Stochastic Mean Payoff Games: Smoothed Analysis and Approximation Schemes*

Endre Boros† Khaled Elbassioni‡ Mahmoud Fouz§

Vladimir Gurvich* Kazuhisa Makino¶ Bodo Manthey‖

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.

We show that the existence of a pseudo-polynomial algorithm for BWR-games with a 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

*A preliminary version appeared in the proceedings of ICALP 2011 [5]. 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.

†RUTCOR, Rutgers University, boros/gurvich@rutcor.rutgers.edu

‡Max-Planck-Institute for Informatics, elbassio@mpi-inf.mpg.de

§Saarland University, Department of Computer Science, mfouz@cs.uni-saarland.de

¶University of Tokyo, Graduate School of Information Science and Technology, makino@mist.i.u-tokyo.ac.jp

‖University of Twente, Department of Applied Mathematics, b.manthey@utwente.nl

is that of a *Nash equilibrium*, which defines a state where no agent gains an advantage by changing to another strategy. Nash equilibria serve as predictions 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 lead to a Nash equilibrium. Still, the existence of Nash equilibria might be irrelevant in practice since their computation would take too long. (Finding mixed Nash equilibria in two-player games is PPAD-complete in general [9].) Thus, algorithmic aspects of game theory have gained a lot of interest. Following the dogma that only polynomial time algorithms are feasible algorithms, it is desirable to show polynomial time complexity for the computation of Nash equilibria.

We consider two-player zero-sum stochastic mean payoff games with perfect information. Computing Nash equilibria and the values of these games is in the intersection of NP and co-NP, but it is unknown whether it can be done in polynomial time. In cases where efficient algorithms are not known to exist, an approximate notion of Nash equilibria has been suggested. In an approximate Nash equilibrium, no agent can gain a *substantial* advantage by changing to another strategy. In this paper, we design approximation schemes for Nash equilibria (see Section 1.2 for a definition). Moreover, we advocate an alternative notion of tractability by considering the *smoothed complexity* (see Section 1.3 for a definition). In contrast to the usual worst-case complexity, smoothed complexity aims at analyzing the running-time of algorithms by means of typical instances rather than (artificially constructed) worst-case instances. By establishing smoothed polynomial complexity for computing Nash equilibria for mean payoff games, we argue that the computation of a Nash equilibrium is usually a tractable problem, except for some rare, pathological worst-case instances.

In the remainder of this section, we introduce the concepts used in this paper. Our results are summarized in Section 1.4. After that, we present our approximation schemes (Section 2) and the smoothed analysis (Section 3).

1.1. Stochastic Mean Payoff Games

1.1.1. Definition and Notation

The model that we consider are *stochastic mean payoff games* or *BWR-games* $\mathcal{G} = (G, P, r)$:

- $G = (V, E)$ is a directed graph that may have loops and multiple edges, but no terminal vertices, i.e., vertices of out-degree 0. The vertex set V of G is partitioned into three subsets $V = V_B \cup V_W \cup V_R$ that correspond to black, white, and random *positions*, respectively. The edges 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.

- P is the vector of probability distributions for all vertices $v \in V_R$ owned by nature. We assume that $\sum_{u:(v,u)\in E} p_{v,u} = 1$ for all $v \in V_R$ and $p_{(v,u)} > 0$ for all $v \in V_R$ and $(v, u) \in E$.

- r is the vector of rewards; each edge e has a local reward r_e.

Starting from some vertex $v_0 \in V$, a token is moved along one edge e in every round of the game. If the token is on a black vertex, Black selects an outgoing edge e and moves the token along e. If the token is on a white vertex, then White selects an outgoing edge e. In a random position $v \in V_R$, a move $e = (v, u)$ is chosen according to the probabilities $p_{(v,u)}$ of the outgoing edges of v. In all cases, BLACK pays WHITE the reward r_e on the selected edge e.

Starting from a given initial position $v_0 \in V$, the game yields an infinite walk $(v_0, v_1, v_2, \ldots)$ (called a *play*). Let b_i denote the reward $r_{(v_{i-1},v_i)}$ received by WHITE in step i. The *undiscounted limit average effective payoff* is defined as the *Cesàro average* $c = \liminf_{n\to\infty} \frac{\sum_{i=1}^{n} \mathbb{E}[b_i]}{n}$. WHITE's objective is to maximize c, while the objective of BLACK is the opposite, i.e., to minimize $\limsup_{n\to\infty} \frac{\sum_{i=1}^{n} \mathbb{E}[b_i]}{n}$.

Formally, a strategy $s_W \in S_W$ for WHITE is a mapping that assigns a move $(v, u) \in E$ to each position in V_W. We sometimes abbreviate $s_W(v) = (v, u)$ by $s_W(v) = u$. Strategies $s_B \in S_B$ for BLACK are analogously defined. A pair of strategies $s = (s_W, s_B)$ is called a *situation*. By abusing notation, let $s(v) = u$ if $v \in V_W$ and $s_W(v) = u$ or $v \in V_B$ and $s_B(v) = u$.

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 in the obvious way:

$$p_{(v,u)}(s) = \begin{cases} 1 & \text{if } v \in V_W \cup V_B \text{ and } u = s(v), \\ 0 & \text{if } v \in V_W \cup V_B \text{ and } u \neq s(v), \text{ and} \\ p_{(v,u)} & \text{if } v \in V_R. \end{cases}$$

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 will use $(\mathcal{G}, v_0)$ to denote the game starting from v_0. We will simply write $\rho(s)$ for $\rho(s, v_0)$ if 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 $R = R(\mathcal{G}) = r^+ - r^-$ be the size of the range.

1.1.2. Strategies and Saddle Points

If we consider $c_{v_0}(s)$ for all possible situations s, we obtain a matrix game $C_{v_0} : S_W \times S_B \to \mathbb{R}$, with entries $C_{v_0}(s_W, s_B) = c_{v_0}(s_W, s_B)$. It is known that every such game has a *saddle point* in pure strategies [13,22]. Such a saddle point defines an equilibrium state in which no player has an incentive to switch to another strategy. The value at that state coincides with the limiting payoff in the corresponding BWR-game [13, 22].

We call a pair of strategies optimal if they correspond to a saddle point. It is well-known that there exists optimal strategies (s_W^*, s_B^*) that do not depend on the starting position v_0. Such strategies are called *uniformly optimal.* Of course there might be several optimal strategies, but 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. Note also that for an arbitrary situation s, $\mu_{v_0}(\mathcal{G}(s))$ denotes the effective payoff $c_{v_0}(s)$ in the Markov chain $\mathcal{G}(s)$.

A game is called *ergodic* if the value does not depend on the initial position. A game is called *structurally ergodic* if it is ergodic for *any* choice of rewards.

An algorithm is said to solve the game if it computes an optimal pair of strategies.

1.1.3. Previous Results

BWR-games are an equivalent formulation [15] of the stochastic games with perfect information and mean payoff that were introduced in 1957 by Gillette [13]. This generalizes a variety of games and problems: BWR-games without random vertices ($V_R = \emptyset$) are called *cyclic* or *mean payoff* games [15]; 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 [24]. If both are empty ($V_B = V_W = \emptyset$), we get a *weighted Markov chain*. If $V = V_W$ or $V = V_B$, we obtain the *minimum mean-weight cycle problem*, which can be solved in polynomial time [20].

If all rewards are 0 except for m terminal loops, we obtain the so-called *Backgammon-like* games [8]. The special case $m = 1$ is called *simple stochastic games* (SSGs), introduced by Condon [11, 12]. 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 has been shown that Gillette games (and hence BWR-games [6]) are equivalent to SSGs under polynomial-time reductions [1]. Thus, by recent results of Halman [16], all these games can be solved in randomized strongly subexponential time $2^{O(\sqrt{n_d \log n_d})} \operatorname{poly}(|V|)$, where $n_d = |V_B| + |V_W|$ is the number of *deterministic* vertices.

Besides their many applications [18, 23], all these games are of interest to complexity theory: The decision problem "whether the value of a BW-game is positive" is in the intersection of NP and co-NP [21, 35]. Yet, no polynomial algorithm is known even in this special case. A similar complexity claim holds for SSGs and BWR-games [1, 6]. We refer to Vorobyov [34] for a recent survey. On the other hand, there exists algorithms that solve BW-games in practice very fast [15]. The situation for these games is thus comparable to linear programming before the discovery of the ellipsoid method: Linear programming was known to lie in the intersection of NP and co-NP, and the simplex method proved to be fast in practice. In fact, a polynomial algorithm for linear programming in the *unit cost model* would already imply a polynomial algorithm for BW-games [30]. Spielman and Teng [31] introduced smoothed analysis to explain the practical performance of the simplex method. We further enforce the analogy between linear programming and BWR-games by showing that BWR-games have smoothed polynomial complexity.

While there are numerous pseudo-polynomial algorithms known for BW-games [15, 26, 35], pseudo-polynomiality for BWR-games (with no restriction on the number of random nodes) is in fact equivalent to polynomiality [1]. Gimbert and Horn [14] have shown that simple stochastic games on k random vertices can be solved in time $O(k!(|V||E| + L))$, where L is the maximum bit length of a transition probability. (Even though BWR-games are polynomially reducible to simple stochastic games, under this reduction the number k of random vertices becomes a polynomial in n, even if the original BWR-game has a constant number of random vertices.) Recently, a pseudo-polynomial algorithm was given 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* [7]. However, the existence of a similar algorithm for the non-ergodic case or a non-constant number of random vertices remains open, as the approach by Boros et al. [7] does not seem to generalize to these cases.

1.2. Approximation and Approximate Equilibria

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

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

An alternative concept of an approximate equilibrium are *ε-relative equilibria*. They are determined by a situation (s_W^*, s_B^*) such that

$$\begin{aligned} &\max_{s_W} \mu_v(\mathcal{G}(s_W, s_B^*)) \leq (1+\varepsilon)\mu_v(\mathcal{G}(s_W^*, s_B^*)) \\ \text{and} \quad &\min_{s_B} \mu_v(\mathcal{G}(s_W^*, s_B)) \geq (1-\varepsilon)\mu_v(\mathcal{G}(s_W^*, s_B^*)). \end{aligned} \tag{2}$$

Note that, for sufficiently small ε, an ε-relative approximation implies a $\Theta(\varepsilon)$-relative equilibrium, and vice versa. Thus, in what follows, we will use these notions interchangeably. For relative approximations and relative equilibria, we assume that the rewards are non-negative integers.

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

$$\max_{s_W} \mu_v(\mathcal{G}(s_W, s_B^*)) \leq \mu_v(\mathcal{G}) + \varepsilon \quad \text{and} \quad \min_{s_B} \mu_v(\mathcal{G}(s_W^*, s_B)) \geq \mu_v(\mathcal{G}) - \varepsilon, \tag{3}$$

or, equivalently, for an ε-absolute equilibrium:

$$\max_{s_W} \mu_v(\mathcal{G}(s_W, s_B^*)) \leq \mu_v(\mathcal{G}(s_W^*, s_B^*)) + \varepsilon \quad \text{and} \quad \min_{s_B} \mu_v(\mathcal{G}(s_W^*, s_B)) \geq \mu_v(\mathcal{G}(s_W^*, s_B^*)) - \varepsilon. \tag{4}$$

Again, an ε-absolute approximation implies a 2ε-absolute equilibrium. In the other direction, an ε-absolute equilibrium implies an ε-absolute approximation. For absolute equilibria and absolute approximations, we assume that the rewards come from the interval $[-1, 1]$.

A situation (s_W^*, s_B^*) that satisfies (1) is called relatively ε-optimal. If a situation satisfies (3), it is called absolutely ε-optimal. In the following, we will drop the specification of absolute and relative if it is clear from the context. If the pair (s_W^*, s_B^*) is ε-optimal for any starting position, it is called *uniformly* ε-optimal.

An algorithm for approximating the values of the game is said to be a fully polynomial-time approximation scheme (FPTAS) if the running-time depends polynomially on the input size and $1/\varepsilon$. In what follows, we assume without loss of generality that $1/\varepsilon$ is an integer.

1.3. Smoothed Analysis

For quite a few algorithms, the classical worst-case analysis is far too pessimistic. Worst-case analysis is often dominated by artificially constructed worst-case instances, that do not reflect typical instances. An alternative measure is average-case analysis, where performance is measured by means of random instances. But the drawback of average-case analysis is that it is dominated by random instances. Random instances usually have very special properties with high probability, and these properties are often not shared with typical instances.

To overcome the drawbacks of worst-case and average-case complexity and to explain the speed of the simplex method for linear programming, Spielman and Teng have introduced smoothed analysis [31]. Smoothed analysis is a hybrid of worst-case and average-case analysis: An adversary specifies an instance, which is then subjected to a small amount of random noise. The smoothed complexity is the maximum expected running-time that the adversary can achieve by his choice of an instance. Smoothed analysis rules out the drawbacks of both average-case and worst-case analysis and often allows more realistic conclusions about the performance of an algorithm or the complexity of a problem. It takes into account that realistic data is often subjected to a small amount of random noise, be it from measurement or rounding errors or from some arbitrary unknown circumstances. Smoothed analysis has originally been invented to explain the practical performance of the simplex method [31]. Since then, smoothed analysis has been applied successfully to a variety of algorithms and problems [2, 27, 33]. We refer to Spielman and Teng for a survey [32].

A generalization of smoothed analysis has been introduced by Beier and Vöcking [3]: They went from the (original) two-step model (an adversary chooses the instance, which is then perturbed) to a one-step model. In the one-step model, the adversary specifies the density functions according to which the numbers of the instance are drawn. In this case, the perturbation parameter ϕ restricts the adversary to density functions that are bounded by ϕ: The larger ϕ, the more powerful the adversary. For instance, in case of Gaussian perturbation, ϕ is proportional to the inverse of the standard deviation.

To characterize integer programs that can typically be solved in polynomial time, Beier and Vöcking introduced the notion of polynomial smoothed complexity [3]. Their notion is inspired by the notion of average polynomial time [4]. 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

$$\forall \phi \geq 1, \forall n \in \mathbb{N} : \max_{\vec{f} \in \mathcal{D}_N(\phi)} \mathbb{E}_{X \sim \vec{f}}(T(X)^\alpha) = O(n\phi) \,. \tag{5}$$

Here, $\mathcal{D}_N(\phi)$ denotes all possible vectors of density functions bounded by ϕ for instances of size n, and X is an instance drawn according to $\vec{f}$. Another way to phrase smoothed polynomial running-time is that there exists a polynomial $P(n, \phi, 1/\varepsilon)$ such that the probability that $\mathcal{A}$ exceeds a running-time of $P(n, \phi, 1/\varepsilon)$ is at most ε. Note that smoothed polynomial running-time does not always give polynomial expected running-time. To get the latter, the α must be placed outside the expectation. The reason for defining smoothed polynomial running-time in this way is that it makes the notion robust against, e.g., simulation on a slower machine. (We refer to Trevisan and Bogdanov for a discussion of this phenomenon [4].)

1.4. Our Results

1.4.1. Approximation Schemes for BWR-Games

As for approximation schemes, the only result we are aware of is the observation made by Roth et al. [29] 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 [15, 26, 35].

In this paper, we generalize this result 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 admits a pseudo-polynomial algorithm $\mathbb{A}$. This means that $\mathbb{A}$ 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. For instance, digraphs without random vertices (i.e., BW-games) are one possible class $\mathbb{G}$. The same holds for digraphs that have a constant number of random nodes and are *structurally ergodic* [17]. 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 this paper, we will assume that the number of random vertices k is bounded by a constant.

The following theorems says that a pseudo-polynomial algorithm can be turned into an *absolute* approximation scheme.

Theorem 1.1. *Let $\mathbb{G}$ be a class of digraphs that admits a pseudo-polynomial algorithm. For any $\varepsilon > 0$, there is an algorithm that returns, for any given BWR-game on $G \in \mathbb{G}$ with rewards in $[-1, 1]$, a pair of strategies that approximates the value within an* absolute *error of ε. The running-time of the algorithm is bounded by* $\text{poly}(n, 1/p_{\min}, 1/\varepsilon)$.

We also obtain an approximation scheme with a *relative* error.

Theorem 1.2. *Let $\mathbb{G}$ be a class of digraphs that admits a pseudo-polynomial algorithm. For any $\varepsilon > 0$, there is an algorithm that returns, for any given BWR-game on $G \in \mathbb{G}$ with non-negative integral rewards and rational transition probabilities, a pair of strategies that approximates the value within a* relative *error of ε. The running-time of the algorithm is bounded by* $\text{poly}(n, \log R, 1/\varepsilon)$.

Optimal strategies that are independent of the starting vertex are called *uniformly optimal strategies.* Our reduction in Theorem 1.1, unlike Theorem 1.2, has the property that if the pseudo-polynomial algorithm returns uniformly optimal strategies, then so does the approximation scheme. For BW-games, i.e., the special case without random vertices, we can also strengthen the result of Theorem 1.2 to return a pair of strategies that is uniformly ε-optimal.

In deriving these approximation schemes from a pseudo-polynomial algorithm, we face two main technical challenges that distinguish the computation of ε-equilibria of BWR-games from similar standard techniques used in optimization: First, 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, in order to obtain a *fully polynomial-time approximation scheme* (FPTAS) with an absolute guarantee whose running-time is independent of D, we have to truncate the probabilities and bound the change in the game value. But this is a *non-linear* function of D. Second, in order to obtain an FPTAS with a relative guarantee, one needs (as often in optimization) a (trivial) lower/upper bound on the optimum value. This, however, 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. This is because we have to output just a single pair of strategies which guarantees ε-optimality from any starting position.

In order to resolve the first issue, we analyze the change in the game values and optimal strategies if the rewards or transition probabilities are changed. Roughly speaking, we use results from Markov chain perturbation theory to show that if the probabilities are perturbed by a small error δ, then the change in the game value is $O(\delta^2 n^3 / p_{\min}^{2k})$ (see Section 2.1). The second issue is resolved through repeated applications of the pseudo-polynomial algorithm on a truncated game. After each such application we have one of the following situations: Either the value of the game has already been approximated within the required accuracy or it is guaranteed that the range of the rewards can be shrunk by a constant factor without changing the value of the game (see Sections 2.3 and 2.4).

Since BW-games and structurally ergodic BWR-games with constant number of random vertices admit pseudo-polynomial algorithms, we obtain the following results.

Corollary 1.3. *(i) There is an FPTAS that solves, within a relative error guarantee, in uniformly ε-optimal strategies, any BW-games with non-negative (rational) rewards.*

(ii) There is an FPTAS that solves, within an absolute error guarantee, in uniformly ε-optimal strategies, any structurally ergodic BWR-game with a constant number of random nodes, $1/p_{\min} = \text{poly}(n)$, and rewards in $[-1, 1]$.

(iii) There is an FPTAS that solves, within a relative error guarantee, in uniformly ε-optimal strategies, any structurally ergodic BWR-game with a constant number of random nodes, $1/p_{\min} = \text{poly}(n)$, and non-negative rational rewards.

Note that (i) strengthens the absolute FPTAS for BW-games [29], and (ii) and (iii) enlarge the class of games for which an FPTAS exists.

1.4.2. Smoothed Analysis for BWR-Games

We further show that typical instances of digraphs, taken from classes that admit pseudo-polynomial algorithms, can be solved in polynomial time. Towards this end, we do a smoothed analysis using the one-step model introduced by Beier and Vöcking [3]: Given an upper bound ϕ for the densities, an adversary specifies a BWR game $\mathcal{G}$ including the probabilities for the out-going arcs of the random nodes and together with one density function for each arc. 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.

Theorem 1.4. *Let $\mathbb{G}$ be a class of digraphs that admit a pseudo-polynomial algorithm. There is an algorithm that solves any BWR-game on any $G \in \mathbb{G}$ with rational transition probabilities and $D = \text{poly}(n)$ in smoothed polynomial time.*

Our proof of Theorem 1.4 follows the general paradigm introduced by Beier and Vöcking [3] for using pseudo-polynomial algorithms to analyze the smoothed complexity of integer programs. However, in the case of BWR-games, the situation becomes more complicated by the following two facts. First, we have to deal with two players who want to optimize the objective function in different directions. Second, the coefficients of the objectives are not given explicitly, but they correspond to the limiting distributions in the Markov chains corresponding to the different strategies. To prove that BWR-games can be solved in smoothed polynomial time, we first need a new isolation lemma. In contrast to the existing isolation lemmas used in smoothed analysis of optimization problems, our isolation lemma has to deal with two players who optimize the same objective function in two different directions. Second, our procedure for certifying that the solution found is indeed the optimal solution is considerably more involved. The reason is again that we have two competing players, which requires careful rounding of the coefficients in order to certify optimality.

Chen et al. [9] have analyzed the smoothed complexity of the Lemke-Howson algorithm for computing equilibria in bimatrix games. They have shown that the Lemke-Howson algorithm – and also any other algorithm – has smoothed polynomial complexity only if all problems in PPAD can be solved in randomized polynomial time. In contrast to their negative result, our smoothed analysis shows that equilibria of BWR-games can typically be computed efficiently.

As a corollary to Theorem 1.4 above, we obtain the following results.

Corollary 1.5. *(i) BW-games can be solved in smoothed polynomial time.*

(ii) Structurally ergodic BWR-games with $D = \mathrm{poly}(n)$ can be solved in smoothed polynomial time.

2. Approximation Schemes

When considering relative errors, we assume that the rewards are non-negative. If we consider absolute errors, then we assume that the rewards lie in a certain range, say, $[-1, 1]$. Under such assumptions, the notion of relative approximation becomes stronger. Indeed, an ε-relative approximation of the game $\hat{\mathcal{G}}$ with local rewards given by $\hat{r} = r + 1 \geq 0$ (where the addition and comparison is meant component-wise) implies strategies (s_W^*, s_B^*) that satisfy

$$\begin{aligned}\max_{s_W} \mu_v(\mathcal{G}(s_W, s_B^*)) &= \max_{s_W} \mu_v(\hat{\mathcal{G}}(s_W, s_B^*)) - 1 \leq (1+\varepsilon)\mu_v(\hat{\mathcal{G}}) - 1 \\ &= \mu_v(\mathcal{G}) + \varepsilon\mu_v(\mathcal{G}) + \varepsilon \leq \mu_v(\mathcal{G}) + 2\varepsilon\end{aligned}$$

and

$$\begin{aligned}\min_{s_B} \mu_v(\mathcal{G}(s_W^*, s_B)) &= \min_{s_B} \mu_v(\hat{\mathcal{G}}(s_W^*, s_B)) - 1 \geq (1-\varepsilon)\mu_v(\hat{\mathcal{G}}) - 1 \\ &= \mu_v(\mathcal{G}) - \varepsilon\mu_v(\mathcal{G}) - \varepsilon \geq \mu_v(\mathcal{G}) - 2\varepsilon.\end{aligned}$$

This is because $\mu_v(\hat{\mathcal{G}}(s)) = \mu_v(\mathcal{G}(s)) + 1$ for any situation s and $\mu_v(\mathcal{G}) \leq 1$. Thus, we obtain a 2ε-absolute approximation for the value of the original game.

2.1. The Effect of Perturbation

Our approximation schemes are based on the following three lemmas. The first one (which is known) says that a linear change in the rewards will correspond to a linear change in the game value. In our approximation schemes, we truncate and scale the rewards to be able to run the pseudo-polynomial algorithm in polynomial time. We will then need the lemma to bound the error in the game value resulting from the truncation.

Lemma 2.1. *Let $\mathcal{G} = (G = (V, E), P, r)$ be a BWR-game. Let $\theta_1, \gamma_1, \theta_2, \gamma_2$ be constants such that $\theta_1, \theta_2 > 0$. Let $\hat{\mathcal{G}}$ be a game $(G = (V, E), P, \hat{r})$ with $\theta_1 r + \gamma_1 \leq \hat{r} \leq \theta_2 r + \gamma_2$ (inequalities of this form are meant component-wise). Then for any $v \in V$, we have $\theta_1\mu_v(\mathcal{G}) + \gamma_1 \leq \mu_v(\hat{\mathcal{G}}) \leq \theta_2\mu_v(\mathcal{G}) + \gamma_2$. Moreover, if $(\hat{s}_W, \hat{s}_B)$ is an absolutely ε-optimal situation in $(\hat{\mathcal{G}}, v)$, then*

$$\begin{aligned}\max_{s_W} \mu_v(\mathcal{G}(s_W, \hat{s}_B)) &\leq \frac{\theta_2\mu_v(\mathcal{G}) + \gamma_2 - \gamma_1 + \varepsilon}{\theta_1} \quad \textit{and} \\ \min_{s_B} \mu_v(\mathcal{G}(\hat{s}_W, s_B)) &\geq \frac{\theta_1\mu_v(\mathcal{G}) + \gamma_1 - \gamma_2 - \varepsilon}{\theta_2}.\end{aligned} \tag{6}$$

Proof. This uses only standard techniques, and we give the proof only for completeness. Let (s_W^*, s_B^*) and $(\hat{s}_W, \hat{s}_B)$ be pairs of optimal strategies for $(\mathcal{G}, v)$ and $(\hat{\mathcal{G}}, v)$, respectively. Denote by $\rho^*, \hat{\rho}, \rho'$, and ρ'' the (arc) limiting distributions for the Markov chains starting from v_0 and corresponding to pairs (s_W^*, s_B^*), $(\hat{s}_W, \hat{s}_B)$, $(s_W^*, \hat{s}_B)$, and $(\hat{s}_W, s_B^*)$, respectively. By the definition of optimal strategies and the facts that $\|\rho'\|_1 = \|\rho''\|_1 = 1$ (because they are probability distributions), we have the following series of inequalities:

$$\mu_v(\hat{\mathcal{G}}) = (\hat{\rho})^T \hat{r} \geq (\rho')^T \hat{r} \geq \theta_1 (\rho')^T r + \gamma_1 \geq \theta_1 (\rho^*)^T r + \gamma_1 = \theta_1 \mu_v(\mathcal{G}) + \gamma_1 \quad \text{and}$$
$$\mu_v(\hat{\mathcal{G}}) = (\hat{\rho})^T \hat{r} \leq (\rho'')^T \hat{r} \leq \theta_2 (\rho'')^T r + \gamma_2 \leq \theta_2 (\rho^*)^T r + \gamma_2 = \theta_2 \mu_v(\mathcal{G}) + \gamma_2.$$

To see the first bound in (6), note that for any s_W, we have $\mu_v(\mathcal{G}(s_W, \hat{s}_B)) \leq \frac{1}{\theta_1}(\mu_v(\hat{\mathcal{G}}(s_W, \hat{s}_B)) - \gamma_1)$. Also, by the ε-optimality of $\hat{s}_W$ in $(\hat{\mathcal{G}}, v)$, we have $\mu_v(\hat{\mathcal{G}}(s_W, \hat{s}_B)) \leq \mu_v(\hat{\mathcal{G}}) + \varepsilon \leq \theta_2 \mu_v(\mathcal{G}) + \gamma_2 + \varepsilon$. The first bound in (6) follows. The second bound can be shown similarly. □

The second lemma, which is new as far as we are aware, states that if we truncate the transition probabilities within a small error ε, then the change in the game value is bounded by $O(\varepsilon^2 n^3 / p_{\min}^{2k})$. More precisely, for a BWR-game $\mathcal{G}$ and a constant $\varepsilon > 0$, define

$$\delta(\mathcal{G}, \varepsilon) := \left(\frac{\varepsilon n^2}{2} \left(\frac{p_{\min}}{2}\right)^{-k} \left(\varepsilon n k(k+1) \left(\frac{p_{\min}}{2}\right)^{-k} + 3k + 1 \right) + \varepsilon n \right) r_*, \tag{7}$$

where $n = n(\mathcal{G})$, $p_{\min} = p_{\min}(\mathcal{G})$, $k = k(\mathcal{G})$, and $r_* = r_*(\mathcal{G}) := \max\{|r_+(\mathcal{G})|, |r_-(\mathcal{G})|\}$.

Lemma 2.2. *Let $\mathcal{G} = (G = (V, E), P, r)$ be a BWR-game with $r \in [-1, 1]^E$, and let $\varepsilon \leq p_{\min}/2 = p_{\min}(\mathcal{G})/2$ be a positive constant. Let $\hat{\mathcal{G}}$ be a game $(G = (V, E), \hat{P}, r)$ with $\|P - \hat{P}\|_\infty \leq \varepsilon$. Then we have $|\mu_v(\mathcal{G}) - \mu_v(\hat{\mathcal{G}})| \leq \delta(\mathcal{G}, \varepsilon)$ for any $v \in V$. Moreover, if the pair $(\tilde{s}_W, \tilde{s}_B)$ is absolutely ε'-optimal in $(\hat{\mathcal{G}}, v)$, then it is absolutely $(\varepsilon' + 2\delta(\mathcal{G}, \varepsilon))$-optimal in $(\mathcal{G}, v)$.*

Proof. We apply Lemma A.3. Let (s_W^*, s_B^*) and $(\hat{s}_W, \hat{s}_B)$ be pairs of optimal strategies for $(\mathcal{G}, v)$ and $(\hat{\mathcal{G}}, v)$, respectively. Write $\delta = \delta(\mathcal{G}, \varepsilon)$. Then optimality and Lemma A.3 imply the following two series of inequalities:

$$\begin{aligned}
\mu_v(\hat{\mathcal{G}}) &= \mu_v(\hat{\mathcal{G}}(\hat{s}_W, \hat{s}_B)) \geq \mu_v(\hat{\mathcal{G}}(s_W^*, \hat{s}_B)) \\
&\geq \mu_v(\mathcal{G}(s_W^*, \hat{s}_B)) - \delta \geq \mu_v(\mathcal{G}(s_W^*, s_B^*)) - \delta = \mu_v(\mathcal{G}) - \delta \quad \text{and} \\
\mu_v(\hat{\mathcal{G}}) &= \mu_v(\hat{\mathcal{G}}(\hat{s}_W, \hat{s}_B)) \leq \mu_v(\hat{\mathcal{G}}(\hat{s}_W, s_B^*)) \\
&\leq \mu_v(\mathcal{G}(\hat{s}_W, s_B^*)) + \delta \leq \mu_v(\mathcal{G}(s_W^*, s_B^*)) + \delta = \mu_v(\mathcal{G}) + \delta.
\end{aligned}$$

To see the second claim, note that for any $s_W \in S_W$, we have

$$\mu_v(\mathcal{G}(s_W, \tilde{s}_B)) \leq \mu_v(\hat{\mathcal{G}}(s_W, \tilde{s}_B)) + \delta \leq \mu_v(\hat{\mathcal{G}}(\hat{s}_W, \hat{s}_B)) + \varepsilon' + \delta \leq \mu_v(\mathcal{G}) + \varepsilon' + 2\delta.$$

Similarly, we can show that $\mu_v(\mathcal{G}(\tilde{s}_W, s_B)) \geq \mu_v(\mathcal{G}) - \varepsilon' - 2\delta$ for all $s_B \in S_B$. □

Since we assume that the running-time of the pseudo-polynomial algorithm for $\mathbb{G}$ depends on the common denominator D of the transition probabilities, we have to truncate the probabilities to remove this dependence on D. By Lemma 2.2, the value of the game does not change too much after such a truncation.

The third result that we need concerns relative approximation. The main idea is to use the pseudo-polynomial algorithm to test whether the value of the game is larger than a certain threshold. If it is, we get already a good relative approximation. Otherwise, the next lemma says that we can reduce all large rewards without changing the value of the game.

Lemma 2.3. *Let $\mathcal{G} = (G = (V, E), P, r)$ be a BWR-game with $r \geq 0$, and let v be any vertex with $\mu_v(\mathcal{G}) < t$. Suppose that $r_e \geq t' = ntp_{\min}^{-(2k+1)}$ for some $e \in E$. Let $\hat{\mathcal{G}} = (G = (V, E), P, \hat{r})$, where $\hat{r}_e \geq (1+\varepsilon)t'$ for some $\varepsilon \geq 0$, and $\hat{r}_{e'} = r_{e'}$ for all $e' \neq e$. Then $\mu_v(\hat{\mathcal{G}}) = \mu_v(\mathcal{G})$, and any relatively ε-optimal situation in $(\hat{\mathcal{G}}, v)$ is also ε-optimal in $(\mathcal{G}, v)$.*

Proof. Let $s^* = (s_W^*, s_B^*)$ be an optimal strategy for $(\mathcal{G}, v)$. This means that $\mu_v(\mathcal{G}) = \mu_v(\mathcal{G}(s^*)) = \rho(s^*)^T r < t$. By Lemma A.1 says that $\rho_e(s^*) > 0$ implies $\rho_e(s^*) \geq p_{\min}^{2k+1}/n$. Hence, $r_e \rho_e(s^*) \leq \rho(s^*)^T r = \mu_v(\mathcal{G}) < t$ implies that $r_e < t'$. We conclude that $\rho_e(s^*) = 0$, and hence $\mu_v(\hat{\mathcal{G}}(s^*)) = \mu_v(\mathcal{G})$.

Since $\hat{r} \leq r$, we have $\mu_v(\hat{\mathcal{G}}(s)) \leq \mu_v(\mathcal{G}(s))$ for all situations s. In particular, for any $s_W \in S_W$,

$$\mu_v(\hat{\mathcal{G}}(s_W, s_B^*)) \leq \mu_v(\mathcal{G}(s_W, s_B^*)) \leq \mu_v(\mathcal{G}(s_W^*, s_B^*)) = \mu_v(\hat{\mathcal{G}}(s_W^*, s_B^*)).$$

We claim that also $\mu_v(\hat{\mathcal{G}}(s_W^*, s_B)) \geq \mu_v(\hat{\mathcal{G}}(s_W^*, s_B^*))$ for all $s_B \in S_B$. Indeed, if there is a strategy s_B for BLACK such that $\mu_v(\hat{\mathcal{G}}(s_W^*, s_B)) < \mu_v(\hat{\mathcal{G}}(s_W^*, s_B^*)) = \mu_v(\mathcal{G}) < t$, then, by the same argument as above, we must have $\rho_e(s_W^*, s_B) = 0$. This is since $\rho_e(s_W^*, s_B)(1+\varepsilon)t' \leq \rho_e(s_W^*, s_B)\hat{r}_e \leq \rho(s_W^*, s_B)^T \hat{r} = \mu_v(\hat{\mathcal{G}}(s_W^*, s_B)) < t$. This, however, implies $\mu_v(\mathcal{G}(s_W^*, s_B)) = \mu_v(\hat{\mathcal{G}}(s_W^*, s_B)) < \mu_v(\hat{\mathcal{G}}(s_W^*, s_B^*)) = \mu_v(\mathcal{G}(s_W^*, s_B^*))$, which is in contradiction to the optimality of s^* in $\mathcal{G}$. We conclude that (s_W^*, s_B^*) is also optimal in $\hat{\mathcal{G}}$ and hence $\mu_v(\hat{\mathcal{G}}) = \mu_v(\mathcal{G})$.

Suppose that $(\hat{s}_W, \hat{s}_B)$ is a relatively ε-optimal situation in $(\hat{\mathcal{G}}, v)$. Then $\rho_e(s_W, \hat{s}_B) = 0$ for any $s_W \in S_W$. Indeed,

$$\begin{aligned}\rho_e(s_W, \hat{s}_B)(1+\varepsilon)t' = \rho_e(s_W, \hat{s}_B)\hat{r}_e \leq \rho(s_W, \hat{s}_B)^T \hat{r} = \mu_v(\hat{\mathcal{G}}(s_W, \hat{s}_B)) \\ \leq (1+\varepsilon)\mu_v(\hat{\mathcal{G}}) = (1+\varepsilon)\mu_v(\mathcal{G}) < (1+\varepsilon)t,\end{aligned}$$

gives a contradiction with Lemma A.1 if $\rho_e(s_W, \hat{s}_B) > 0$. It follows that, for any $s_W \in S_W$, $\mu_v(\mathcal{G}(s_W, \hat{s}_B)) = \mu_v(\hat{\mathcal{G}}(s_W, \hat{s}_B)) \leq (1+\varepsilon)\mu_v(\mathcal{G})$. Furthermore, for any $s_B \in S_B$,

$$\mu_v(\mathcal{G}(\hat{s}_W, s_B)) \geq \mu_v(\hat{\mathcal{G}}(\hat{s}_W, s_B)) \geq (1-\varepsilon)\mu_v(\hat{\mathcal{G}}) = (1-\varepsilon)\mu_v(\mathcal{G}).$$

□

2.2. Absolute Approximation

Let $\mathbb{G}$ be a class of digraphs $G = (V, E)$ that admits a pseudo-polynomial algorithm $\mathbb{A}$, and let $\mathcal{G} = (G, P, r)$ 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 $(p_{(v,u)})_{u \in V}$ for each random node $v \in V_R$, as described in the following lemma.

Lemma 2.4. *Let $\alpha \in [0, 1]^n$ with $\|\alpha\|_1 = 1$. Let $B \in \mathbb{N}$ 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 $c_i \in \mathbb{N}$ is an integer;

(iii) for all $i = 1, \ldots, n$, $\alpha'_i > 0$ if and only $\alpha_i > 0$; and

(iv) $\|\alpha - \alpha'\|_\infty \leq 2^{-B}$.

Proof. This is straight-forward, and we include the proof only for completeness. Without loss of generality, we assume $\alpha_i > 0$ for all i. Initialize $\varepsilon_0 = 0$ and iterate for $i = 1, \ldots, n$: We set $\alpha'_i = \lfloor \alpha_i + \varepsilon_{i-1} \rceil_{2^{-B}}$ and $\varepsilon_i = \alpha_i + \varepsilon_{i-1} - \alpha'_i$. The construction implies (ii). Furthermore, $|\varepsilon_i| \leq 2^{-(B+1)}$ for all i, and $\varepsilon_n = \sum_i \alpha_i - \sum_i \alpha'_i$, which implies (i). Furthermore, $|\alpha_i - \alpha'_i| = |\varepsilon_i - \varepsilon_{i-1}| \leq 2^{-B}$, which implies (iv). Note finally that (iii) follows from (iv) since $\min_{i:\alpha_i>0}\{\alpha_i\} > 2^{-B}$. □

Lemma 2.5. *Let $\mathbb{G}$ be a class of graphs $G = (V, E)$ that admit a pseudo-polynomial algorithm $\mathbb{A}$ that solves, in (uniformly) optimal strategies, any BWR-game on G in time $\tau(n, D, R)$. Then for any $\varepsilon > 0$, there is an algorithm that solves, in (uniformly) absolutely ε-optimal strategies, any given BWR-game $\mathcal{G} = (G, P, r)$ in time bounded by $\tau\big(n, \frac{2^{2k+5}n^3k^2}{\varepsilon p_{\min}^{2k}}, \frac{4}{\varepsilon}\big)$.*

Proof. We apply $\mathbb{A}$ to the game $\tilde{\mathcal{G}} = (G, \tilde{P}, \tilde{r})$, where $\tilde{r} := \frac{2}{\varepsilon}\lfloor r \rceil_{\frac{\varepsilon}{2}}$. The probabilities $\tilde{P}$ are obtained from P by applying Lemma 2.4 with $B = \lceil \log_2(1/\varepsilon') \rceil$, where we select ε' such that $\delta(\mathcal{G}, \varepsilon') \leq \frac{\varepsilon}{4}$ (as defined by (7)). Note that all rewards in $\tilde{\mathcal{G}}$ are integer in the range $[-\frac{4}{\varepsilon}, \frac{4}{\varepsilon}]$. Note further that $\delta(\mathcal{G}, \varepsilon') \leq 4^{k+1}\varepsilon' n^3 k^2 p_{\min}^{-2k}$. Hence, $\delta(\mathcal{G}, \varepsilon') \leq \varepsilon/4$ for $\varepsilon' = \frac{\varepsilon p_{\min}^{2k}}{4^{k+2}n^3k^2}$. Since $D(\tilde{\mathcal{G}}) = 2^B$ and $R(\tilde{\mathcal{G}}) = 4/\varepsilon$, the statement about the running-time follows.

Let $\tilde{s}$ be the pair of (uniformly) optimal strategies returned by $\mathbb{A}$ on input $\tilde{\mathcal{G}}$. Let $\hat{\mathcal{G}}$ be the game $(G, \tilde{P}, r)$. Since $\|\tilde{r} - \frac{2}{\varepsilon}r\|_\infty \leq 1$, we can apply Lemma 2.1 (with $\theta_1 = \theta_2 = \frac{2}{\varepsilon}$ and $\gamma_1 = -\gamma_2 = -\frac{1}{2}$) to conclude that $\tilde{s}$ is a (uniformly) $\frac{\varepsilon}{2}$-optimal pair for $\hat{\mathcal{G}}$. Now we apply Lemma 2.2 and conclude that $\tilde{s}$ is (uniformly) $(\frac{\varepsilon}{2} + 2\delta(\mathcal{G}, \varepsilon'))$-optimal for $\mathcal{G}$. □

Note that the above technique yields an approximation algorithms with polynomial running-time only for $k = O(1)$, even if the pseudo-polynomial algorithm $\mathbb{A}$ works for arbitrary k.

2.3. Relative Approximation

Let $\mathbb{G}$ be a class of digraphs $G = (V, E)$ that admits a pseudo-polynomial algorithm $\mathbb{A}$. Let $\mathcal{G} = (G, P, r)$ be a BWR-game on G with non-negative integral rewards. This means that $r^- = 0$ and $\min_{e:r_e>0} r_e \geq 1$. The algorithm is given as Algorithm 1. The main idea is to truncate the rewards, scaled by a certain factor of $1/K$, and use the pseudo-polynomial algorithm on the truncated game $\hat{\mathcal{G}}$. If the value $\mu_w(\hat{\mathcal{G}})$ in the truncated game from the starting node w is large enough (step 8), 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 12); we invoke Lemma 2.3 for this. 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 2.4 with $B = \lceil \log(1/\varepsilon') \rceil$, where we select $\varepsilon' = \frac{p_{\min}^{2k}}{2^{2k+3}n^3k^2\theta}$ with $\theta = \theta(\mathcal{G}) := \frac{2(1+\varepsilon)(3+2\varepsilon)n}{\varepsilon p_{\min}^{2k+1}}$. Thus, we have $2\delta(\mathcal{G}, \varepsilon') \leq r_+(\mathcal{G})/\theta(\mathcal{G}) = K(\mathcal{G})$.

Lemma 2.6. *Let $\mathbb{G}$ be a class of graphs $G = (V, E)$ that admits a pseudo-polynomial algorithm $\mathbb{A}$ that solves, in (uniformly) optimal strategies, any BWR-game on G in time $\tau(n, D, R)$. Then, for any $\varepsilon \in (0, 1)$, there is an algorithm that solves, in relatively ε-optimal strategies,*

input: a BWR-game $\mathcal{G} = (G = (V, E), P, r)$, a starting vertex $w \in V$, and an accuracy ε
output: an ε-optimal pair $(\tilde{s}_W, \tilde{s}_B)$ for the game $(\mathcal{G}, w)$
1: **if** $r^+(\mathcal{G}) = 1$ **then**
2: $\quad\hat{\mathcal{G}} \leftarrow (G, \tilde{P}, r)$
3: $\quad$**return** $\mathbb{A}(\hat{\mathcal{G}}, v)$
4: $K \leftarrow r^+(\mathcal{G})/\theta(\mathcal{G})$
5: $\hat{r}_e \leftarrow \lfloor r_e/K \rfloor$ for $e \in E$
6: $\hat{\mathcal{G}} = (G, \tilde{P}, \hat{r})$
7: $(\tilde{s}_W, \tilde{s}_B) \leftarrow \mathbb{A}(\hat{\mathcal{G}}, w)$
8: **if** $\mu_w(\hat{\mathcal{G}}) \geq 3/\varepsilon$ **then**
9: $\quad$**return** $(\tilde{s}_W, \tilde{s}_B)$
10: **else**
11: $\quad$**for all** $e \in E$ **do**
12: $\quad\quad\tilde{r}_e \leftarrow \begin{cases} \lceil \frac{r^+}{2} \rceil & \text{if } r_e > \frac{r^+}{2(1+\varepsilon)} \text{ and} \\ r_e & \text{otherwise} \end{cases}$
13: $\quad\tilde{\mathcal{G}} \leftarrow (G, P, \tilde{r})$
14: $\quad$**return** FPTAS-BWR$(\tilde{\mathcal{G}}, w, \varepsilon)$

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

any BWR-game $(\mathcal{G} = (G, P, r), w)$ from any given starting position w in time

$$O\left(\left(\tau\left(n, \frac{4^{k+2} n^4 k^2 (1+\varepsilon)(3+2\varepsilon)}{\varepsilon p_{\min}^{4k+1}}, \frac{2(1+\varepsilon)(3+2\varepsilon)n}{\varepsilon p_{\min}^{2k+1}}\right) + \text{poly}(n)\right) \cdot \log(R)\right).$$

The running-time in the above theorem simplifies to $\text{poly}(n, 1/\varepsilon, 1/p_{\min}) \cdot \log R$ for $k = O(1)$.

Proof. The algorithm FPTAS-BWR$(\mathcal{G}, w, \varepsilon)$ is given as Algorithm 1. The bound on the running-time follows since, by step (12), each time we recurse on a game $\tilde{\mathcal{G}}$ with $r^+(\tilde{\mathcal{G}})$ reduced by a factor of at least half. Moreover, the rewards in the truncated game $\hat{\mathcal{G}}$ are non-negative integers with a maximum value of $r^+(\hat{\mathcal{G}}) \leq \theta$, and the smallest common denominator of the transition probabilities is at most $\tilde{D} := \frac{4^{k+2} n^4 k^2 (1+\varepsilon)(3+2\varepsilon)}{\varepsilon p_{\min}^{4k+1}}$. Thus the time taken by algorithm $\mathbb{A}$ for each recursive call is at most $\tau\big(n, \tilde{D}, \frac{2(1+\varepsilon)n}{\varepsilon p_{\min}^{2k+1}}\big)$.

What remains to be done is to argue by induction that the algorithm returns a pair $\tilde{s} = (\tilde{s}_W, \tilde{s}_B)$ of ε-optimal strategies. For the base case, note that since $\|P - \tilde{P}\|_\infty \leq \varepsilon'$ and $r_+(\mathcal{G}) = 1$, Lemma 2.2 implies that the pair $(\tilde{s}_W, \tilde{s}_B)$ returned in step 3 is absolutely ε''-optimal, where $\varepsilon'' = 2\delta(\mathcal{G}, \varepsilon') < \frac{\varepsilon p_{\min}^{2k+1}}{n}$. Lemma A.1 and the integrality of the non-negative rewards imply that, for any situation s, $\mu_w(\mathcal{G}(s)) \geq p_{\min}^{2k+1}/n$ if $\mu_w(\mathcal{G}(s)) > 0$. Thus, if $\mu_w(\mathcal{G}) > 0$, then $\varepsilon'' \leq \varepsilon \mu_w(\mathcal{G})$, and it follows that $(\tilde{s}_W, \tilde{s}_B)$ is relatively ε-optimal. On the other hand, if $\mu_w(\mathcal{G}) = 0$, then $\mu_w(\mathcal{G}(\tilde{s})) \leq \mu_w(\mathcal{G}) + \varepsilon'' < p_{\min}^{2k+1}/n$, implying that $\mu_w(\mathcal{G}(\tilde{s})) = 0$. Thus, we get an ε-approximation in both cases.

Now let us consider the general case. Note that $\frac{1}{K} \cdot r - 1 \leq \hat{r} \leq \frac{1}{K} \cdot r$, and $\|P - \tilde{P}\|_\infty \leq \varepsilon'$. Hence, by Lemmas 2.1 and 2.2, we have

$$K\mu_w(\hat{\mathcal{G}}) - 2\delta(\mathcal{G}, \varepsilon') \leq \mu_w(\mathcal{G}) \leq K\mu_w(\hat{\mathcal{G}}) + K + 2\delta(\mathcal{G}, \varepsilon'), \tag{8}$$

and the pair $(\tilde{s}_W, \tilde{s}_B)$ returned in step 9 is absolutely $K + 2\delta(\mathcal{G}, \varepsilon') \leq 2K$-optimal for $\mathcal{G}$. Suppose that $\mathbb{A}$ determines that $\mu_w(\hat{\mathcal{G}}) \geq 3/\varepsilon$ in step 8, and hence the algorithm returns $(\tilde{s}_W, \tilde{s}_B)$. Then (8) implies that

$$K \leq \frac{\mu_w(\mathcal{G})}{\mu_w(\hat{\mathcal{G}}) - 1} \leq \frac{\mu_w(\mathcal{G})}{3/\varepsilon - 1} \leq \frac{\varepsilon}{2}\mu(\mathcal{G}),$$

and we are done. On the other hand, if $\mu_w(\hat{\mathcal{G}}) < 3/\varepsilon$ then, by (8), $\mu_w(\mathcal{G}) < \frac{K(3+2\varepsilon)}{\varepsilon} = \frac{p_{\min}^{2k+1} r^+}{2(1+\varepsilon)n}$. By Lemma 2.3, applied with $t = K(3+2\varepsilon)/\varepsilon$, the game $\tilde{\mathcal{G}}$ defined in step 13 satisfies $\mu_v(\mathcal{G}) = \mu_v(\tilde{\mathcal{G}})$, and any (relatively) ε-optimal strategy in $(\tilde{\mathcal{G}}, w)$ (in particular the one returned in step 14) is also ε-optimal for $(\mathcal{G}, w)$. □

Remark 2.7. *For structurally ergodic BWR-games, one can easily modify the above procedure to return* uniformly *ε-optimal strategies.*

2.4. Uniformly Relative Approximation for BW-Games

The FPTAS in Theorem 2.6 does not necessarily return a uniformly ε-optimal situation, even if the given pseudo-polynomial algorithm $\mathbb{A}$ provides a uniformly optimal situation. For BW-games, we can modify this FPTAS to return a uniformly ε-optimal situation. The algorithm is given as Algorithm 2. The main difference is that when we recurse on a game with reduced rewards (step 14), we also have 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 [15]. However, the main technical difficulty is that, with approximate equilibria, White or Black might still have some incentive to move to a lower- or higher-value class, respectively, since the values are just approximations. We show that such a move will be profitable neither for White nor for Black. Recall that the rewards are non-negative integers.

Lemma 2.8. *Let $\mathbb{A}$ be a pseudo-polynomial algorithm that solves, in uniformly optimal strategies, any BW-game $\mathcal{G}$ in time $\tau(n, R)$. Then for any $\varepsilon > 0$, there is an algorithm that solves, in uniformly relatively ε-optimal strategies, any BW-game $\mathcal{G}$, in time $O\big(\big(\tau\big(n, \frac{2(1+\varepsilon')^2 n}{\varepsilon'}\big) + \mathrm{poly}(n)\big) \cdot h\big)$, where $h = \lfloor \log R \rfloor + 1$, and $\varepsilon' = \frac{\ln(1+\varepsilon)}{4h-2}$.*

Proof. The algorithm FPTAS-BW$(\mathcal{G}, \varepsilon)$ is given as Algorithm 2. The bound on the running-time is obvious: In step (12), each time we recurse on a game $\tilde{\mathcal{G}}$ with $r^+(\tilde{\mathcal{G}})$ reduced by a factor of at least half. Moreover, the rewards in the truncated game $\hat{\mathcal{G}}$ are integral with a maximum value of $r^+(\hat{\mathcal{G}}) \leq \frac{r^+}{K} \leq \frac{2(1+\varepsilon')^2 n}{\varepsilon'}$. Thus, the time that algorithm $\mathbb{A}$ needs in each recursive call is bounded from above by $\tau\big(n, \frac{2(1+\varepsilon')n}{\varepsilon'}\big)$.

So it remains to argue (by induction) that the algorithm returns a pair $(\tilde{s}_W, \tilde{s}_B)$ of ε-optimal strategies. Let us index the different recursive calls of the algorithm by $i = 1, 2, \ldots, h' \leq h$ and denote by $\mathcal{G}^{(i)} = (G^{(i)} = (V, E^{(i)}), r^{(i)})$ the game input to the ith recursive call of the algorithm (so $\mathcal{G}^{(1)} = \mathcal{G}$) and by $\hat{s}^{(i)} = (\hat{s}_W^{(i)}, \hat{s}_B^{(i)})$, $\tilde{s}^{(i)} = (\tilde{s}_W^{(i)}, \tilde{s}_B^{(i)})$ the pair of strategies returned either in steps 1, 5, 8, or 16. Similarly, we denote by $V^{(i)} = B_W^{(i)} \cup V_B^{(i)}$, $U^{(i)}$, $r^{(i)}$, $K^{(i)}$ $\hat{r}^{(i)}$, $\hat{\mathcal{G}}^{(i)}$, $\tilde{\mathcal{G}}^{(i)}$ the instantiations of V, U, r, $\hat{r}$, K, $\hat{\mathcal{G}}$, $\tilde{\mathcal{G}}$, respectively, in the ith call of the algorithm. We denote by $S_W^{(i)}$ and $S_B^{(i)}$ the set of strategies in $\mathcal{G}^{(i)}$ for White and Black, respectively. For a set U of vertices, a game $\mathcal{G}$, and a situation s, we denote by $\mathcal{G}[U] = (G[U], r)$ and $s[U]$, respectively, the game and situation induced on U.

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

1: **if** $r^+(\mathcal{G}) = 1$ **then return** $\mathbb{A}(\mathcal{G})$
2: $K \leftarrow \frac{\varepsilon' r^+}{2(1+\varepsilon')^2 n}$
3: $\hat{r}_e \leftarrow \lfloor r_e / K \rfloor$ for $e \in E$
4: $\hat{\mathcal{G}} \leftarrow (G, \hat{r})$
5: $(\hat{s}_W, \hat{s}_B) \leftarrow \mathbb{A}(\hat{\mathcal{G}})$
6: $U \leftarrow \{u \in V \mid \mu_u(\hat{\mathcal{G}}) \geq 1/\varepsilon'\}$
7: **if** $U = V$ **then**
8: **return** $(\tilde{s}_W, \tilde{s}_B) = (\hat{s}_W, \hat{s}_B)$
9: **else**
10: $\tilde{G} \leftarrow G[V \setminus U]$
11: **for all** $e \in E(\tilde{G})$ **do**
12: $\tilde{r}_e \leftarrow \begin{cases} \lceil \frac{r^+}{2} \rceil & \text{if } r_e > \frac{r^+}{2(1+\varepsilon')} \text{ and} \\ r_e & \text{otherwise} \end{cases}$
13: $\tilde{\mathcal{G}} \leftarrow (\tilde{G}, \tilde{r})$
14: $(\tilde{s}_W, \tilde{s}_B) \leftarrow$ FPTAS-BW$(\tilde{\mathcal{G}}, \varepsilon)$
15: $\tilde{s}(w) \leftarrow \hat{s}(w)$ for all $w \in U$
16: **return** $\tilde{s} = (\tilde{s}_W, \tilde{s}_B)$

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

Claim 2.9. *(i) There does not exist an edge $(v, u) \in E$ such that $v \in V_B^{(i)} \cap U^{(i)}$ and $u \in V^{(i)} \setminus U^{(i)}$.*

(ii) For all $v \in V_W^{(i)} \cap U^{(i)}$, there exists a $u \in U^{(i)}$ with $(v, u) \in E$.

(i′) There does not exist an edge $(v, u) \in E$ such that $v \in V_W^{(i)} \setminus U^{(i)}$ and $u \in U^{(i)}$.

(ii′) For all black nodes $v \in V_B^{(i)} \setminus U^{(i)}$, there exists a $u \in V^{(i)} \setminus U^{(i)}$ such that $(v, u) \in E$.

(iii) Let $\hat{s}^{(i)} = (\hat{s}_W^{(i)}, \hat{s}_B^{(i)})$ be the situation returned in step 5. Then, for all $v \in U^{(i)}$, we have $\hat{s}^{(i)}(v) \in U^{(i)}$, and, for all $v \in V^{(i)} \setminus U^{(i)}$, we have $\hat{s}^{(i)}(v) \in V^{(i)} \setminus U^{(i)}$.

Proof. By the optimality conditions in $\hat{\mathcal{G}}^{(i)}$ (see, e.g., Gurvich et al. [15]), we have

(I) $\mu_v(\hat{\mathcal{G}}^{(i)}) = \min\{\mu_u(\hat{\mathcal{G}}^{(i)}) \mid u \in V^{(i)}$ such that $(v, u) \in E\}$, for $v \in V_B^{(i)}$, and

(II) $\mu_v(\hat{\mathcal{G}}^{(i)}) = \max\{\mu_u(\hat{\mathcal{G}}^{(i)}) \mid u \in V^{(i)}$ such that $(v, u) \in E\}$, for any $v \in V_W^{(i)}$.

(I) and (II), together with the definition of $U^{(i)}$, imply (i) and (ii), respectively. Similarly (i') and (ii') can be shown. The optimality conditions also imply that $\mu_v(\hat{\mathcal{G}}^{(i)}) = \mu_{\hat{s}^{(i)}(v)}(\hat{\mathcal{G}}^{(i)})$, which in turn implies (iii). □

Note that Claim 2.9 implies that the game $\mathcal{G}^{(i)}[V^{(i)} \setminus U^{(i)}]$ is well-defined since the graph $G[V^{(i)} \setminus U^{(i)}]$ has no sinks. For a strategy s_W (and similarly for a strategy s_B) and a subset $V' \subseteq V$, we write $S_W(V') = \{s_W(u) \mid u \in V'\}$.

Claim 2.10. *Let $\hat{s}^{(i)}$ be the situation returned in step 5. Then, for all $i = 1, \ldots, h'$ and any $w \in U^{(i)}$, we have*

$$\max_{s_W:\ s_W(U^{(i)} \cap V_W) \subseteq U^{(i)}} \mu_w(\mathcal{G}^{(i)}(s_W, \hat{s}_B^{(i)})) \leq (1+\varepsilon')\mu_w(\mathcal{G}^{(i)}) \quad \text{and}$$
$$\min_{s_B:\ s_B(U^{(i)} \cap V_B) \subseteq U^{(i)}} \mu_w(\mathcal{G}^{(i)}(\hat{s}_W^{(i)}, s_B)) \geq (1-\varepsilon')\mu_w(\mathcal{G}^{(i)}).$$

Proof. This follows from Lemma 2.1 by the uniform optimality of $\hat{s}^{(i)}$ in $\hat{\mathcal{G}}^{(i)}$ and the fact that $\mu_w(\hat{\mathcal{G}}^{(i)}) \geq 1/\varepsilon'$ for every $w \in U^{(i)}$. □

Claim 2.11. *For all $u \in U^{(i)}$ and $v \in V^{(i)} \setminus U^{(i)}$, we have $(1+\varepsilon')\mu_u(\mathcal{G}^{(i)}) > \mu_v(\mathcal{G}^{(i)})$.*

Proof. For $u \in U^{(i)}$, $v^{(i)} \in V^{(i)} \setminus U^{(i)}$, we have $\mu_u(\hat{\mathcal{G}}^{(i)}) \geq 1/\varepsilon'$ and $\mu_v(\hat{\mathcal{G}}^{(i)}) < 1/\varepsilon'$. Thus,

$$\mu_v(\mathcal{G}^{(i)}) \leq K^{(i)}\mu_v(\hat{\mathcal{G}}^{(i)}) + K^{(i)} < \frac{K^{(i)}}{\varepsilon'}(1+\varepsilon') \leq K^{(i)}\mu_u(\hat{\mathcal{G}}^{(i)})(1+\varepsilon') \leq \mu_u(\mathcal{G}^{(i)})(1+\varepsilon').$$

□

We observe that the strategy $\tilde{s}^{(i)}$, returned by the ith call to the algorithm, is determined as follows (c.f. steps 14 and 15): for $w \in U^{(i)}$, $\tilde{s}^{(i)}(w) = \hat{s}^{(i)}(w)$ is chosen by the solution of the game $\hat{\mathcal{G}}^{(i)}$, and for $w \notin U^{(i)}$, $\tilde{s}^{(i)}(w)$ is determined by the (recursive) solution on the residual game $\tilde{\mathcal{G}}^{(i)} = \mathcal{G}^{(i+1)}$. The following claim states that the value of any vertex $u \in V^{(i)} \setminus U^{(i)}$ in the residual game is a good approximation of the value in the original game $\mathcal{G}^{(i)}$.

Claim 2.12. *For all $i = 1, \ldots, h'$ and any $u \in V^{(i)} \setminus U^{(i)}$, we have*

$$\mu_u(\mathcal{G}^{(i)}) \leq \mu_u(\mathcal{G}^{(i)}[V^{(i)} \setminus U^{(i)}]) \leq (1+2\varepsilon')\mu_u(\mathcal{G}^{(i)}). \tag{9}$$

Proof. Fix $u \in V^{(i)} \setminus U^{(i)}$. Let $s^* = (s_W^*, s_B^*)$ and $(\bar{s}_W, \bar{s}_B)$ be optimal situations in $(\mathcal{G}^{(i)}, u)$ and $(\bar{\mathcal{G}}^{(i)}, u) := (\mathcal{G}^{(i)}[V^{(i)} \setminus U^{(i)}], u)$, respectively. Let us extend $\bar{s}$ to a situation in $\mathcal{G}^{(i)}$ by setting $\bar{s}(v) = \hat{s}^{(i)}(v)$ for all $v \in U^{(i)}$, where $\hat{s}$ is the situation returned in step 5. Then, by Claim 2.9(i'), White has no way to escape to $U^{(i)}$, or in other words, $s_W^*(u') \in V^{(i)} \setminus U^{(i)}$ for all $u' \in V_W^{(i)} \setminus U^{(i)}$. Hence,

$$\begin{aligned} \mu_u(\mathcal{G}^{(i)}) &= \mu_u(\mathcal{G}^{(i)}(s_W^*, s_B^*)) \leq \mu_u(\mathcal{G}^{(i)}(s_W^*, \bar{s}_B)) \\ &= \mu_u(\bar{\mathcal{G}}^{(i)}(s_W^*, \bar{s}_B)) \leq \mu_u(\bar{\mathcal{G}}^{(i)}(\bar{s}_W, \bar{s}_B)) = \mu_u(\bar{\mathcal{G}}^{(i)}). \end{aligned}$$

For similar reasons, $\mu_u(\mathcal{G}^{(i)}) \geq \mu_u(\bar{\mathcal{G}}^{(i)})$, if $s_B^*(v) \in V^{(i)} \setminus U^{(i)}$ for all $v \in V_B^{(i)} \setminus U^{(i)}$ such that v is reachable from u in the graph $G(s_W^*, s_B^*)$. Suppose, on the other hand, that there is a $v \in V_B^{(i)} \setminus U^{(i)}$ such that $u' = s_B^*(v) \in U^{(i)}$, and v is reachable from u in the graph $G(s_W^*, s_B^*)$. Then $\mu_u(\mathcal{G}^{(i)}) = \mu_{u'}(\mathcal{G}^{(i)}) \geq K^{(i)}\mu_{u'}(\hat{\mathcal{G}}^{(i)}) \geq \frac{K^{(i)}}{\varepsilon'}$. Moreover, the optimality of $(\hat{s}_W, \hat{s}_B)$ in $\hat{\mathcal{G}}^{(i)}$ and the fact that $\frac{1}{K^{(i)}}r^{(i)} - 1 \leq \hat{r}^{(i)} \leq \frac{1}{K^{(i)}}r^{(i)}$ imply that

$$\begin{aligned} \forall s_W \in S_W^{(i)}:\ \mu_u(\mathcal{G}^{(i)}(\hat{s}_W, \hat{s}_B)) &\geq K^{(i)}\mu_u(\hat{\mathcal{G}}^{(i)}(\hat{s}_W, \hat{s}_B)) \geq K^{(i)}\mu_u(\hat{\mathcal{G}}^{(i)}(s_W, \hat{s}_B)) \\ &\geq \mu_u(\mathcal{G}^{(i)}(s_W, \hat{s}_B)) - K^{(i)} \\ &\geq \mu_u(\mathcal{G}^{(i)}(s_W, \hat{s}_B)) - \varepsilon'\mu_u(\mathcal{G}^{(i)}) \end{aligned}$$

and

$$\forall s_B \in S_B^{(i)}: \ \mu_u(\mathcal{G}^{(i)}(\hat{s}_W, \hat{s}_B)) \leq K^{(i)}\mu_u(\hat{\mathcal{G}}^{(i)}(\hat{s}_W, \hat{s}_B)) + K^{(i)} \leq K^{(i)}\mu_u(\hat{\mathcal{G}}^{(i)}(\hat{s}_W, s_B)) + K^{(i)}$$
$$\leq \mu_u(\mathcal{G}^{(i)}(\hat{s}_W, s_B)) + K^{(i)} \leq \mu_u(\mathcal{G}^{(i)}(\hat{s}_W, s_B)) + \varepsilon'\mu_u(\mathcal{G}^{(i)}).$$

In particular,

$$\begin{aligned}\mu_u(\mathcal{G}^{(i)}) &= \mu_u(\mathcal{G}^{(i)}(s_W^*, s_B^*)) \geq \mu_u(\mathcal{G}^{(i)}(\hat{s}_W, s_B^*)) \geq \mu_u(\mathcal{G}^{(i)}(\hat{s}_W, \hat{s}_B)) - \varepsilon'\mu_u(\mathcal{G}^{(i)})\\ &\geq \mu_u(\mathcal{G}^{(i)}(\bar{s}_W, \hat{s}_B)) - 2\varepsilon'\mu_u(\mathcal{G}^{(i)}) = \mu_u(\bar{\mathcal{G}}^{(i)}(\bar{s}_W, \hat{s}_B)) - 2\varepsilon'\mu_u(\mathcal{G}^{(i)})\\ &\geq \mu_u(\bar{\mathcal{G}}^{(i)}(\bar{s}_W, \bar{s}_B)) - 2\varepsilon'\mu_u(\mathcal{G}^{(i)}) = \mu_u(\bar{\mathcal{G}}^{(i)}) - 2\varepsilon'\mu_u(\mathcal{G}^{(i)}),\end{aligned}$$

where $\mu_u(\mathcal{G}^{(i)}(\bar{s}_W, \hat{s}_B)) = \mu_u(\bar{\mathcal{G}}^{(i)}(\bar{s}_W, \hat{s}_B))$ follows from Claim 2.9 (since $(\bar{s}_W, \hat{s}_B)(v) \in V^{(i)} \setminus U^{(i)}$). It follows that $\mu_u(\mathcal{G}^{(i)}) \geq \frac{1}{1+2\varepsilon'}\mu_u(\bar{\mathcal{G}}^{(i)})$. □

Let us fix $\varepsilon_{h'} = \varepsilon'$, and for $i = 1, 2, \ldots, h' - 1$, let us choose ε_i such that $1 + \varepsilon_i \geq (1 + \varepsilon')(1 + 2\varepsilon')(1 + \varepsilon_{i+1})$. Next, we claim that the strategies $(\tilde{s}_W^{(i)}, \tilde{s}_B^{(i)})$ returned by the ith call are relatively ε_i-optimal in $\mathcal{G}^{(i)}$.

Claim 2.13. *For all $i = 1, \ldots, h'$ and any $w \in V^{(i)}$, we have*

$$\max_{s_W \in S_W^{(i)}} \mu_w(\mathcal{G}^{(i)}(s_W, \tilde{s}_B^{(i)})) \leq (1 + \varepsilon_i)\mu_w(\mathcal{G}^{(i)}) \text{ and} \tag{10}$$

$$\min_{s_B \in S_B^{(i)}} \mu_w(\mathcal{G}^{(i)}(\tilde{s}_W^{(i)}, s_B)) \geq (1 - \varepsilon_i)\mu_w(\mathcal{G}^{(i)}). \tag{11}$$

Proof. The proof is by induction on $i = h', h' - 1, \ldots, 1$. For $i = h'$, the statement follows directly from Lemma 2.1 since $\mu_w(\hat{\mathcal{G}}^{(i)}) \geq 1/\varepsilon'$ for all $w \in V^{(i)}$. So suppose that $i < h'$. We distinguish two cases.

Case I: Consider an arbitrary strategy $s_W \in S_W^{(i)}$ for WHITE. Suppose first that $w \in U^{(i)}$. Note that, by Claim 2.9(iii), $\tilde{s}_B^{(i)}(v) \in U^{(i)}$ for all $v \in V_B \cap U^{(i)}$. If also $s_W(v) \in U^{(i)}$ for all $v \in V_W \cap U^{(i)}$, such that v is reachable from w in the graph $G(s_W, \tilde{s}_B^{(i)})$, then Claim 2.10 implies $\mu_w(\mathcal{G}^{(i)}(s_W, \tilde{s}_B^{(i)})) \leq (1 + \varepsilon')\mu_w(\mathcal{G}^{(i)}) \leq (1 + \varepsilon_i)\mu_w(\mathcal{G}^{(i)})$.

Suppose that $u = s_W(v) \notin U^{(i)}$ for some $v \in V_W \cap U^{(i)}$ such that v is reachable from w in the graph $G(s_W, \tilde{s}_B^{(i)})$.

By induction, $\bar{s}^{(i)} = (\bar{s}_W^{(i)}, \bar{s}_B^{(i)}) := (\tilde{s}_W^{(i)}, \tilde{s}_B^{(i)})[V^{(i)} \setminus U^{(i)}]$ is ε_{i+1}-optimal in $\mathcal{G}^{(i+1)} = \tilde{\mathcal{G}}^{(i)}$. Recall that the game $\tilde{\mathcal{G}}^{(i)}$ is obtained from $\bar{\mathcal{G}}^{(i)} := \mathcal{G}^{(i)}[V^{(i)} \setminus U^{(i)}]$ by reducing the rewards according to step (12). Thus, Lemma 2.3 yields that $\mu_u(\bar{\mathcal{G}}^{(i)}) = \mu_u(\tilde{\mathcal{G}}^{(i)})$, and

$$\max_{s'_W \in S_W^{(i+1)}} \mu_u(\mathcal{G}^{(i)}(s'_W, \bar{s}_B^{(i)})) \leq (1 + \varepsilon_{i+1})\mu_u(\bar{\mathcal{G}}^{(i)}) \tag{12}$$

$$\min_{s'_B \in S_B^{(i+1)}} \mu_u(\mathcal{G}^{(i)}(\bar{s}_W^{(i)}, s'_B)) \geq (1 - \varepsilon_{i+1})\mu_u(\bar{\mathcal{G}}^{(i)}). \tag{13}$$

Note that $\tilde{s}_B^{(i)}(u') \in V^{(i)} \setminus U^{(i)}$ for all $u' \in V_B^{(i)} \setminus U^{(i)}$, and by Claim 2.9(i'), $S_W^{(i+1)}$ is the restriction of $S_W^{(i)}$ to $V^{(i)} \setminus U^{(i)}$. Thus, we get the following series of inequalities:

$$\begin{aligned}\mu_w(\mathcal{G}^{(i)}(s_W, \tilde{s}_B^{(i)})) &= \mu_u(\mathcal{G}^{(i)}(s_W, \tilde{s}_B^{(i)})) \leq (1 + \varepsilon_{i+1})\mu_u(\mathcal{G}^{(i)}[V^{(i)} \setminus U^{(i)}])\\ &\leq (1 + \varepsilon_{i+1})(1 + 2\varepsilon')\mu_u(\mathcal{G}^{(i)})\\ &< (1 + \varepsilon_{i+1})(1 + 2\varepsilon')(1 + \varepsilon')\mu_w(\mathcal{G}^{(i)}) \leq (1 + \varepsilon_i)\mu_w(\mathcal{G}^{(i)}).\end{aligned}$$

The first inequality holds by (12). The second inequality holds because of (9). The third one follows from Claim 2.11. The fourth inequality holds since $(1+\varepsilon_{i+1})(1+2\varepsilon')(1+\varepsilon') \leq (1+\varepsilon_i)$.

If $w \in V^{(i)} \setminus U^{(i)}$, then the above argument also shows that $\mu_w(\mathcal{G}^{(i)}(s_W, \tilde{s}_B^{(i)})) \leq (1+\varepsilon_{i+1})(1+2\varepsilon')\mu_w(\mathcal{G}^{(i)}) \leq (1+\varepsilon_i)\mu_w(\mathcal{G}^{(i)})$. Thus, (10) follows.

Case II: Consider an arbitrary strategy $s_B \in S_B^{(i)}$ for Black. If $w \in U^{(i)}$, then we have $\mu_w(\mathcal{G}^{(i)}(\tilde{s}_W^{(i)}, s_B)) \geq (1-\varepsilon')\mu_w(\mathcal{G}^{(i)}) \geq (1-\varepsilon_i)\mu_w(\mathcal{G}^{(i)}w)$ follows from Claims 2.9(i), 2.9(iii), and 2.10.

Suppose now that $w \in V^{(i)} \setminus U^{(i)}$. If $s_B \in S_B^{(i+1)}$, then we get by (13) and (9) that $\mu_w(\mathcal{G}^{(i)}(\tilde{s}_W^{(i)}, s_B)) \geq (1-\varepsilon_{i+1})\mu_w(\mathcal{G}^{(i)}) \geq (1-\varepsilon_i)\mu_w(\mathcal{G}^{(i)})$. A similar situation holds if $s_B(v) \in V^{(i)} \setminus U^{(i)}$ for all $v \in V_B^{(i)} \setminus U^{(i)}$ such that v is reachable from w in the graph $G(\tilde{s}_W^{(i)}, s_B)$. So it remains to consider the case when there is a $v \in V_B^{(i)} \setminus U^{(i)}$ such that $u = s_B(v) \in U^{(i)}$, and v is reachable from w in the graph $G(\tilde{s}_W^{(i)}, s_B)$. Since Black has no escape from $U^{(i)}$ in this case (by Claim 2.9(i)), Claims 2.10 and 2.11 yield

$$\begin{aligned}\mu_w(\mathcal{G}^{(i)}(\tilde{s}_W^{(i)}, s_B)) &= \mu_u(\mathcal{G}^{(i)}(\tilde{s}_W^{(i)}, s_B)) \geq (1-\varepsilon')\mu_u(\mathcal{G}^{(i)}) \\ &> (1-\varepsilon')^2\mu_w(\mathcal{G}^{(i)}) \geq (1-\varepsilon_i)\mu_w(\mathcal{G}^{(i)}).\end{aligned}$$

□

To finish the proof of Lemma 2.8, we set the ε's such that $\varepsilon_1 = \big((1+2\varepsilon')(1+\varepsilon')\big)^{h'-1}(1+\varepsilon') - 1 \leq \varepsilon$. □

3. Smoothed Analysis

In this section, we show that games that can be solved in pseudo-polynomial time have polynomial smoothed complexity. To do this, let $\mathbb{G}$ be a class of graphs $G = (V, E)$ that admits a pseudo-polynomial algorithm $\mathbb{A}$. Let $\mathcal{G} = (G, P, r)$ be a BWR-game on G. We let an adversary specify G and P together with density functions for the rewards (one density function for each arc). These density functions are bounded by $\phi \geq 1$. Then the actual reward of each arc is drawn independently according to its density function. We design an algorithm and show that its running-time can be bounded as in (5).

One (technical) issue is that the perturbed rewards are of course real, non-rational numbers with probability 1. Thus, we cannot represent these numbers and compute anything with them on an ordinary RAM. To cope with this problem, we follow Beier and Vöcking's approach [3]: We assume that the rewards are in the interval $[-1, 1]$ and that we can access the bits of the rewards one-by-one by querying an oracle.

To state our results in a bit more general setting, we will assume that $\mathbb{A}$ solves any BWR-game on G in uniformly optimal strategies. If this is 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.

Before describing the procedure (Algorithm 3), we need to introduce some notation. Let us write $\lfloor x \rfloor_b$ for x being rounded down to the next multiple of 2^{-b}. This means that we cut off all bits after the b-th bit of the binary representation of x. Let

$$\gamma = \gamma(\mathcal{G}) = (kn)^{-2}(2D)^{-2(k+2)} \tag{14}$$

and $\varepsilon > 0$. Given a game $\mathcal{G} = (G = (V, E), P, r)$, we define, for each $e \in E$, the game $\mathcal{G}_{e,\varepsilon} = (G, P, r(e))$, where

$$r_{e'}(e) = \begin{cases} r_e + 2\gamma^{-1}\varepsilon & \text{if } e' = e \text{ and} \\ r_{e'} & \text{otherwise.} \end{cases} \tag{15}$$

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. Since the rewards are rounded down, we can scale them to obtain integral rewards (Step 5). This can be done in polynomial-time as long as we have at most roughly $O(\log n)$ bits. Then we try to certify that the solution obtained is also a solution for the true (but unknown) rewards (Step 7). If this succeeds, then we are done. If this fails, then we request one more bit for every reward and repeat the process.

To prove a smoothed polynomial running-time, we have 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 for proving that the computed equilibrium is indeed an equilibrium for the original game (we will show that such a certificate is given in Step 7). 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.

3.1. Isolation Lemma

A key ingredient for our smoothed analysis is an adaption of the isolation lemma [25] to our setting. A variant of the isolation lemma has already been applied successfully in smoothed analysis of integer programs [3, 27]. It basically says the following: Of course there are exponentially many alternative strategies for each player. But if we consider the optimal pair of strategies, then choosing an alternative strategy makes the payoff for the respective player significantly worse (with high probability). As usual, we assume that the maximum over an empty set is $-\infty$ and the minimum is $+\infty$.

Our variant of the isolation is similar to Röglin and Vöcking's variant for integer programming [28]. It generalizes their isolation lemma in the sense that it only requires a significant difference of a single component of any pair of vectors.

Lemma 3.1 (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*

$$\Pr(\text{gap}(w) \leq \varepsilon) \leq |E|\varepsilon\phi\kappa^2/\gamma,$$

where $\kappa = \max_{e \in E} |\mathcal{F}_e|$, and $\mathcal{F}_e = \{x \mid \rho_e = x \text{ for some } \rho \in \mathcal{F}\}$.

Proof. For $e \in E$ and $x, y \in \mathbb{R}_+$, we define

$$\Delta_{e,x,y} = \max_{\rho \in \mathcal{F},\ \rho_e = x} (w^T\rho - w_e x) - \max_{\rho \in \mathcal{F},\ \rho_e = y} (w^T\rho - w_e y).$$

It is crucial to observe that $\Delta_{e,x,y}$ is *independent* of w_e. With a probability of 1, there exist unique ρ^* and ρ^{**}, as defined in the lemma above. Since $\rho^* \neq \rho^{**}$, there exists an $e \in E$

with $|\rho_e^* - \rho_e^{**}| \geq \gamma$ by assumption. Suppose that $\rho_e^* = x$ and $\rho_e^{**} = y$. Then $w^T\rho^* - w_e x = \max_{\rho\in\mathcal{F},\rho_e=x}(w^T\rho - w_e x)$ and $w^T\rho^{**} - w_e y = \max_{\rho\in\mathcal{F},\rho_e=y}(w^T\rho - w_e y)$ imply that $w^T\rho^* - w^T\rho^{**} = \Delta_{e,x,y} + w_e(x-y)$. Thus,

$$
\begin{aligned}
\Pr(\mathrm{gap}(w) < \varepsilon) &\leq \Pr(\exists e \in E, x, y : 0 \leq w^T\rho^* - w^T\rho^{**} \leq \varepsilon, \rho_e^* = x, \rho_e^{**} = y, |x-y| \geq \gamma) \\
&= \sum_{e\in E} \sum_{x,y\in\mathcal{F}_e:|x-y|\geq\gamma} \Pr(0 \leq w^T\rho^* - w^T\rho^{**} \leq \varepsilon, \rho_e^{**} = x, \rho_e^* = y) \\
&\leq \sum_{e\in E} \sum_{x,y\in\mathcal{F}_e:|x-y|\geq\gamma} \Pr(0 \leq \Delta_{e,x,y} + w_e(x-y) \leq \varepsilon) \\
&= \sum_{e\in E} \sum_{x,y\in\mathcal{F}_e:x-y\geq\gamma} \Pr\left(\frac{-\Delta_{e,x,y}}{x-y} \leq w_e \leq \frac{\varepsilon - \Delta_{e,x,y}}{x-y}\right) \\
&\quad + \sum_{e\in E} \sum_{x,y\in\mathcal{F}_e:y-x\geq\gamma} \Pr\left(\frac{-\varepsilon + \Delta_{e,x,y}}{y-x} \leq w_e \leq \frac{\Delta_{e,x,y}}{y-x}\right) \\
&\leq \sum_{e\in E} \sum_{x,y\in\mathcal{F}_e:|x-y|\geq\gamma} \frac{\phi\varepsilon}{|x-y|} \leq |E|\kappa^2 \frac{\phi\varepsilon}{\gamma}.
\end{aligned}
$$

□

We will use Lemma 3.1 above 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. In order to be able to do this, we need bounds for κ and γ, which are given by the following lemma.

Lemma 3.2. *Let $\mathcal{G} = (G = (V, E), P, r)$ be a BWR-game, $u \in V$ be any vertex, and s be an arbitrary any 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 a/b with $a, b \in \mathbb{N}$ 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 $\rho_e(s) - \rho_e(s') \geq \gamma = \gamma(\mathcal{G})$ (see (14)).

Proof. The first claim has been proved by Boros et al. [7], and the second claim is an immediate consequence. It is also immediate that if $\rho(s) \neq \rho(s')$, then there is an arc e such that $|\rho_e(s) - \rho_e(s')| \geq \gamma$. To see the stronger claim in (*iii*), we assume without loss of generality that there is no arc e such that $\rho_e(s) > 0$ and $\rho_e(s') = 0$. Then for all e, $\rho_e(s) > 0$ if and only if $\rho_e(s') > 0$. Hence, $P(s)$ and $P(s')$ have the same absorbing classes. (Indeed, if the absorbing classes of $P(s)$ and $P(s')$ are $C_1, \ldots, C_\ell$ and $C'_1, \ldots, C'_{\ell'}$, respectively, then for all $i \in [\ell]$, there is a $j \in [\ell']$ such that $C_i \subseteq C'_j$. Suppose that for some i and j, $C_i \subset C'_j$. Then there should be vertices $v \in V_W \cup V_B$ and $u \in C'_j \setminus C_i$, such that $\rho_{(v,u)}(s) = 0$ while $\rho_{(v,u)}(s') > 0$. But then there should also exist a $u' \in C_i$ (corresponding to the strategy $s(v)$) such that $\rho_{(v,u')}(s) > 0$ and $\rho_{(v,u')}(s') = 0$, which contradicts our assumption. Hence, $\ell = \ell'$ and $C_i = C'_i$ for all i.) Let $\pi_{C_i}(s)$ and $\pi_{C_i}(s')$ be the absorption probabilities into class C_i in $P(s)$ and $P(s')$, respectively. Since $\rho(s) \neq \rho(s')$ and $\sum_i \pi_{C_i}(s) = \sum_i \pi_{C_i}(s') = 1$, there must exist an $i \in [\ell]$ such $\pi_{C_i}(s) > \pi_{C_i}(s')$. Hence, any arc $e = (v, u)$ with $u, v \in C_i$ satisfies $\rho_e(s) > \rho_e(s')$, which implies the claim. □

3.2. Truncation

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 the following: 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 a probability of 1, 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 must 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 $s_B^* \in S_B$ that satisfies $\mu_u(\mathcal{G}(s_W, s_B^*)) \leq \mu_u(\mathcal{G}(s_W, s_B))$ 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 3.3. *Let $\mathcal{G} = (G = (V, E), P, r)$ be a BWR-game such that $r = (r_e)_{e \in E}$ is a vector of independent continuous random variables with maximum density ϕ. Let $\varepsilon > 0$, and let let $\theta = \frac{2n^3 \varepsilon \phi}{\gamma(\mathcal{G})^3}$. Then the following holds for any situation s:*

(i) $\Pr(\exists r' : \|r' - r\|_\infty \leq \varepsilon$, *$s$ is not uniformly optimal in $\mathcal{G}'$ | s is uniformly optimal in $\mathcal{G}$*$) \leq 2\theta$*, where $\mathcal{G}'$ is the game obtained from $\mathcal{G}$ by replacing r by r'.*

(ii) $\Pr($*s is not uniformly optimal in $\mathcal{G}$ |* $\exists r' : \|r' - r\|_\infty \leq \varepsilon$, *$s$ is uniformly optimal in $\mathcal{G}'$*$) \leq 2\theta$*, where $\mathcal{G}'$ is defined as in* (i).

Proof. We prove the following claim for any $s_B \in S_B$. Together with the analogous claim for any $s_W \in S_W$, part (i) of the lemma follows.

Claim 3.4. *Let $s_B \in S_B$ be an arbitrary strategy of* BLACK*, and $s_W^* \in \mathrm{UBR}_{\mathcal{G}}(s_B)$. Then*

$$\Pr\big(\exists u \in V, s_W \in S_W, r' : \; \mu_u(\mathcal{G}'(s_W, s_B)) \geq \mu_u(\mathcal{G}'(s_W^*, s_B)) \text{ and } \rho((s_W, s_B), u) \neq \rho((s_W^*, s_B), u) \text{ and } \|r - r'\|_\infty \leq \varepsilon\big) \leq \theta.$$

Proof. For a starting vertex u, let $A(u)$ be the event that there exists a strategy $s_W \in S_W$ of WHITE and rewards r' with $\|r - r'\|_\infty \leq \varepsilon$ such that $\mu_u(\mathcal{G}'(s_W, s_B)) \geq \mu_u(\mathcal{G}'(s_W^*, s_B))$ and $\rho((s_W, s_B), u) \neq \rho((s_W^*, s_B), u))$. This means that the strategy s_W is better for WHITE than s_W^* and it also leads to a different stationary distribution.

The probability for which we need an upper bound is $\Pr(\bigcup_{u \in V} A(u))$. We estimate $\Pr(A(u))$ for a fixed vertex u and then apply a union bound over all vertices.

Suppose there exist a strategy s_W and a reward vector r' that cause event $A(u)$ to occur. By the optimality of s_W^* for $\mathcal{G}$, we have $\mu_u(\mathcal{G}(s_W^*, s_B)) \geq \mu_u(\mathcal{G}(s_W, s_B))$. We have equality if and only if $\rho((s_W^*, s_B), u) = \rho((s_W, s_B), u))$. Since $\|r' - r\|_\infty \leq \varepsilon$, Lemma 2.1 and the optimality of s_W for $\mathcal{G}'$ yield

$$\mu_u(\mathcal{G}(s_W, s_B)) \geq \mu_u(\mathcal{G}'(s_W, s_B)) - \varepsilon \geq \mu_u(\mathcal{G}'(s_W^*, s_B)) - \varepsilon.$$

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: $\ell_0 \leftarrow \log((nD)^{c_0}\phi)$ $\quad\triangleright$ c_0 is a constant to be specified later on
2: $i \leftarrow 0$
3: **repeat**
4: $\quad \ell \leftarrow \ell_0 + i$; $\varepsilon \leftarrow 2^{-\ell}$; $i \leftarrow i + 1$
5: $\quad \tilde{r} \leftarrow \lfloor r \rfloor_\ell$; $\tilde{\mathcal{G}} \leftarrow (G, P, \tilde{r})$; $\tilde{\mathcal{G}}' \leftarrow (G, P, 2^\ell \tilde{r})$
6: $\quad (\tilde{s}_W, \tilde{s}_B) \leftarrow \mathbb{A}(\tilde{\mathcal{G}}')$
7: **until** $\tilde{s}$ is optimal in $\tilde{\mathcal{G}}_{e,\varepsilon}$ for all $e \in E$

Algorithm 3: Solving stochastic mean payoff games with polynomial smoothed complexity.

Furthermore, Lemma 2.1 gives us $\mu_u(\mathcal{G}'(s_W^*, s_B)) \geq \mu_u(\mathcal{G}(s_W^*, s_B)) - \varepsilon$. Together, we obtain

$$\mu_u(\mathcal{G}(s_W^*, s_B)) \geq \mu_u(\mathcal{G}(s_W, s_B)) \geq \mu_u(\mathcal{G}(s_W^*, s_B)) - 2\varepsilon. \tag{16}$$

Now we show that the existence of a strategy s_W and a reward vector r' is unlikely. To do this, we use Lemma 3.1. Let $\mathcal{F} = \{\rho((s_W, s_B), u) \mid s_W \in S_W\}$. Then the elements of $\mathcal{F}$ satisfy the conditions of Lemma 3.1, with γ and κ as defined in Lemma 3.2, and

$$\text{gap}(r) = \mu_u(\mathcal{G}(s_W^*, s_B)) - \max_{s'_W : \rho((s'_W, s_B), u) \neq \rho((s_W^*, s_B), u)} \mu_u(\mathcal{G}(s'_W, s_B)).$$

Note that (16) implies $\text{gap}(r) \leq 2\varepsilon$. By Lemmas 3.1 and 3.2, the probability that this happens is bounded from above by $\frac{2n^2\varepsilon\phi}{\gamma(\mathcal{G})^3}$. The claim follows. □

Let us finish the proof of part (i) of the lemma. First, suppose that $s = (s_W, s_B)$ is uniformly optimal in $\mathcal{G}$. Then $s_W \in \text{UBR}_\mathcal{G}(s_B)$ and $s_B \in \text{UBR}_\mathcal{G}(s_W)$, and the claim implies immediately that $\Pr(s \text{ is not uniformly optimal in } \mathcal{G}' \text{ for some } r') \leq 2\theta$.

Now we turn to the proof of part (ii) of the lemma. Suppose that there exists rewards r' such that $s = (s_W, s_B)$ is uniformly optimal in $\mathcal{G}'$. Pick $s_W^* \in \text{UBR}_\mathcal{G}(s_B)$ and $s_B^* \in \text{UBR}_\mathcal{G}(s_W)$. Claim 3.4 yields the following: With a probability of at least $1 - \theta$, for all $u \in V$, either $\mu_u(\mathcal{G}'(s_W, s_B)) < \mu_u(\mathcal{G}'(s_W^*, s_B))$ or $\rho((s_W, s_B), u) = \rho((s_W^*, s_B), u)$. The former condition contradicts the optimality of s_W in $\mathcal{G}'$, so we must have the latter condition, which in turn implies that $\mu_u(\mathcal{G}(s_W, s_B)) = \mu_u(\mathcal{G}(s_W^*, s_B))$, i.e., $s_W \in \text{UBR}_\mathcal{G}(s_B)$. Similarly, we can show that $s_B \in \text{UBR}_\mathcal{G}(s_W)$ with a probability of at least $1 - \theta$, which completes the proof of part (ii). □

3.3. Certificates

Although unlikely, it can still 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 7 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 3.3 says that, with high probability, $\tilde{s}$ is a uniformly optimal situation in $\mathcal{G}$. Hence, it is very likely that 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 of the game, the reward on a single arc is changed 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 3.5. *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 (15)). Then $\tilde{s}$ is also optimal in $(\mathcal{G} = (G, P, r), u)$, for any r with $\|r - \tilde{r}\|_\infty \leq \varepsilon$.*

Proof. Fix an r such that $\|r - \tilde{r}\|_\infty \leq \varepsilon$. Let $s_W \in S_W$ be any strategy of White. If $\mu_u(\mathcal{G}(s_W, \tilde{s}_B)) \neq \mu_u(\mathcal{G}(\tilde{s}_W, \tilde{s}_B))$, then according to Lemma 3.2, there exists an arc e with $\rho_e(s_W, \tilde{s}_B) - \rho_e(\tilde{s}_W, \tilde{s}_B) \geq \gamma$. Note that for any situation s, $\mu_u(\tilde{\mathcal{G}}_{e,\varepsilon}(s)) = \mu_u(\tilde{\mathcal{G}}(s)) + \frac{2\varepsilon}{\gamma}\rho_e(s)$. Thus, we have

$$\begin{aligned}
\mu_u(\mathcal{G}(s_W, \tilde{s}_B)) &\leq \mu_u(\tilde{\mathcal{G}}(s_W, \tilde{s}_B)) + \varepsilon && \text{by Lemma 2.1}\\
&= \mu_u(\tilde{\mathcal{G}}_{e,\varepsilon}(s_W, \tilde{s}_B)) - \frac{2\varepsilon}{\gamma}\rho_e(s_W, \tilde{s}_B) + \varepsilon\\
&\leq \mu_u(\tilde{\mathcal{G}}_{e,\varepsilon}(\tilde{s}_W, \tilde{s}_B)) - \frac{2\varepsilon}{\gamma}\rho_e(s_W, \tilde{s}_B) + \varepsilon && \text{since } \tilde{s} \text{ is optimal in } \tilde{\mathcal{G}}_{e,\varepsilon}\\
&= \mu_u(\tilde{\mathcal{G}}(\tilde{s}_W, \tilde{s}_B)) - \frac{2\varepsilon}{\gamma}(\rho_e(s_W, \tilde{s}_B) - \rho_e(\tilde{s}_W, \tilde{s}_B)) + \varepsilon\\
&\leq \mu_u(\tilde{\mathcal{G}}(\tilde{s}_W, \tilde{s}_B)) - \varepsilon\\
&\leq \mu_u(\mathcal{G}(\tilde{s}_W, \tilde{s}_B)). && \text{by Lemma 2.1}
\end{aligned}$$

This shows that White cannot improve by switching to a different strategy. By a similar argument, $\mu_u(\mathcal{G}(\tilde{s}_W, s_B)) \geq \mu_u(\mathcal{G}(\tilde{s}_W, \tilde{s}_B))$ for all $s_B \in S_B$, which completes the proof. □

3.4. Completing the Smoothed Analysis

Now we have all ingredients to prove that BWR-games, on classes of graphs that admit a pseudo-polynomial algorithm and have a constant number of random vertices, can be solved in smoothed polynomial time.

Theorem 3.6. *Algorithm 3 solves (in uniformly optimal strategies) any BWR-game $\mathcal{G} = (G, P, r)$ in smoothed polynomial time, given that G admits a pseudo-polynomial algorithm $\mathbb{A}$ (that solves any such $\mathcal{G}$ in uniformly optimal strategies), the number k of random vertices is constant, and $D = \mathrm{poly}(n)$.*

Proof. For the correctness of Algorithm 3, it suffices to show the correctness of the certificate step (Step 7). Note that we have $\|\tilde{r} - r\|_\infty \leq \varepsilon = 2^{-\ell}$ in the iteration corresponding to $i = \ell$. By Lemma 3.5, if the situation $\tilde{s}$ found in Step 6 is uniformly optimal for the game $(\tilde{\mathcal{G}}_{e,\varepsilon}, u)$ for all $e \in E$, then it is also uniformly optimal for the game $\mathcal{G}$ (even though we do not know the rewards precisely).

What remains to be done is to reason about the running-time of the algorithm. To do this, we bound from above the probability that we fail to certify the optimality of $\tilde{s}$ for $\tilde{\mathcal{G}}_{e,\varepsilon}$ for some $e \in E$. For any $e \in E$, we have

$$\|\tilde{r}(e) - r\|_\infty \leq 2^{-\ell} + 2\gamma^{-1} \cdot 2^{-\ell} \leq 3 \cdot 2^{-\ell}\gamma^{-1}. \tag{17}$$

Conditioned on the event that $\tilde{s}$ is uniformly optimal in $\tilde{\mathcal{G}}$, let A be the event that $\tilde{s}$ is uniformly optimal in $\mathcal{G}$, and let, for $e \in E$, A_e be the event that $\tilde{s}$ is not uniformly optimal for $\tilde{\mathcal{G}}_{e,\varepsilon}$. By

Lemma 3.3(i), applied to $\mathcal{G}$ and $\mathcal{G}' = \tilde{\mathcal{G}}_{e,\varepsilon}$, where the difference in rewards satisfies (17), we have $\Pr(A_e \mid A) \le \frac{12n^3\phi 2^{-\ell}}{\gamma(\mathcal{G})^4}$. By Lemma 3.3(i), applied to $\mathcal{G}$ and $\mathcal{G}' = \tilde{\mathcal{G}}$, we have $\Pr(\overline{A}) \le \frac{4n^3\phi 2^{-\ell}}{\gamma(\mathcal{G})^3}$. Thus, the probability that $\tilde{s}$ is not uniformly optimal in $\tilde{\mathcal{G}}_{e,\varepsilon}$ for some $e \in E$ can be bounded from above by

$$\begin{aligned}\Pr(\exists e \in E:\ A_e) &\le \sum_{e\in E}\Pr(A_e) \le \sum_{e\in E}(\Pr(A_e \mid A) + \Pr(\overline{A}))\\ &\le |E|\left(\frac{12n^3\phi 2^{-\ell}}{\gamma(\mathcal{G})^4} + \frac{4n^3\phi 2^{-\ell}}{\gamma(\mathcal{G})^3}\right) \le \frac{16n^5\phi 2^{-\ell}}{\gamma(\mathcal{G})^4} \le (nD)^{c_0}2^{-\ell}\phi,\end{aligned}$$

for some constant $c_0 > 1$.

Note that Step 7 of the procedure can be implemented to run in time polynomial in n and $\log D$. This is because it boils down to solving a number of Markov decision processes with polynomially many bits, which can be solved by linear programming [24]. The running-time of each iteration of Algorithm 3 is dominated by the time required by the pseudo-polynomial algorithm. Let $c_4 n^{c_1} 2^{\ell c_2} D^{c_3}$ be an upper bound on the running-time of each iteration of the Algorithm 3. Here, n is the number of vertices, 2^ℓ is the maximum weight, and all probabilities are integer multiples of $1/D$. The numbers c_1, c_2, c_3, c_4 are appropriately chosen non-negative constants. By our assumption, there exists a non-negative constant c_5, such that $D \le n^{c_5}$. Define $\alpha = \min\{\frac{1}{c_1 + c_3c_5 + c_0c_2 + c_0c_2c_5}, \frac{1}{2c_2}\}$. Then

$$\begin{aligned}\mathbb{E}(\text{running-time}^\alpha) &\le \sum_{\ell \ge \ell_0}(nD)^{c_0}2^{-\ell}\phi\big(c_4 n^{c_1}2^{\ell c_2}D^{c_3}\big)^\alpha\\ &= c_4^\alpha n^{\alpha(c_1+c_3c_5+c_0c_2+c_0c_2c_5)}\phi^{\alpha c_2}\cdot\sum_{i=0}^{\infty}2^{-i(1-\alpha c_2)} = O(n\phi).\end{aligned}$$

This shows that Algorithm 3 runs in smoothed polynomial time. □

4. Concluding Remarks

In this paper, we have proved that computing the game values of classes of stochastic mean payoff games admits approximation schemes and has polynomial smoothed complexity, provided that the class of games at hand can be solved in pseudo-polynomial time.

While stochastic mean payoff games include parity games as a special case, the probabilistic model that we used here does not make sense for parity games. However, parity games can be solved in pseudo-polynomial time and in subexponential time [19]. We conjecture that they also have polynomial smoothed complexity with a reasonable probabilistic model.

Finally, let us remark that removing the assumption that k is constant in the above results remains a challenging open problem that seems to require totally new ideas. As an intermediate step, we ask two questions: First, can stochastic mean payoff games be solved in parameterized pseudo-polynomial time with the number k of stochastic vertices as the parameter? Second, is it possible to get rid of the dependence of the common denominator D of the probabilities?

References

[1] Daniel Andersson and Peter Bro Miltersen. The complexity of solving stochastic games on graphs. In *20th Int. Symp. on Algorithms and Computation (ISAAC)*, volume 5878 of *Lecture Notes in Computer Science*, pages 112–121. Springer, 2009.

[2] David Arthur, Bodo Manthey, and Heiko Röglin. k-means has smoothed polynomial complexity. In *Proc. of the 50th Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 405–414. IEEE Computer Society, 2009.

[3] René Beier and Berthold Vöcking. Typical properties of winners and losers in discrete optimization. *SIAM Journal on Computing*, 35(4):855–881, 2006.

[4] Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1):1–106, 2006.

[5] Endre Boros, Khaled Elbassioni, Mahmoud Fouz Vladimir Gurvich, Kazuhisa Makino, and Bodo Manthey. Stochastic mean payoff games: Smoothed analysis and approximation schemes. In *Proc. of the 38th Int. Colloquium on Automata, Languages and Programming (ICALP), Part I*, volume 6755 of *Lecture Notes in Computer Science*, pages 147–158. Springer, 2011.

[6] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. Every stochastic game with perfect information admits a canonical form. RRR-09-2009, RUTCOR, Rutgers University, 2009.

[7] Endre Boros, Khaled M. Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. A pumping algorithm for ergodic stochastic mean payoff games with perfect information. In *Proc. of the 14th Int. Conf. on Integer Programming and Combinatorial Optimization (IPCO)*, volume 6080 of *Lecture Notes in Computer Science*, pages 341–354. Springer, 2010.

[8] Endre Boros and Vladimir Gurvich. Why chess and backgammon can be solved in pure positional uniformly optimal strategies? RRR-21-2009, RUTCOR, Rutgers University, 2009.

[9] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *Journal of the ACM*, 56(3), 2009.

[10] Grace E. Cho and Carl D. Meyer. Markov chain sensitivity measured by mean first passage times. *Linear Algebra and its Applications*, 316(1–3):21–28, 2000.

[11] Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.

[12] Anne Condon. On algorithms for simple stochastic games. In Jin-Yi Cai, editor, *Advances in Computational Complexity Theory*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 51–73, 1993.

[13] Dean Gillette. Stochastic games with zero stop probabilities. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games, Volume 3*, volume 39 of *Annals of Mathematics Studies*, pages 179–187. Princeton University Press, 1957.

[14] Hugo Gimbert and Florian Horn. Simple stochastic games with few random vertices are easy to solve. In *Proc. 11th Int. Conf. Foundations of Software Science and Computational Structures (FoSSaCS)*, volume 4962 of *Lecture Notes in Computer Science*, pages 5–19. Springer, 2008.

[15] Vladimir Gurvich, Alexander V. 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.

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

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

[18] Marcin Jurdziński. *Games for Verification: Algorithmic Issues.* PhD thesis, University of Aarhus, BRICS, 2000.

[19] Marcin Jurdziński, Mike Paterson, and Uri Zwick. A deterministic subexponential algorithm for solving parity games. In *Proc. 17th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 117–123. SIAM, 2006.

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

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

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

[23] Michael L. Littman. *Algorithms for Sequential Decision Making.* PhD thesis, Brown University, Department of Computer Science, 1996.

[24] H. Mine and S. Osaki. *Markovian decision process.* Elsevier, 1970.

[25] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

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

[27] Heiko Röglin and Shang-Hua Teng. Smoothed analysis of multiobjective optimization. In *Proc. of the 50th Ann. IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 681–690. IEEE Computer Society, 2009.

[28] Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. *Mathematical Programming*, 110(1):21–56, 2007.

[29] Aaron Roth, Maria-Florina Balcan, Adam Kalai, and Yishay Mansour. On the equilibria of alternating move games. In *Proc. of the 21st Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 805–816. SIAM, 2010.

[30] Sven Schewe. From parity and payoff games to linear programming. In *Proc. of the 34th Int. Symp. on Mathematical Foundations of Computer Science (MFCS)*, volume 5734 of *Lecture Notes in Computer Science*, pages 675–686. Springer, 2009.

[31] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[32] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, 2009.

[33] Roman Vershynin. Beyond hirsch conjecture: Walks on random polytopes and smoothed complexity of the simplex method. *SIAM Journal on Computing*, 39(2):646–678, 2009.

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

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

A. Lemmas About Markov Chains

For a situation s, let $d_{G(s)}(u,v)$ be the *stochastic distance* from u to v in $G(s)$, which is the shortest (directed) distance between vertices u and v in the graph obtained from $G(s)$ by setting the length of every deterministic arc to 0 and of every stochastic arc to 1. Let

$$\lambda = \lambda(\mathcal{G}) = \{d_{G(s)}(v,u) \mid v,u \in V,\ d_{G(s)}(v,u) \text{ is finite, and } s \text{ a situation}\}$$

be the *stochastic diameter* of $\mathcal{G}$. Clearly, $\lambda(\mathcal{G}) \leq k(\mathcal{G})$. Some of our bounds will be given in terms of λ instead of k, which implies slightly stronger bounds on the running-times of some of the approximation schemes.

A set of vertices $U \subseteq V$ is called an absorbing class of the Markov chain $\mathcal{M}$ if there is no arc with positive probability from U to $V \setminus U$, i.e., U can never be left once it is entered, and U is strongly connected, i.e., any vertex of U is reachable from any other vertex of U.

Lemma A.1. *Let $\mathcal{M} = (G = (V,E), P)$ be a Markov chain on n vertices with starting vertex u. Then the limiting probability of any vertex $v \in V$ is either 0 or at least $p_{\min}^{2\lambda}/n$ and the limiting probability of any arc $(u,v) \in E$ is either 0 or $p_{\min}^{2\lambda+1}/n$.*

Proof. Let π and ρ denote the limiting vertex- and arc-distribution, respectively. Let C be any absorbing classes of $\mathcal{M}$ reachable from u. We deal with π first. Clearly, for any v that does not lie in any of these absorbing classes, we have $\pi_v = 0$. It remains to show that for all i and every $v' \in C$, we have $\pi_{v'} \geq p_{\min}^{2\lambda}/n$. Denote by $\pi_C = \sum_{v \in C} \pi_v$ the total limiting probability of C. Note that π_C is equal to the probability that we reach some vertex $v \in C_i$ starting from u. Since there is a simple path in G from u to C with at most λ stochastic vertices, this probability is at least $p_{\min}^{\lambda}$. Furthermore, there exists a vertex $v \in C$ with $\pi_v \geq \pi_C/|C| \geq p_{\min}^{\lambda}/n$. Now for any $v' \in C$, there exists again a simple path in G from v to v' with at most λ stochastic nodes, so the probability that we reach v' starting from v is at least $p_{\min}^{\lambda}$. It follows that $\pi_{v'} \geq p_{\min}^{2\lambda}/n$.

Now for ρ, note that $\rho_{(u,v)} \geq \pi_u p_{\min}$, if $(u,v) \in E$. Since π_u is either 0 or at least $p_{\min}^{2\lambda}/n$, the claim follows. □

A Markov chain is said to be *irreducible* if its state space is a single absorbing class. For an irreducible Markov chain, let m_{uv} denote the *mean first passage time* from vertex u to vertex v, and m_{vv} denotes the *mean return time* to vertex v: m_{uv} is the expected number of steps to reach vertex v for the first time, starting from vertex u, and m_{vv} is the expected number of steps to return to vertex v for the first time, starting from vertex v. The following lemma relates these values to the sensitivity of the limiting probabilities of a Markov chain.

Lemma A.2 (Cho, Meyer [10]). *Let $\varepsilon > 0$. Let $\mathcal{M} = (G = (V, E), P)$ be an irreducible Markov chain. For any transition probabilities $\tilde{P}$ with $\|\tilde{P} - P\|_\infty \le \varepsilon$ such that the corresponding Markov chain $\tilde{\mathcal{M}}$ is also irreducible, we have $\|\tilde{\pi} - \pi\|_\infty \le \frac{1}{2}\varepsilon \max_v \frac{\max_{u \neq v} m_{uv}}{m_{vv}}$, where m_{vu} are the mean values defined with respect to $\mathcal{M}$.*

Let $\mathcal{M} = (G = (V, E), P, r)$ be a weighted Markov chain. We denote by $\mu_u(\mathcal{M}) := \sum_{(v,u) \in E} \pi_v p_{vu} r_{vu}$ the limiting average weight, where π is the limiting distribution when u is the starting position. We will write μ_u when $\mathcal{M}$ is clear from the context.

Lemma A.3. *Let $\mathcal{M} = (G = (V, E), P, r)$ be a weighted Markov chain with arc weights in $[r_-, r_+]$, and let $\varepsilon \le \frac{1}{2} p_{\min} = \frac{1}{2} p_{\min}(\mathcal{M})$ be a positive constant. Let $\tilde{\mathcal{M}} = (G = (V, E), \tilde{P}, r)$ be a weighted Markov chain with transition probabilities $\tilde{P}$ such that $\|\tilde{P} - P\|_\infty \le \varepsilon$. Then, for any $u \in V$, we have $|\mu_u(\tilde{\mathcal{M}}) - \mu_u(\mathcal{M})| \le \delta(\mathcal{M}, \varepsilon)$, where δ is defined as in (7).*

Proof. Fix the starting vertex $u_0 \in V$. Let π and $\tilde{\pi}$ denote the limiting distributions corresponding to $\mathcal{M}$ and $\tilde{\mathcal{M}}$, respectively. We first bound $|\pi - \tilde{\pi}|_\infty$. Since $\varepsilon < p_{\min}$, we have $\tilde{p}_{uv} = 0$ if and only if $p_{uv} = 0$. It follows that $\mathcal{M}$ and $\tilde{\mathcal{M}}$ have the same absorbing classes. Let $C_1, \ldots, C_\ell$ denote these classes. Denote by $\pi_{C_i} = \sum_{v \in C_i} \pi_v$ and $\tilde{\pi}_{C_i} = \sum_{v \in C_i} \tilde{\pi}_v$ the total limiting probability of C_i with respect to π and $\tilde{\pi}$, respectively. Furthermore, let $\pi^{|i}$ and $\tilde{\pi}^{|i}$ be the limiting distributions, corresponding to $\mathcal{M}$ and $\tilde{\mathcal{M}}$, respectively, conditioned on the event that the Markov process is started in C_i (i.e., $u_0 \in C_i$). Note that these conditional limiting distributions describe the limiting distributions for the irreducible Markov chains restricted to C_i. By Lemma A.2, we have $|\pi^{|i} - \tilde{\pi}^{|i}|_\infty \le \frac{1}{2}\varepsilon \max_{v \in C_i} \max_{\substack{u \in C_i \\ u \neq v}} \frac{m_{uv}}{m_{vv}}$.

Claim A.4. *For any $u, v \in C_i$, we have $m_{uv} \le \frac{(\lambda+1)|C_i|}{p_{\min}^\lambda}$.*

Proof. Fix $v \in C_i$. Note that, for any $u \in C_i$, we have

$$m_{uv} = \sum_{w \neq v} p_{uw}(1 + m_{wv}) + p_{uv}. \tag{18}$$

Let $h = \max\{d_G(u, v) \mid u \in C_i\}$. For $j = 0, 1, \ldots, h$, let $X_j = \max\{m_{uv} \mid u \in C_i,\ d_G(u, v) = j\}$. Let $\ell = \operatorname{argmax}\{X_j \mid j \in \{1, \ldots, h\}\}$. Then $X_0 \le |C_i|$ and, for $j = 1, \ldots, h$, (18) implies that

$$X_j \le |C_i| + p_{\min} X_{j-1} + (1 - p_{\min}) X_\ell. \tag{19}$$

(For a vertex for $u \in V$ such that $d_G(u, v) = j$, there is a path Q from u to v with j stochastic arcs. Let u' be the vertex closest to u on Q such that $d_G(u', v) = j - 1$, and let u'' be the vertex on Q preceding u'. Then u'' is stochastic, and hence by (18)

$$\begin{aligned}
m_{u''v} &\le p_{u''u'}(1 + X_{j-1}) + \sum_{w \neq u'} p_{u''w}(1 + X_\ell) \\
&= p_{u''u'}(1 + X_{j-1}) + (1 - p_{u''u'})(1 + X_\ell) \\
&\le p_{\min}(1 + X_{j-1}) + (1 - p_{\min})(1 + X_\ell),
\end{aligned}$$

using the fact that $X_j \leq X_\ell$ for all j and $p_{u''u'} \geq p_{\min}$. Finally, $m_{uv} \leq |C_i| - 1 + m_{u''v}$ implies (19).)

Applying (19) for $j = 1, \ldots, \ell$ yields

$$X_\ell \leq |C_i| \cdot \frac{1 - p_{\min}^{\ell+1}}{1 - p_{\min}} + X_\ell(1 - p_{\min}^\ell).$$

This implies that $X_\ell \leq |C_i| \frac{1-p_{\min}^{\ell+1}}{1-p_{\min}} p_{\min}^{-\ell} \leq |C_i|(\lambda+1)p_{\min}^{-\lambda}$. □

It follows that $\|\pi^{|i} - \tilde{\pi}^{|i}\|_\infty \leq \frac{\varepsilon(\lambda+1)|C_i|}{2p_{\min}^\lambda}$.

Claim A.5. $|\pi_{C_i} - \tilde{\pi}_{C_i}| \leq \varepsilon n \lambda p_{\min}^{-\lambda}$.

Proof. Without loss of generality we assume that $u_0 \notin C_i$. For a transient vertex v, let y_v and $\tilde{y}_v$ be the absorption probability into class C_i in $\mathcal{M}$ and $\tilde{\mathcal{M}}$, respectively. In particular $y_{u_0} = \pi_{C_i}$. Let $p_{vC_i} = \sum_{u \in C_i} p_{vu}$. Then we have

$$y_v = \sum_{u \notin C_i} p_{vu} y_u + p_{vC_i}. \tag{20}$$

Similarly,

$$\tilde{y}_v = \sum_{u \notin C_i} \tilde{p}_{vu} \tilde{y}_u + \tilde{p}_{vC_i} = \sum_{u \notin C_i} p_{vu} \tilde{y}_u + \sum_{u \notin C_i} (\tilde{p}_{vu} - p_{vu}) \tilde{y}_u + \tilde{p}_{vC_i}. \tag{21}$$

Write $\Delta_v := |\tilde{y}_v - y_v|$. Subtracting (20) from (21) yields

$$\begin{aligned} \Delta_v &\leq \sum_{u \notin C_i} p_{vu} \Delta_u + \sum_{u \notin C_i} |\tilde{p}_{vu} - p_{vu}| \tilde{y}_u + |\tilde{p}_{vC_i} - p_{vC_i}| \\ &\leq \sum_{u \notin C_i} p_{vu} \Delta_u + (n - |C_i|)\varepsilon + |C_i|\varepsilon = \sum_{u \notin C_i} p_{vu} \Delta_u + \varepsilon n. \end{aligned} \tag{22}$$

Let $h = \max\{d_G(u, C_i) \mid u \notin C_i \wedge d_G(u, C_i) < \infty\}$, where $d_G(u, C_i) = \min\{d_G(u, v) \mid v \in C_i\}$ is the stochastic distance in G from u to C_i. For $j = 0, 1, \ldots, h$, let $X_j = \max\{\Delta_u \mid u \notin C_i \wedge d(u, C_i) = l\}$, and let $\ell = \operatorname{argmax}\{X_j \mid j \in \{1 \ldots, h\}\}$. Then $X_0 = 0$ (since deterministic vertices in $\mathcal{M}$ remain deterministic in $\tilde{\mathcal{M}}$) and, for $j = 1, \ldots, h$, (22) implies

$$X_j \leq \varepsilon n + p_{\min} X_{j-1} + (1 - p_{\min}) X_\ell. \tag{23}$$

Applying this iteratively gives us $X_\ell \leq \varepsilon n \frac{1-p_{\min}^\ell}{1-p_{\min}} p_{\min}^{-\ell} \leq \varepsilon n \lambda p_{\min}^{-\lambda}$. □

Let $v \in V$ be an arbitrary vertex. If v does not lie in any absorbing class, then $\pi_v = \tilde{\pi}_v = 0$. Otherwise, let $v \in C_i$. By the above claims, we have

$$\begin{aligned} \pi_v &= \pi_{C_i} \pi_v^{|i} \leq (\tilde{\pi}_{C_i} + \varepsilon n \lambda p_{\min}^{-\lambda})(\tilde{\pi}_v^{|i} + \frac{\varepsilon}{2}(\lambda+1)|C_i| p_{\min}^{-\lambda}) \\ &\leq \tilde{\pi}_v + \frac{\varepsilon}{2} n p_{\min}^{-\lambda}[\varepsilon n \lambda(\lambda+1) p_{\min}^{-\lambda} + 3\lambda + 1] := \tilde{\pi}_v + \delta'(\mathcal{M}, \varepsilon). \end{aligned}$$

Similarly, we can conclude that $\tilde{\pi}_v \leq \pi_v + \delta'(\tilde{\mathcal{M}}, \varepsilon)$. Note that $p_{\min}(\tilde{\mathcal{M}}) \geq p_{\min}(\mathcal{M})/2$, since $\varepsilon \leq p_{\min}(\mathcal{M})/2$. It follows that

$$\begin{aligned}
|\mu_{u_0}(\mathcal{M}) - \mu_{u_0}(\tilde{\mathcal{M}})| &\leq \sum_{(u,v)\in E} |\pi_u p_{uv} - \tilde{\pi}_u \tilde{p}_{uv}||r_{uv}| \\
&\leq \sum_{(u,v)\in E} (|\pi_u - \tilde{\pi}_u| p_{uv} + \tilde{\pi}_u |\tilde{p}_{uv} - p_{uv}|) r_* \\
&\leq \sum_{(u,v)\in M} (\delta'(\tilde{\mathcal{M}}, \varepsilon) p_{uv} + \tilde{\pi}_u \varepsilon) r_* \\
&\leq (\delta'(\tilde{\mathcal{M}}, \varepsilon) + \varepsilon) n \leq \delta(\mathcal{M}, \varepsilon) r_*,
\end{aligned}$$

which completes the proof. □