r^- = 0$). Without loss of generality, we may assume that the rewards are integral with $\min_{e:r_e>0} r_e = 1$. The algorithm is given as Algorithm 1. The main idea is to truncate the rewards, scaled by a certain factor $1/K$, and use the pseudo-polynomial algorithm on the truncated game $\hat{\mathcal{G}}$. If the value in the truncated game $\mu_w(\hat{\mathcal{G}})$, from the starting node w, is large enough (step 5) then we get a good relative approximation of the original value and we are done. Otherwise, the information that $\mu_w(\hat{\mathcal{G}})$ is small allows us to reduce the maximum reward by a factor of 2 in the original game (step 8). Thus the algorithm terminates in polynomial time (in the bit length of $R(\mathcal{G})$). To remove the dependence on D in the running time, we need also to truncate the transition probabilities. In the algorithm, we denote by $\tilde{P}$ the transition probabilities obtained from P by applying Lemma 1 with $B = \lceil \log 1/\varepsilon' \rceil$, where we select $\varepsilon' = \frac{p_{\min}^{2k}}{2^{2k+3} n^3 k^2 \theta}$, where $\theta = \theta(\mathcal{G}) := \frac{2(1+\varepsilon)(3+2\varepsilon)n}{\varepsilon p_{\min}^{2k+1}}$, so that $2\delta(\mathcal{G}, \varepsilon') \le \frac{r_+(\mathcal{G})}{\theta(\mathcal{G})} := K(\mathcal{G})$, where $\delta(\mathcal{G}, \varepsilon) := \Big(\frac{\varepsilon}{2} n^2 \big(\frac{1}{2} p_{\min}\big)^{-k} \big[\varepsilon n k(k+1)\big(\frac{1}{2} p_{\min}\big)^{-k} + 3k + 1\big] + \varepsilon n\Big) r_*$ with $r_* = r_*(\mathcal{G}) := \max\{|r_+(\mathcal{G})|, |r_-(\mathcal{G})|\}$.

Lemma 3. *Let* $\mathbb{A}$ *be a pseudo-polynomial algorithm that solves any BWR-game* $\mathcal{G} = (G, P, r)$ *with* $G \in \mathbb{G}$ *in time* $\tau(n, D, R)$. *Then for any* $\varepsilon \in (0, 1)$, *there*

Algorithm 1 FPTAS-BWR($\mathcal{G}, w, \varepsilon$)

Input: a BWR-game $\mathcal{G} = (G = (V, E), P, r)$, a starting vertex $w \in V$, an accuracy ε.
Output: an ε-optimal pair $(\tilde{s}_W, \tilde{s}_B)$ for the game $(\mathcal{G}, w)$.

if $r^+(\mathcal{G}) = 1$ **then**
  $\hat{\mathcal{G}} := (G, \tilde{P}, r)$; **return** $\mathbb{A}(\hat{\mathcal{G}}, v)$
$K := \frac{r^+(\mathcal{G})}{\theta(\mathcal{G})}$; $\hat{r}_e = \lfloor \frac{r_e}{K} \rfloor$ for $e \in E$; $\hat{\mathcal{G}} = (G, \tilde{P}, \hat{r})$
$(\tilde{s}_W, \tilde{s}_B) := \mathbb{A}(\hat{\mathcal{G}}, w)$
if $\mu_w(\hat{\mathcal{G}}) \geq \frac{3}{\varepsilon}$ **then**
  return $(\tilde{s}_W, \tilde{s}_B)$
else
  for all $e \in E$, let $\tilde{r}_e = \begin{cases} \lceil \frac{r^+}{2} \rceil & \text{if } r_e > \frac{r^+}{2(1+\varepsilon)} \\ r_e & \text{otherwise} \end{cases}$
  $\tilde{\mathcal{G}} := (G, P, \tilde{r})$; **return** FPTAS-BWR($\tilde{\mathcal{G}}, w, \varepsilon$)

Algorithm 2 FPTAS-BW($\mathcal{G}, \varepsilon$)

Input: a BW-game $\mathcal{G} = (G = (V = V_B \cup V_W, E), r)$, and accuracy ε.
Output: a uniformly ε-optimal pair $(\tilde{s}_W, \tilde{s}_B)$ for $\mathcal{G}$.

if $r^+(\mathcal{G}) = 1$ **then**
  return $\mathbb{A}(\mathcal{G})$
$K := \frac{\varepsilon' r^+}{2(1+\varepsilon')^2 n}$; $\hat{r}_e = \lfloor \frac{r_e}{K} \rfloor$ for $e \in E$; $\hat{\mathcal{G}} = (G, \hat{r})$
$(\hat{s}_W, \hat{s}_B) := \mathbb{A}(\hat{\mathcal{G}})$; $U := \{u \in V \mid \mu_u(\hat{\mathcal{G}}) \geq \frac{1}{\varepsilon'}\}$
if $U = V$ **then**
  return $(\tilde{s}_W, \tilde{s}_B) = (\hat{s}_W, \hat{s}_B)$
else
  $\tilde{G} := G[V \setminus U]$
  for all $e \in E(\tilde{G})$, let $\tilde{r}_e = \begin{cases} \lceil \frac{r^+}{2} \rceil & \text{if } r_e > \frac{r^+}{2(1+\varepsilon')} \\ r_e & \text{otherwise} \end{cases}$
  $\tilde{\mathcal{G}} := (\tilde{G}, \tilde{r})$
  $(\tilde{s}_W, \tilde{s}_B) :=$FPTAS-BW($\tilde{\mathcal{G}}, \varepsilon$)
  $\tilde{s}(w) := \hat{s}(w)$ for all $w \in U$; $\tilde{s} = (\tilde{s}_W, \tilde{s}_B)$

is an algorithm that solves, in relative ε-optimal strategies, any BWR-game $(\mathcal{G} = (G, P, r), w)$ with $G \in \mathbb{G}$, from any given starting position w, in time $\left(\tau\left(n, \frac{4^{k+2} n^4 k^2 (1+\varepsilon)(3+2\varepsilon)}{\varepsilon p_{\min}^{2k}}, \frac{2(1+\varepsilon)(3+2\varepsilon)n}{\varepsilon p_{\min}^{2k+1}}\right) + \mathrm{poly}(n)\right)(\lfloor \log R \rfloor + 1)$.

Remark 1. It is easy to see that, for structurally ergodic BWR-games, one can modify the above procedure to return uniformly ε-optimal strategies.

3.3 Uniformly relative ε-approximation for BW-games

Note that the FPTAS in Lemma 3 does not necessarily return a uniformly ε-optimal situation, even if the pseudo-polynomial algorithm $\mathbb{A}$ provides a uniformly optimal situation. In case of BW-games, we can modify this FPTAS to return a situation which is ε-optimal for all $v \in V$. The algorithm is given as Algorithm 2. The main difference is that when we recurse on a game with reduced rewards (step 11), we have also to delete all nodes that have large values $\mu(\tilde{\mathcal{G}}, v)$ in the truncated game. This is similar to the approach used to decompose a BW-game into ergodic classes [5]. However, the main technical difficulty is that, with approximate equilibria, White (resp., Black) might still have some incentive to move from a higher-value (resp., lower-value) class to a lower-value (resp., higher-value) class, since the values are just estimated approximately. We show that such a move will not be very profitable for White (resp., Black). As before, we assume that the rewards are integral with $\min_{e:r_e>0} r_e = 1$.

Lemma 4. *Let $\mathbb{A}$ be a pseudo-polynomial algorithm that solves, in uniformly optimal strategies, any BWR-game $\mathcal{G} = (G, P, r)$ with $G \in \mathbb{G}$ in time $\tau(n, R)$. Then for any $\varepsilon > 0$, there is an algorithm that solves, in uniformly relative ε-optimal strategies, any BW-game $\mathcal{G} = (G, P, r)$ with $G \in \mathbb{G}$, in time $(\tau(n, \frac{2(1+\varepsilon')^2 n}{\varepsilon'}) + \text{poly}(n))h$, where $h = \lfloor \log R \rfloor + 1$, and $\varepsilon' = \frac{\ln(1+\varepsilon)}{4h-2}$.*

4 Smoothed Analysis

We use the following notion of polynomial smoothed complexity introduced by Beier and Vöcking [2]. A problem is said to have smoothed polynomial complexity if and only if there exists an algorithm $\mathcal{A}$ with running-time T and a constant α such that

$$\forall \phi \geq 1, \forall n \in \mathbb{N} : \max_{\boldsymbol{f} \in D_n(\phi)} \mathbb{E}_{X \sim \boldsymbol{f}}(T(X)^\alpha) = O(n\phi) . \quad (3)$$

Here, $D_n(\phi)$ denotes all possible vectors of density functions bounded by ϕ for instances of size n, and X is an instance drawn according to $\boldsymbol{f}$. Equivalently, there exists a polynomial $P(n, \phi, 1/\varepsilon)$ such that with probability at most ε, $\mathcal{A}$ exceeds a running-time of $P(n, \phi, 1/\varepsilon)$.

Let $\mathbb{A}$ be a pseudo-polynomial algorithm that solves any BWR-game $\mathcal{G} = (G, P, r)$ with $G = (V, E) \in \mathbb{G}$. In this section, we show that any such game can be solved in smoothed polynomial time. For this, we assume that an adversary specifies a game together with density functions for the rewards (one for each arc), and these density functions are bounded by ϕ, and show a bound as in (3). One (technical) issue is that the perturbed rewards are of course real, non-rational numbers with probability 1. Thus, we cannot really use existing algorithms as sub-routine, and we cannot even compute anything with these numbers on an ordinary RAM. To cope with this problem, we use Beier and Vöcking's [2] approach and assume that the rewards are in $[-1, 1]$ and that we can access the bits of the rewards one-by-one.

To state our results in a bit more general setting, we will assume that $\mathbb{A}$ solves any BWR-game $\mathcal{G}$ in uniformly optimal strategies. If this was not the case, then it is easy to modify the procedure and analysis in this section to solve the game starting from a given vertex.

Algorithm 3 Solve($\mathcal{G}$)

Input: a BWR-game $\mathcal{G} = (G = (V, E), P, r)$.
Output: an optimal pair $(\tilde{s}_W, \tilde{s}_B)$ for the game $\mathcal{G}$.

```
1: ℓ_0 ← log((nD)^{c_0} φ); i ← 0 {c_0 is a constant to be specified later}
2: repeat
3:     ℓ := ℓ_0 + i; ε ← 2^{−ℓ}; i := i + 1
4:     r̃ := ⌊r⌋_ℓ; G̃ := (G, P, r̃); G̃' := (G, P, 2^ℓ r̃)
5:     (s̃_W, s̃_B) := 𝔸(G̃')
6: until s̃ is optimal in G̃_{e,ε} for all e ∈ E
```

Before describing the procedure (Algorithm 3), we need to introduce some notation. Let us write $\lfloor x \rfloor_b$ for the largest integer smaller than or equal to x that has b bits (i.e., we basically cut off all bits after the b-th bit). Let $\gamma = \gamma(\mathcal{G}) := (kn)^{-2}(2D)^{-2(k+2)}$ and $\varepsilon > 0$. Given the game $\mathcal{G} = (G = (V, E), P, r)$, define, for each $e \in E$, the game $\mathcal{G}_{e,\varepsilon} = (G, P, r(e))$, where

$$r_{e'}(e) = \begin{cases} r_e + 2\gamma^{-1}\varepsilon & \text{if } e' = e, \\ r_{e'} & \text{otherwise.} \end{cases} \tag{4}$$

The basic idea behind our smoothed analysis is as follows: We use a certain number of bits for each reward. Then we run the pseudo-polynomial algorithm to solve the resulting game with the rewards rounded down (and scaled to integers) because we do not have more bits at that point (Step 4). This can be done in polynomial-time as long as we have $O(\log n)$ bits. Then we try to certify that the solution obtained is also a solution for the true rewards (Step 6). If this succeeds, then we are done. If this fails, then we use one more bit and repeat the process.

To prove a smoothed polynomial running time, we need to show that with high probability a logarithmic number of bits suffices to compute an equilibrium for the original (untruncated) game. Furthermore, we have to devise a certificate proving that the computed equilibrium is indeed an equilibrium for the original game (we will show that such a certificate is given in Step 6). Both results are based on a sensitivity analysis of the game: we show that by changing the rewards slightly, an optimal strategy remains optimal for the changed game.

A key ingredient for our smoothed analysis is an adaption of the isolation lemma [10] to our setting. An adaption of the isolation lemma has already been used successfully in smoothed analysis of integer programs [2]. It basically says the following: Of course, there are exponentially many alternative strategies for each player. But if a player replaces the optimal strategy with an alternative strategy, the payoff for the respective player gets worse significantly.

Lemma 5 (Isolation Lemma). *Let E be a finite set, and $\mathcal{F} \subset \mathbb{R}_+^E$ be a family of (distinct) vectors, such that for any distinct $\rho, \rho' \in \mathcal{F}$, there exists an $e \in E$ with $|\rho_e - \rho'_e| \geq \gamma$. Let $\{w_e\}_{e \in E}$ be independent continuous random variables with maximum density ϕ. Define* $\text{gap}(w) := w^T\rho^* - w^T\rho^{**}$*, where $\rho^* = \text{argmax}_{\rho \in \mathcal{F}} w^T\rho$ and $\rho^{**} = \text{argmax}_{\rho \in \mathcal{F},\ \rho \neq \rho^*} w^T\rho$. Then* $\Pr(\text{gap}(w) \leq \varepsilon) \leq |E|\varepsilon\phi\frac{\kappa^2}{\gamma}$*, where $\kappa = \max_{e \in E} |\mathcal{F}_e|$, and $\mathcal{F}_e = \{x \mid \rho_e = x \text{ for some } \rho \in \mathcal{F}\}$.*

We use the above lemma with the set $\mathcal{F}$ representing a set of arc-limiting distributions, corresponding to a set of situations in the game starting from a certain vertex. For that we need bounds for κ and γ.

Lemma 6. *Let $\mathcal{G} = (G = (V, E), P, r)$ be a BWR-game, $u \in V$ be any vertex, and s be an arbitrary situation. Then (i) every entry of the arc-limiting distribution $\rho(s)$ for the Markov chain $(\mathcal{G}(s), u)$ can be written as rational numbers of the form $\frac{a}{b}$, where $a, b \in \mathbb{Z}_+$ and $a, b \leq kn(2D)^{k+2}$. Hence, (ii) the number of possible entries in $\rho(s)$ is bounded by $\kappa = (kn)^2(2D)^{2(k+2)}$, and (iii) for any situation s' such that $\rho(s') \neq \rho(s)$, there is an arc e such that $\rho_e(s) - \rho_e(s') \geq \gamma = \gamma(\mathcal{G})$.*

To use the given pseudo-polynomial algorithm, we have to truncate the (perturbed) rewards after a certain number of bits. The following lemma assures that this is possible (with high probability) without changing the optimal strategies, as long as the rounded rewards and the true rewards are close enough. Before we state the lemma, it is useful to observe that, if the rewards are continuous, independently distributed random variables, then, for any two distinct situations s and s', we have $\Pr(\mu_u(\mathcal{G}(s)) = \mu_u(\mathcal{G}(s'))) = 0$ if and only if $\rho(s) \neq \rho(s')$. Thus for the structurally ergodic case, with probability one, two distinct situations result in two distinct values. On the other hand, in the general case, there might be many optimal situations, but all of them lead to the same limiting distribution.

Given a strategy $s_W \in S_W$ of White we call a *uniform best response* (UBR) of Black any strategy $s_B^* \in S_B$, such that $\mu_u(\mathcal{G}(s_W, s_B^*)) \leq \mu_u(\mathcal{G}(s_W, s_B))$ for all $s_B \in S_B$. Similarly, a UBR of White is defined. (Note that the existence of such a UBR is an immediate corollary of the existence of uniformly optimal situations in BWR-games.) We denote by $\mathrm{UBR}_{\mathcal{G}}(s_W)$ and $\mathrm{UBR}_{\mathcal{G}}(s_B)$ the sets of uniform best responses in $\mathcal{G}$, corresponding to strategies s_W and s_B, respectively.

Lemma 7. *Let $\mathcal{G} = (G = (V, E), P, r)$, $\mathcal{G}' = (G = (V, E), P, r')$ be two BWR-games such that $r = (r_e)_{e \in E}$ is a vector of independent continuous random variables with maximum density ϕ, and $\|r' - r\|_\infty \leq \varepsilon$, for some given $\varepsilon > 0$. Let $\theta := \frac{2n^3\varepsilon\phi}{\gamma(\mathcal{G})^3}$. Then, the following holds for any situation s:*

i. $\Pr(s$ *is not uniformly optimal in* $\mathcal{G}' \mid s$ *is uniformly optimal in* $\mathcal{G}) \leq 2\theta$;
ii. $\Pr(s$ *is not uniformly optimal in* $\mathcal{G} \mid s$ *is uniformly optimal in* $\mathcal{G}') \leq 2\theta$.

Still, it can happen that rounding results in different optimal strategies. How can we be sure that the solution obtained from the rounded rewards is also optimal for the game with the true rewards? Step 6 in Algorithm 3 is one way to do this. The basic idea is as follows: Let $\tilde{s}$ be a uniformly optimal situation in the rounded game. Lemma 7 says that with high probability $\tilde{s}$ is a uniformly optimal situation in $\mathcal{G}$, and hence it is also uniformly optimal in any game on the same graph and transition matrix, but with rewards lying in a small interval around the rounded rewards. Thus, we create $|E|$ copies of the truncated game; in each copy the reward on a single arc is perturbed by a certain amount within this small interval. If $\tilde{s}$ is uniformly optimal in all these games, then it is also uniformly optimal for all rewards from that small interval. The following lemma justifies the correctness of this certificate.

Lemma 8. *Let $\tilde{\mathcal{G}} = (G = (V, E), P, \tilde{r})$ be a BWR-game and u be an arbitrary vertex. Consider a situation $\tilde{s} = (\tilde{s}_W, \tilde{s}_B)$ such that, for all $e \in E$, $\tilde{s}$ is optimal in the game $(\tilde{\mathcal{G}}_{e,\varepsilon}, u)$ (defined in (4)). Then $\tilde{s}$ is also optimal in $(\mathcal{G} = (G, P, r), u)$, for any r such that $\|r - \tilde{r}\|_\infty \leq \varepsilon$.*

Now, we have all ingredients to prove that Algorithm 3 solves, in uniformly optimal strategies and in smoothed polynomial time, any BWR-game on a graph which admits a pseudo-polynomial algorithm and have a constant number of random vertices. This establishes Theorem 2.

References

1. D. Andersson and P. B. Miltersen. The complexity of solving stochastic games on graphs. In *ISAAC*, volume 5878 of *LNCS*, pages 112–121, 2009.
2. R. Beier and B. Vöcking. Typical properties of winners and losers in discrete optimization. *SIAM J. Comput.*, 35(4):855–881, 2006.
3. E. Boros, K. M. Elbassioni, V. Gurvich, and K. Makino. A pumping algorithm for ergodic stochastic mean payoff games with perfect information. In *IPCO*, pages 341–354, 2010.
4. 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*, volume 39 of *Annals of Mathematics Studies*, pages 179–187. Princeton University Press, 1957.
5. 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.
6. A. J. Hoffman and R. M. Karp. On nonterminating stochastic games. *Management Science, Series A*, 12(5):359–370, 1966.
7. A. V. Karzanov and V. N. Lebedev. Cyclical games with prohibition. *Mathematical Programming*, 60:277–293, 1993.
8. T. M. Liggett and S. A. Lippman. Stochastic games with perfect information and time-average payoff. *SIAM Review*, 4:604–607, 1969.
9. H. Mine and S. Osaki. *Markovian decision process*. Elsevier, 1970.
10. K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.
11. A. Roth, M.-F. Balcan, A. Kalai, and Y. Mansour. On the equilibria of alternating move games. In *SODA*, pages 805–816, 2010.
12. D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004.
13. D. A. Spielman and S.-H. Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *C. ACM*, 52(10):76–84, 2009.
14. S. Vorobyov. Cyclic games and linear programming. *Discrete Appl. Math.*, 156(11):2195–2231, 2008.
15. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoret. Comput. Sci.*, 158(1-2):343–359, 1996.