A convex programming-based algorithm for mean payoff stochastic games with perfect information

Endre Boros\textsuperscript{1} · Khaled Elbassioni\textsuperscript{2} · Vladimir Gurvich\textsuperscript{1,3} · Kazuhisa Makino\textsuperscript{4}

Received: 21 October 2016 / Accepted: 31 March 2017 / Published online: 7 April 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract We consider two-person zero-sum stochastic mean payoff games with perfect information, or BWR-games, given by a digraph $G = (V, E)$, with local rewards $r: E \to \mathbb{Z}$, and three types of positions: black $V_B$, white $V_W$, and random $V_R$ forming a partition of $V$. It is a long-standing open question whether a polynomial time algorithm for BWR-games exists, even when $|V_R| = 0$. In fact, a pseudo-polynomial algorithm for BWR-games would already imply their polynomial solvability. In this short note, we show that BWR-games can be solved via convex programming in pseudo-polynomial time if the number of random positions is a constant.

Keywords Stochastic games · Perfect information · Mean payoff · Pseudo-polynomial algorithm · Convex programming

Khaled Elbassioni
kelbassioni@masdar.ac.ae

Endre Boros
Endre.Boros@rutgers.edu

Vladimir Gurvich
Vladimir.Gurvich@gmail.com

Kazuhisa Makino
makino@kurims.kyoto-u.ac.jp

1 MSIS Department and RUTCOR, Rutgers University, 100 Rockafellar Road, Livingston Campus, Piscataway, NJ 08854, USA
2 Masdar Institute of Science and Technology, P.O. Box 54224, Abu Dhabi, UAE
3 Higher School of Economics, National Research University, Moscow, Russia
4 Research Institute for Mathematical Sciences (RIMS), Kyoto University, Kyoto 606-8502, Japan
1 Introduction

We consider two-person zero-sum stochastic games with perfect information and mean payoff: let \( G = (V, E) \) be a digraph whose vertex-set \( V \) is partitioned into three subsets \( V = V_B \cup V_W \cup V_R \) that correspond to black, white, and random positions, controlled respectively, by two players, MIN—the minimizer and MAX—the maximizer, and by nature. We also fix a local reward function \( r: E \to \mathbb{Z} \), and probabilities \( p(v, u) > 0 \) for all arcs \((v, u)\) going out of \( v \in V_R \). We assume that \( \sum_{u \in \{v, u\} \in E} p(v, u) = 1 \), for all \( v \in V_R \). Vertices \( v \in V \) and arcs \( e \in E \) are called positions and moves, respectively.

The game begins at time \( t = 0 \) in the initial position \( s_0 = v_0 \). In a general step, in time \( t \), we are at position \( s_t \in V \). The player who controls \( s_t \) chooses an outgoing arc \( e_{t+1} = (s_t, v) \in E \), and the game moves to position \( s_{t+1} = v \). If \( s_t \in V_R \) then an outgoing arc is chosen with the given probability \( p(s_t, s_{t+1}) \). We assume that every vertex in \( G \) has an outgoing arc. In general, the strategy of the player is a policy by which (s)he chooses the outgoing arcs from the vertices (s)he controls. This policy may involve the knowledge of the previous steps as well as probabilistic decisions. We call a strategy stationary if it does not depend on the history and pure if it does not involve probabilistic decisions. For this type of games, it will be enough to consider only such strategies, since these games are known to be (polynomially) equivalent \([3]\) to the perfect information stochastic games considered by Gillette \([15,29]\).

In the course of this game players and nature generate an infinite sequence of edges \( p = (e_1, e_2, \ldots) \) (a play) and the corresponding real sequence \( r(p) = (r(e_1), r(e_2), \ldots) \) of local rewards. There is a global payoff function \( \phi \) that maps any local reward sequence to a real number, and it is assumed that MIN pays MAX the amount \( \phi(r(p)) \) resulting from the play. Naturally, MAX’s aim is to create a play which maximizes \( \phi(r(p)) \), while MIN tries to minimize it (let us note that the local reward function \( r: E \to \mathbb{R} \) may have negative values, and \( \phi(r(p)) \) may also be negative, in which case MAX has to pay MIN \(-\phi(r(p))\)). Let us also note that \( r(p) \) is a random variable since random transitions occur at positions in \( V_R \). Here \( \phi \) stands for the limiting mean payoff

\[
\phi(r(p)) = \lim_{T \to \infty} \inf \sum_{i=1}^{T} \frac{\mathbb{E}[r(e_i)]}{T},
\]

where \( \mathbb{E}[r(e_i)] \) is the expected reward incurred at step \( i \) of the play.

As usual, a pair of (not necessarily pure or stationary) strategies is a saddle point (or equilibrium) if neither of the players can improve individually by changing her/his strategy. The corresponding \( \phi(r(p)) \) is the value \( \mu_{G}(v_0) \) of the game with respect to initial position \( v_0 \). Such a pair of strategies are called optimal; furthermore, it is called uniformly optimal if it provides the value of the game for any initial position. It is known \([15,29]\) that every such game has a pair of uniformly optimal pure stationary strategies. A BWR-game is said to be ergodic if \( \mu_{G}(v) = \mu \) for all \( v \in V \), that is, the value is the same from each initial position.

This class of BWR-games was introduced in \([16]\); see also \([10]\). The special case when \( V_R = \emptyset \), BW-games, is also known as cyclic games. They were introduced for the complete bipartite digraphs in \([31,32]\), for all (not necessarily complete)
A convex programming-based algorithm for mean payoff games on bipartite digraphs in [14], and for arbitrary digraphs in [16]. In fact, BW-games on arbitrary digraphs can be polynomially reduced to BW-games on bipartite digraphs [3]; moreover, the latter class can further be reduced to BW-games on complete bipartite digraphs [11]. A more special case was considered extensively in the literature under the name of parity games [6,7,12,18,20,21], and later generalized also to include random positions in [10]. A BWR-game is reduced to a minimum mean cycle problem in case $V_W = V_R = \emptyset$, see e.g., [23]. If one of the sets $V_B$ or $V_W$ is empty, we obtain a Markov decision process (MDP), which can be expressed as a linear program; see e.g., [30]. Finally, if both are empty, $V_B = V_W = \emptyset$, we get a weighted Markov chain.

For BW-games several pseudo-polynomial and subexponential algorithms are known [6–9,16,18,19,27,33,35–37]; see also [20] for parity games. A very recent result [13] shows that deciding the winner in a parity game can be done in quasi-polynomial time. Besides their many applications (see e.g. [22,28]), all these games are of interest to Complexity Theory: It is known [27,37] that the decision problem “whether the value of a BW-game is positive” is in the intersection of NP and co-NP. Yet, no polynomial algorithm is known for these games, see e.g., the survey by Vorobyov [36]. A similar complexity claim can be shown to hold for BWR-games, see [1,3].

**Main result**

The computational complexity of stochastic games with perfect information is an outstanding open question; see e.g., the survey [34]. While there are numerous pseudo-polynomial algorithms known for the BW-case, it is a challenging open question whether a pseudo-polynomial algorithm exists for BWR-games, as the existence of such an algorithm would imply also the polynomial solvability of this class of games [1].

In [4,5], we gave a pseudo-polynomial algorithm for BWR-games when the number of random positions is fixed. In this note we show that one can obtain a similar result via convex programming, combined with some of the ideas in [4,5].

For a BWR-game $G$ let us denote by $n = |V_W| + |V_B| + |V_R|$ the number of positions, by $k = |V_R|$ the number of random positions, and assume (without loss of generality) that all local rewards are non-negative integers with maximum value $U$ and all transition probabilities are rational with common denominator $D$.

The main result of this paper is as follows.

**Theorem 1** A BWR-game $G$ can be solved in poly($n, U, D^k$) time via convex programming.

This theorem extends the result by Schewe [35], where it was shown that solving BW-games can be reduced to solving linear programming problems with pseudo-polynomial bit length.

According to the results in [4,5], to get a pseudo-polynomial algorithm for BWR-games with constant number of random positions, it is enough to have pseudo-polynomial routines for: (1) solving BW-games; (2) solving ergodic BWR-games; and (3) finding the top and bottom classes in a non-ergodic BWR-game (that is, the sets of
positions with highest and lowest values). The high-level idea of the reduction from solving a BWR-game $G$ to (1)–(3) is to “guess” the ranks of the random positions, when they are ordered in non-increasing order of value (there are constantly many guesses if the number of random positions is constant), then consider the random positions in that order. When considering a given rank $g$, the value of all random positions with rank $g$ is assumed to be a parameter $x$ that can determined by first solving a certain BW-game derived from $G$ by replacing the random positions with rank $g$ by a self-loop with local reward parameter $x$. This way we can identify a set of maximal intervals, in each of which, the values of different positions as functions of $x$ are either constant or equal to $x$ in the entire interval. Since we do not know the real value of $x$, we guess among the identified intervals one that contains $x$; for this interval, we can provide optimal strategies for the positions that have values above the lower bound of the interval, assuming our guess is correct, by suing an algorithm for solving ergodic BWR-games. Each of our guesses above yields a pair of strategies that can be verified for optimality by solving two MDPs.

There are several pseudo-polynomial algorithms for solving BW-games, e.g., [16, 33,37]. One may also use the LP-based algorithm given in [35] to obtain a solution for (1). For (2) we show in Sect. 5 how to obtain the top (resp., bottom) class in a BWR-game, and a pair of strategies solving the game induced by the top (resp., bottom) class. Since in the ergodic case, the top class is whole set of positions, (2) also provides an algorithm for ergodic BWR-games as required in (3).

Our approach for proving Theorem 1 is to express the decision question whether the value of all the positions are above (or below) a certain threshold as a convex program. Then we can apply the Ellipsoid method [24,26] by showing that the separation problem [17] can be solved in polynomial time. To arrive at the convex program, we first use the sufficient and necessary conditions for optimality via potential transformations [3]. Then we replace the “max” and “min” operators in these conditions by a smoothed softmax and softmin as in [35] to obtain a convex program. The details are given in the following sections.

2 Potential transformations and canonical forms

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

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

(2)

It is not difficult to verify that the obtained game $G^x$ and the original game $G$ are equivalent (see [3]). In particular, their optimal (pure stationary) strategies coincide, and their value functions also coincide: $\mu_{G^x} = \mu_G$.

It is known that for BW-games there exists a potential transformation such that, in the obtained game the locally optimal strategies are globally optimal, and hence, the value and optimal strategies become obvious [16]. This result was extended for the more general class of BWR-games in [3]: in the transformed game, the equilibrium
value $\mu_G(v) = \mu_{G_x}(v)$ is given simply by the maximum local reward for $v \in V_W$, the minimum local reward for $v \in V_B$, and the average local reward for $v \in V_R$. In this case we say that the transformed game is in canonical form. To define this more formally, let us use the following notation throughout this section: Given functions $f: E \to \mathbb{R}$ and $g: V \to \mathbb{R}$, we define the functions $M[f], M[g]: V \to \mathbb{R}$.

$$M[f](v) = \begin{cases} \max_{u|(v,u) \in E} f(v,u), & \text{for } v \in V_W, \\ \min_{u|(v,u) \in E} f(v,u), & \text{for } v \in V_B, \\ \sum_{u|(v,u) \in E} p(v,u) f(v,u), & \text{for } v \in V_R. \end{cases}$$

$$M[g](v) = \begin{cases} \max_{u|(v,u) \in E} g(u), & \text{for } v \in V_W, \\ \min_{u|(v,u) \in E} g(u), & \text{for } v \in V_B, \\ \sum_{u|(v,u) \in E} p(v,u) g(u), & \text{for } v \in V_R. \end{cases}$$

We say that a BWR-game $G$ is in (strong) canonical form if there exist vectors $\mu, x \in \mathbb{R}^V$ such that

(C1) $\mu = M[\mu] = M[r_x]$ and,
(C2) for every $v \in V_W \cup V_B$, every move $(v,u) \in E$ such that $\mu(v) = r_x(v,u)$ must also have $\mu(v) = \mu(u)$, or in other words, every locally optimal move $(v,u)$ is globally optimal.

**Theorem 2** [3] For each BWR-game $G$ there is a potential transformation $x \in \mathbb{R}^V$ that brings $G$ to canonical form with $\|x\|_\infty \leq L := nUDO(k)$. Furthermore, in a game in canonical form we have $\mu_G = M[r_x]$.

In this paper, we will provide a convex programming formulation based on the existence of potential transformations.

We will need the following upper bound on the required accuracy.

**Lemma 1** [2,5] For any position $v$ in the top (resp., bottom) class in a BWR-game $G$, the value $\mu_G(v)$ is a rational number with a denominator at most $\Lambda := nD^{O(k)}$.

**Lemma 2** Consider a BWR-game $G$ and denote by $1$ the vector of all ones. Then there exists a potential vector $x \in \mathbb{R}^V$ and $t \in \mathbb{R}$ such that $M[r_x] \geq t1$ if and only if $\mu_G \geq t1$.

**Proof** Indeed, if MAX (MIN) applies a locally optimal strategy $s_W$ in the transformed game $G_x$ then after every own move (s)he will get (pay) at least $t$, while for each move of the opponent the local reward will be at least $t$, and finally, for each random position the expected local reward is at least $t$. Thus, the expected local reward $\mathbb{E}[r_x(e_i)]$ at each step of the play is at least $t$. Hence, by (1), strategy $s_W$ guarantees MAX at least $t$ from any starting position.

The other direction follows from Theorem 2. \qed

A symmetric version of Lemma 2 can also be obtained by similar arguments.

**Lemma 3** Consider a BWR-game $G$. Then there exists a potential vector $x \in \mathbb{R}^V$ and $t \in \mathbb{R}$ such that $M[r_x] \leq t1$ if and only if $\mu_G \leq t1$.
3 The convex programs

The following simple facts relate the softmax (resp., softmin) to the maximum (resp., minimum) of a set of numbers.

**Fact 1** For any numbers $a_1, \ldots, a_n \in \mathbb{R}$ and $b > 1$:

(i) $\max_i a_i \leq \log_b \sum_i b^{a_i} \leq \max_i a_i + \log_b n$;

(ii) $\min_i a_i \geq -\log_b \sum_i b^{-a_i} \geq \min_i a_i - \log_b n$.

**Proof** These follow from the trivial inequalities $b \max_i a_i \leq \sum_i b^{a_i} \leq nb \max_i a_i$. \(\square\)

**Fact 2** Let $\alpha_1, \ldots, \alpha_n$ be given positive numbers such that $\sum_{i=1}^{n} \alpha_i = 1$. Then the function $f(x) = \prod_{i=1}^{n} x_i^{\alpha_i}$ is a concave function of $x$ for $x \geq 0$.

**Proof** Note that for any $x, y \in \mathbb{R}_+^n$, if for some $i$, $x_i = 0$ then for any $\lambda \in [0, 1]$,

$$\lambda f(x) + (1 - \lambda) f(y) = (1 - \lambda) \prod_{i=1}^{n} y_i^{\alpha_i} = \prod_{i=1}^{n} (1 - \lambda) y_i^{\alpha_i}$$

$$\leq \prod_{i=1}^{n} (\lambda_i x_i + (1 - \lambda) y_i)^{\alpha_i} = f(\lambda x + (1 - \lambda)y).$$

Thus, it is enough to show that $\nabla^2 f(x)$ is a negative semi-definite matrix for $x > 0$. Note that

$$\frac{\partial f}{\partial x_i} = \frac{\alpha_i}{x_i} f(x), \text{ for } i = 1, \ldots, n$$

$$\frac{\partial^2 f}{\partial x_i^2} = \frac{\alpha_i(\alpha_i - 1)}{x_i^2} f(x), \text{ for } i = 1, \ldots, n$$

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\alpha_i \alpha_j}{x_i x_j} f(x), \text{ for } i, j = 1, \ldots, n, i \neq j.$$

Consider any $y \in \mathbb{R}^n$. Then

$$y^T \nabla^2 f(x) y = \left( \sum_i \alpha_i (\alpha_i - 1) \frac{y_i^2}{x_i^2} + \sum_{i \neq j} \alpha_i \alpha_j \frac{y_i y_j}{x_i x_j} \right) f(x)$$

$$= \left( \sum_i \alpha_i \frac{y_i}{x_i} \right)^2 - \sum_i \alpha_i \left( \frac{y_i}{x_i} \right)^2 \leq 0,$$

where the last inequality follows from Jensen’s inequality applied to the convex function $f(w) = w^2$. \(\square\)
Given $t \in \mathbb{R}$, let us replace the max operator in the system $M[r_x] \geq t$ by the softmax approximation:

$$\log_b \sum_{u \mid (v, u) \in E} b^{r(v, u) + x(v) - x(u)} \geq t, \quad \text{for } v \in V_W,$$

$$r(v, u) + x(v) - x(u) \geq t, \quad \text{for } u \text{ s.t. } (v, u) \in E \quad \text{for } v \in V_B,$$

$$\sum_{u \mid (v, u) \in E} p(v, u)(r(v, u) + x(v) - x(u)) \geq t, \quad \text{for } v \in V_R,$$

where the constant $b$ will be determined later. Defining the new variables $y(v) := b^{-x(v)}$, we can rewrite the system (3) as follows:

$$\sum_{u \mid (v, u) \in E} b^{r(v, u)} y(u) \geq b^t y(v), \quad \text{for } v \in V_W,$$

$$b^{r(v, u)} y(u) \geq b^t y(v), \quad \text{for } u \text{ s.t. } (v, u) \in E \quad \text{for } v \in V_B,$$

$$\prod_{u \mid (v, u) \in E} (b^{r(v, u)} y(u))^p(v, u) \geq b^t y(v), \quad \text{for } v \in V_R.$$

Note that $y(v) > 0$ if and only if $x(v)$ is finite. In fact, since we may assume by Theorem 2 that $\|x\|_{\infty} \leq L$, we may add also the inequalities:

$$b^{-L} \leq y(v) \leq b^L, \quad \text{for } v \in V.$$

Note that, without the lower bounds $y(v) \geq b^{-L}$, the system (4) is always feasible. As we shall see later, it will be necessary to test the feasibility of the system with $y(v) > 0$ for some $v \in V$. For convenience, let us write more generally the following set of upper and lower bounds, where $V' \subseteq V$ is a fixed subset of positions, to be chosen later:

$$0 \leq y(v) \leq b^L, \quad \text{for } v \in V, \text{ and } y(v) \geq b^{-L}, \quad \text{for } v \in V'.$$

Similarly, we replace the min operator in the system $M[r_x] \leq t$ by the softmin approximation:

$$r(v, u) + x(v) - x(u) \leq t, \quad \text{for } u \text{ s.t. } (v, u) \in E \quad \text{for } v \in V_W,$$

$$-\log_b \sum_{u \mid (v, u) \in E} b^{-r(v, u) + x(v) + x(u)} \leq t, \quad \text{for } v \in V_B,$$

$$\sum_{u \mid (v, u) \in E} p(v, u)(r(v, u) + x(v) - x(u)) \leq t, \quad \text{for } v \in V_R.$$
and defining the new variables \( y(v) := b^x(v) \), we can rewrite the system (6) as follows:

\[
\begin{align*}
    b^{-r(v,u)}y(u) & \geq b^{-t}y(v), \quad \text{for } u \text{ s.t. } (v,u) \in E \quad \text{for } v \in V_W, \quad (7a) \\
\sum_{u|(v,u) \in E} b^{-r(v,u)}y(u) & \geq b^{-t}y(v), \quad \text{for } v \in V_B, \quad (7b) \\
\prod_{u|(v,u) \in E} (b^{-r(v,u)}y(u))^{p(v,u)} & \geq b^{-t}y(v), \quad \text{for } v \in V_R, \quad (7c)
\end{align*}
\]

together with the lower and upper bounds:

\[
0 \leq y(v) \leq b^L, \quad \text{for } v \in V, \quad \text{and } y(v) \geq b^{-L}, \quad \text{for } v \in V'. \quad (8)
\]

### 4 Solving the convex programs

We will use the ellipsoid method [17,24,26]. For this we need to show that the separation problem can be solved in polynomial time. For convenience, let us consider the following relaxation of the convex programs (4)–(5) and (7)–(8):

\[
\begin{align*}
\sum_{u|(v,u) \in E} b^{r(v,u)}y(u) & \geq b^t y(v) - \delta, \quad \text{for } v \in V_W, \quad (9a) \\
b^{r(v,u)}y(u) & \geq b^t y(v) - \delta, \quad \text{for } u \text{ s.t. } (v,u) \in E \quad \text{for } v \in V_B, \quad (9b) \\
\prod_{u|(v,u) \in E} (b^{r(v,u)}y(u))^{p(v,u)} & \geq b^t y(v) - \delta, \quad \text{for } v \in V_R, \quad (9c) \\
0 \leq y(v) & \leq b^L + \delta, \quad \text{for } v \in V', \quad \text{and } b^L y(v) \geq 1, \quad \text{for } v \in V'. \quad (9d)
\end{align*}
\]

where \( \delta \in (0, 1) \) is a rational number that will be chosen appropriately. Let \( K \) and \( K_\delta \) be the set of \( y \in \mathbb{R}^E \) satisfying the systems (4)–(5) and (9), respectively. Similarly, Let \( K' \) and \( K'_\delta \) be the set of \( y \in \mathbb{R}^E \) satisfying the systems (7)–(8) and (10), respectively. For a vector (or scalar) \( y \), we denote by \( |y| \) the bit length of \( y \), that is, the number of binary bits needed to represent \( y \).

In our application, we will set \( b = n^2 \lambda^2 \) and \( t \in [0, U] \) to be a rational number with denominator \( \Lambda = nD^{O(k)} \). In particular, \( b^{\pm t} \) is a rational number of bit length \( \langle b^{\pm t} \rangle = O(U \log n) \). Also by assuming without loss of generality (by scaling \( r \) and
replacing \( U \) by \( UD \) that \( r(v, u) \) is a multiple of \( D \), \( b^{\perp r(v, u)p(v, u)} \) is a rational number of bit length \( \langle b^{\perp r(v, u)} \rangle = O(\Lambda UD \log n) \).

**Lemma 4** For \( 0 < \epsilon \leq b^{-1} \delta \) and any \( y \in K \) (resp., \( y \in K' \)), the box \( B(y) := \{ y' \in \mathbb{R}^n \mid y \leq y' \leq y + \epsilon \cdot 1 \} \) is contained in \( K_\delta \) (resp., \( K'_\delta \)), where \( 1 \) is the \( n \)-dimensional vector of all ones. In particular, if \( K \neq \emptyset \) (resp., \( K' \neq \emptyset \)) then \( K_\delta \) (resp., \( K'_\delta \)) is full-dimensional.

**Proof** We prove the statement for \( K_\delta \); the proof for \( K'_\delta \) is similar. Clearly, \( K \subseteq K_\delta \) for \( \delta \geq 0 \). Furthermore, for any \( y \in K \), any \( y' \in B(y) \) satisfies \( (9) \). Indeed, the left-hand sides of \( (9) \) increase when \( y \) is replaced by \( y' \), while the right-hand sides are \( b^T y'(v) - \delta \leq b^T y(v) + b^\epsilon - \delta \leq b^T y(v) \). Also by \( (5) \), \( y'(v) \leq y(v) + \epsilon \leq b^L + \epsilon \leq b^L + \delta \), so \( y' \) satisfies \( (9) \). \( \square \)

Now we consider the (semi-weak) separation problem for \( K_\delta \) (resp., \( K'_\delta \)).

Given \( \tilde{y} \in \mathbb{Q}^n \) and \( 0 < \delta' \in \mathbb{Q} \), either assert that \( \tilde{y} \in K_\delta \) (resp., \( \tilde{y} \in K'_\delta \)) or find a vector \( c \in \mathbb{Q}^n \) such that \( c^T y + \delta' \geq c^T \tilde{y} \) for all \( y \in K_\delta \) (resp., \( y \in K'_\delta \)).

**Lemma 5** The separation problems for \( K_\delta \) and \( K'_\delta \) can be solved in \( \text{poly}(U, A, \langle \tilde{y} \rangle, (\delta')) \) time.

**Proof** We present the proof for \( K_\delta \); the proof for \( K'_\delta \) is similar. Clearly, we can check in \( \text{poly}(\langle \tilde{y} \rangle, (\delta'), (\delta)) \) time if \( \tilde{y} \) satisfies the linear inequalities \( (9a), (9b) \) and \( (9d) \); if one is violated, the corresponding hyperplane defines a (exact) separator \( c \in \mathbb{Q}^n \) and we are done. Assume therefore that \( \tilde{y} \) satisfies \( (9a), (9b) \) and \( (9d) \). Let us now consider an inequality of the form \( (9c) \), corresponding to \( v \in V_R \), violated by \( \tilde{y} \). Let \( f(y) = \prod_{u|v(u) \in E} (b^r(v, u)y(u))^{p(v, u)} - b^T y(v) := A \cdot g(y) - B \cdot y(v) \), where \( A := \prod_{u|v(u) \in E} b^r(v, u) \), \( B := b^T \) and \( g(y) := \prod_{u|v(u) \in E} y(u)^{p(v, u)} \). Note that \( f(\tilde{y}) < -\delta \) if and only if \( A^D \cdot g(\tilde{y})D < (B \cdot \tilde{y}(v) - \delta)D \), which can be checked in \( \text{poly}(\langle \tilde{y} \rangle, (\delta), (D)) \) time, as both \( A^D \) and \( B^D \) are non-negative integers (in case of \( K'_\delta \), they are rational numbers of denominator at most \( n^{ADU} \)). Without loss of generality, we assume that for every \( u \in V \) there is exactly one edge \( (u, u) \in E \). Indeed, (for this part of the proof) we can set \( p(v, u) = 0 \) whenever \( (v, u) \notin E \), and replace parallel edges by one edge, whose transition probability (resp., local reward) is the sum of the transition probabilities (resp., weighted average of local rewards) of all corresponding parallel edges. Then \( \nabla f(y) := A \cdot \left( \frac{p(v, u)}{y(u)} : u \in V \right) g(y) - B1_v \), where \( 1_v \) is the unit dimensional vector with 1 in position \( v \). It follows from the concavity of \( f(y) \) that, for all \( y \in K_\delta, \ -\delta \leq f(y) \leq f(\tilde{y}) + \nabla f(\tilde{y})^T (y - \tilde{y}) \). This gives a separating inequality:

\[
\nabla f(\tilde{y})^T y > \nabla f(\tilde{y})^T \tilde{y} \quad \text{for all } y \in K_\delta.
\]

It is easy to verify that \( (11) \) remains valid even if \( \tilde{y}(u) = 0 \) for some \( u \in V \). Note that the vector \( \nabla f(\tilde{y}) \) can be irrational (it is irrational whenever \( g(\tilde{y}) \) is). We define a rational approximation \( \tilde{g} \) such that \( \tilde{g} \geq g(\tilde{y}) \geq \tilde{g} - \frac{\delta}{\Lambda} \) and let \( c := A \cdot \left( \frac{p(v, u)}{y(u)} : u \in V \right) \tilde{g} - B1_v \). Since \( r(v, u) \) is assumed to be integer multiple of \( D \), \( A \) is an integer (it is a rational number in case of \( K'_\delta \)) and hence \( \tilde{g} \) can be chosen to be a rational number of bit

\( \square \)
length $\langle \tilde{g} \rangle = \langle A \rangle + \langle \delta' \rangle + \langle \tilde{y} \rangle$. It follows also that $c$ is a rational vector of bit length $\text{poly}(U, D, A, \langle \tilde{y}, \langle \delta' \rangle \rangle)$. Note that

\[ c^T y - \nabla f(\tilde{y})^T y = A \cdot \left( \frac{p(v, u)}{\tilde{y}(u)} : u \in V \right)^T y \cdot (\tilde{g} - g(\tilde{y})) \geq 0 \quad \text{for all } y \in K_\delta, \]

while

\[ c^T \tilde{y} - \nabla f(\tilde{y})^T \tilde{y} = A \cdot \left( \frac{p(v, u)}{\tilde{y}(u)} : u \in V \right)^T \tilde{y} \cdot (\tilde{g} - g(\tilde{y})) \leq \delta'. \]

It follows from (11), (12), and (13) that $c^T y + \delta' \geq c^T \tilde{y}$ for all $y \in K_\delta$. □

**Lemma 6** Given $t \in \mathbb{R}$ and $\delta \in (0, 1)$ we can decide in time $\text{poly}(n, U, D^k, \log \frac{1}{\delta})$ if the system (4)–(5) (resp., (7)–(8)) is infeasible, or find $y(v) \in [b^{-L}, b^L + \delta]$, for $v \in V'$ and $y(v) \in [0, b^L + \delta]$, for $v \in V \setminus V'$, such that the left hand sides of the inequalities in (4) (resp., (7)) are at least $b^L y(v) - \delta$ (resp., $b^{-L} y(v) - \delta$), for all $v \in V$.

**Proof** Let us consider the system (4)–(5) and the corresponding perturbed system (9); the proof for (7)–(8) is similar. Given a polynomial-time algorithm for the separation problem for the convex set $K_\delta$, a circumscribing ball of radius $H$ for $K_\delta$, and any $\epsilon' > 0$, the ellipsoid method terminates in $N := O(n \log \frac{1}{\epsilon'} + n^2 \log |H|)$ calls to the separation algorithm using $\delta' = 2^{-O(N)}$, and either (1) finds a vector $y \in K_\delta$, or (2) asserts that $\text{vol}(K_\delta) \leq \epsilon'$; see e.g., Theorem 3.2.1 in [17]. In the first case, we get a vector $y$ satisfying the conditions in the statement of the lemma. In the second case, we conclude that $K_\delta$ and hence $K$ is empty if $\epsilon' < (b^{-L} \delta)^n$. Indeed by Lemma 4, if $K \neq \emptyset$ and $\epsilon := b^{-L} \delta$, then $\text{vol}(K_\delta) \geq \epsilon^n > \epsilon'$, given a contradiction to the assertion in (2).

By (9d), the radius of the bounding ball can be chosen as $H := 2b^L \sqrt{n}$. Furthermore, the ellipsoid method works only with numbers having precision of $O(N)$ bits. By Lemma 5, the separation problem can be solved in time $\text{poly}(n, U, D^k, \log \frac{1}{\delta})$ and the overall running time is $\text{poly}(n, U, D^k, \log \frac{1}{\delta})$. □

**Remark 1** By raising inequalities (4c) and (7c) to power $D$, we obtain systems of polynomial inequalities. Khachiyan [25,26] gave a polynomial-time algorithm for (approximately) solving a system of convex polynomial inequalities. However, it is not possible to use this algorithm directly to solve the convex programs (4)–(5) and (7)–(8), since the polynomials obtained after raising inequalities (4c) and (7c) to power $D$ are not necessarily convex. For instance, the function $\sqrt{x^2 - z}$ is concave for $x, y, z \in \mathbb{R}_+$, while the function $xy - z^2$ is not.

### 5 A pseudo-polynomial algorithm for $k = O(1)$

Let $G$ be a BWR-game. Let $t_{\text{max}} := \max_{v \in V} \mu_G(v)$ and $t_{\text{min}} := \min_{v \in V} \mu_G(v)$. Define the *top* and *bottom* classes of $G$ as $T := \{ v \in V \mid \mu_G(v) = t_{\text{max}} \}$ and $B := \{ v \in V \mid \mu_G(v) = t_{\text{min}} \}$, respectively.

**Proposition 1** The top and bottom classes necessarily satisfy the following properties.
(i) There exists no arc \((v, u) \in E\) such that \(v \in (V_W \cup V_R) \cap B\), \(u \notin B\);
(ii) there exists no arc \((v, u) \in E\) such that \(v \in (V_B \cup V_R) \cap T\), \(u \notin T\);
(iii) there exists no arc \((v, u) \in E\) such that \(v \in V_W \setminus T\), \(u \in T\);
(iv) there exists no arc \((v, u) \in E\) such that \(v \in V_B \setminus B\), \(u \in B\);
(v) for every \(v \in V_W \cap T\), there exists an arc \((v, u) \in E\) such that \(u \in T\);
(vi) for every \(v \in V_B \cap B\), there exists an arc \((v, u) \in E\) such that \(u \in B\);
(vii) for every \(v \in (V_B \cup V_R) \setminus T\), there exists an arc \((v, u) \in E\) such that \(u \notin T\);
(viii) for every \(v \in (V_W \cup V_R) \setminus B\), there exists an arc \((v, u) \in E\) such that \(u \notin B\).

Proof All claims follow from the existence of a canonical form for \(G\), by Theorem 2. Indeed, the existence of arcs forbidden by (i), (ii), (iii) and (iv), or the non-existence of arcs required by (v), (vi), (vii) and (viii) would violate the value equations (C1) of the canonical form. □

Lemma 7 Consider the convex program defined by (4a)–(4c) [resp., (7a)–(7c)]. Then for \(t := t_{\max}\) (resp., \(t := t_{\min}\)), there is a feasible solution with \(y(v) \geq b^{-L}\) for all \(v \in T\) (resp., \(v \in B\)).

Proof Consider the game \(G[T]\) (resp., \(G[B]\)) induced by the top class \(T\) (resp., the bottom class \(B\)). Let \(\tilde{x} \in \mathbb{R}^T\) (resp., \(\tilde{x} \in \mathbb{R}^B\)) be the potential vector guaranteed by Theorem 2 for the game \(G[T]\) (resp., \(G[B]\)). Extend \(\tilde{x}\) to \(x \in \mathbb{R}^n\) by setting \(x(v) = +\infty\) if \(v \notin T\) (resp., \(x(v) = -\infty\) if \(v \notin B\)). Set \(y(v) := b^{-x(v)}\) for \(v \in V\) (resp., \(y(v) := b^{x(v)}\) for \(v \in V\)). Then \(y(v) \geq b^{-L}\) for all \(v \in T\) (resp., \(v \in B\)). It is also easy to verify by Proposition 1 that the system (4) [resp., (7)] is feasible. Indeed, (4a) is satisfied for every position \(v \in V_W \cap T\) (resp., \(v \in V_B \cap B\)) by the definition of \(x\) and Fact 1:

\[
t_{\max} \leq \max_{u|(v,u) \in E} \{ r(v,u) + x(v) - x(u) \} \leq \log_b \sum_{u|(v,u) \in E} b^{r(v,u) + x(v) - x(u)}
\]

resp., \(- t_{\min} \leq -\min_{u|(v,u) \in E} \{ r(v,u) + x(v) - x(u) \}
\]

\[
\leq \log_b \sum_{u|(v,u) \in E} b^{-r(v,u) - x(v) + x(u)}
\]

Moreover, for \(v \in (V_B \cup V_R) \cap T\) (resp., \(v \in (V_W \cup V_R) \cap B\)) we have (4b) and (4c) [resp., (7b) and (7c)] satisfied by the definition of \(x\) and Proposition 1(ii) [resp., Proposition 1(i)], while for \(v \in V \setminus T\) (resp., \(v \in V \setminus B\)) (4a)–(4c) [resp., (7a)–(7c)] are trivially satisfied. □

In the following we set \(\delta = \delta(t) := \frac{1}{2} b^{-t-L} (1 - \frac{1}{n})\) and \(\varepsilon := \frac{1}{2^{A_t+2}}\). Recall that \(b := n^2 \Lambda^2 = n \frac{1}{2^{2A_t}}\).

Lemma 8 The values \(t_{\max}\) and \(t_{\min}\) can be found in time \(\text{poly}(n, U, D^k)\).

Proof We only show how to find \(t_{\max}\); in a similar fashion we can determine \(t_{\min}\). We apply Lemma 6 in a binary search manner to check the feasibility of the system (7)–(8) for \(V' = V\), \(t \in [0, U]\), and \(\delta(t)\) as specified above. Note that, by Lemma 1,
$t_{\text{max}} \in [0, U]$ can be written as a rational number with denominator less than $2\varepsilon$. So we may restrict our search steps to integer multiples of $2\varepsilon$. We stop the search when the length of the search interval becomes a constant multiple of $2\varepsilon$, and then apply linear search for the remaining small interval.

For each $t$, Lemma 6 states that we can, in polynomial time, either discover that the convex program (7)–(8) is infeasible, or find a $\delta(t)$-approximately feasible solution [that is, a feasible solution for (10)]. Suppose that the system is infeasible. Then Lemma 3 implies that $t_{\text{max}} > t$. On the other hand, if $y \in \mathbb{R}^V$ is a $\delta(t)$-approximately feasible solution for (7)–(8), then as $\delta(t) \leq \frac{1}{2}b^{-L}$, the new vector $y' := 2y$ satisfies $y'(v) \in [b^{-L}, 2b^{L} + b^{-L}]$ for all $v \in V$. Also, $y'$ satisfies (7) within an error of $2\delta(t)$, that is, the left-hand sides of (7a)–(7c), when $y$ is replaced by $y'$, are at least $b^{-t}y'(v) - 2\delta(t) = b^{-t}y'(v) - b^{-t-L}(1 - \frac{1}{n}) \geq b^{-t-\log_b n}y'(v)$. Set $x(v) := \log_b y'(v)$. Then $x$ satisfies (6) with $t$ replaced by $t + \log_b n$. This in turn implies by Fact 1 that $M[r_x] \leq (t + 2\log_b n)1 = (t + 4\varepsilon^2)1 \leq (t + \varepsilon)1$. It follows then from Lemma 3 that $t_{\text{max}} \leq t + \varepsilon$. Recall that we assume both $t$ and $t_{\text{max}}$ are multiples of $2\varepsilon$; hence, $t_{\text{max}} \leq t$.

Since the number of binary search steps is at most $\log \frac{U}{\varepsilon} = O(k \log (knUD))$ and each step requires time $\text{poly}(n, U, \log b, \log \frac{1}{\varepsilon}) = \text{poly}(n, U, D^k)$, the bound on the running time follows.

\[\text{Lemma 9}\quad \text{We can find the top class } T \text{ (resp., bottom class } B) \text{ in time } \text{poly}(n, U, D^k).\]

\[\text{Proof}\quad \text{We can check if a vertex } w \in V \text{ belongs to the top class (resp., bottom class) as follows. We write the convex program (4)–(5) [resp., (7)–(8)] with } t := t_{\text{max}} \text{ (resp., } t = t_{\text{min}}) \text{ and } V' = \{w\}. \text{ Then we check the feasibility of this system by solving the convex program (9) [resp., (10)]. If the system is infeasible then we know by Lemma 7 that } w \notin T \text{ (resp., } w \notin B).\]

Suppose, on the other hand, that $y \in \mathbb{R}^V$ is a feasible solution for (9) [resp., (10)] with $\delta = \delta(t_{\text{max}})$ (resp., $\delta = \delta(t_{\text{min}})$). Then as in the proof of Lemma 8, the new vector $y' := 2y$ satisfies $y'(v) \geq b^{-L} > 0$, and the left-hand sides of (4a)–(4c) [resp., (7a)–(7c)], when $y$ is replaced by $y'$, are at least $b^{-t}y'(v)$ (resp., $b^{-t-\log_b n}y'(v)$).

Now we claim that $V^+ := \{v \in V : y(v) > 0\} \subseteq T$ (resp., $V^- := \{v \in V : y(v) > 0\} \subseteq B$), which would in turn imply that $w \in T$ (resp., $w \in B$). Indeed, constraints (4a)–(4c) [resp., (7a)–(7c)], applied to $y$ replaced by $y'$, imply that (1) if $v \in V_W \cap V^+$ then there exists an arc $(v, u) \in E$ such that $u \in V^+$; (2) if $v \in (V_B \cup V_R) \cap V^+$ then all arcs $(v, u) \in E$ must have $u \in V^+$ [resp., (1) if $v \in V_B \cap V^+$ then there exists an arc $(v, u) \in E$ such that $u \in V^+$; (2) if $v \in (V_W \cup V_R) \cap V^+$ then all arcs $(v, u) \in E$ must have $u \in V^+$]. These imply that the game induced by $V^+$ is well-defined. Furthermore, if we set $x(v) := -\log_b y'(v)$ (resp., $x(v) := \log_b y'(v)$), then $x$ satisfies (3) [resp., (6)] with $t$ replaced by $t - \log_b n$ (resp., $t + \log_b n$). This in turn implies by Fact 1 and Lemma 2 (resp., Lemma 3) that $\mu_G(v) \geq (t - 2\log_b n) = (t - 4\varepsilon^2)$ [resp., $\mu_G(v) \leq (t + 2\log_b n) = (t + 4\varepsilon^2)$] for all $v \in V^+$. Since, by Lemma 1, $t_{\text{max}} \in [0, U]$ (resp., $t_{\text{min}} \in [0, U]$) can be written as a rational number with denominator less than $\frac{1}{2\varepsilon}$, it follows that all its positions in $V^+$ have value at least $t_{\text{max}}$ (resp., at most $t_{\text{min}}$). Indeed, if this is not the case, then Lemma 1 implies that the top and bottom classes in the game induced by $V^+$ differ in value by more than $4\varepsilon^2$. The lemma follows. \[\square\]
Finally, given the top and bottom classes, we can find an optimal pair of strategies in the games induced by $T$ and $B$, as stated in the next lemma.

**Lemma 10** We can find optimal pairs of strategies in the games induced by the top class $T$ and bottom class $B$ in time $\text{poly}(n, U, D^k)$.

**Proof** We prove the lemma only for $T$; the proof for $B$ can be done similarly. We solve two (feasible) systems, $S_1$ defined by (4a)–(4c) on $G[T]$ and $S_2$ defined by (7a)–(7c) on $G[T]$, with $t := t_{\text{max}}$ to within an accuracy of $\delta(t_{\text{max}})$. Let $y^1, y^2 \in \mathbb{R}^T$ be the $\delta(t_{\text{max}})$-approximate solutions to $S_1$ and $S_2$, respectively. By the same arguments as in Lemma 8, the corresponding potential vectors $x^1, x^2$ [defined by $x^1(v) := -\log_b(2y^1(v))$ and $x^2(v) := \log_b(2y^2(v))$] ensure that $M[x^1] \geq (t_{\text{max}} - 4\varepsilon^2)1$ and $M[x^2] \leq (t_{\text{max}} + 4\varepsilon^2)1$. Since $\varepsilon$ is sufficiently small, by Lemmas 2 and 3, the locally optimal strategies defined by the operator $M$ with respect to $x^1$ and $x^2$ give optimal strategies for MAX and MIN in $G[T]$, respectively. \qed

Finally, we obtain Theorem 1 by combining the above lemmas with the algorithm in [4,5].

**Acknowledgements** Part of this research was done at the Mathematisches Forschungsinstitut Oberwolfach during a stay within the “Research in Pairs” Program from July 26, 2015 to August 15, 2015. The research of the third author has been partially funded by the Russian Academy of Excellence Project ‘5-100’. The fourth author was partially supported by Grant-in-Aid for Scientific Research 24106002, 25280004, 26280001 and JST CREST Grant Number JPMJCR1402, Japan.

**References**

1. Andersson, D., Miltersen, P.B.: The complexity of solving stochastic games on graphs. In: Proceedings of the 20th International Symposium on Algorithms and Computation (ISAAC), pp. 112–121 (2009)
2. Boros, E., Elbassioni, K., Gurvich, V., Makino, K.: A pumping algorithm for ergodic stochastic mean payoff games with perfect information. In: Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization (IPCO), pp. 341–354 (2010)
3. Boros, E., Elbassioni, K., Gurvich, V., Makino, K.: On canonical forms for zero-sum stochastic mean payoff games. Dyn. Games Appl. 3(2), 128–161 (2013)
4. Boros, E., Elbassioni, K.M., Gurvich, V., Makino, K.: A pseudo-polynomial algorithm for mean payoff stochastic games with perfect information and a few random positions. In: Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP) Part I, pp. 220–231 (2013)
5. Boros, E., Elbassioni, K., Gurvich, V., Makino, K.: A pseudo-polynomial algorithm for mean payoff stochastic games with perfect information and few random positions. CoRR, arXiv:1508.03431 (2015)
6. Beffara, E., Vorobyov, S.: Adapting Gurvich–Karzanov–Khachiyan’s Algorithm for Parity Games: Implementation and Experimentation. Technical Report 2001-020, Department of Information Technology, Uppsala University. https://www.it.uu.se/research/reports/#2001 (2001)
7. Beffara, E., Vorobyov, S.: Is Randomized Gurvich–Karzanov–Khachiyan’s Algorithm for Parity Games Polynomial? Technical Report 2001-025. Department of Information Technology, Uppsala University. https://www.it.uu.se/research/reports/#2001 (2001)
8. Björklund, H., Vorobyov, S.: Combinatorial structure and randomized sub-exponential algorithm for infinite games. Theor. Comput. Sci. 349(3), 347–360 (2005)
9. Björklund, H., Vorobyov, S.: A combinatorial strongly sub-exponential strategy improvement algorithm for mean payoff games. Discrete Appl. Math. 155(2), 210–229 (2007)
10. Chatterjee, K., Henzinger, T.A.: Reduction of stochastic parity to stochastic mean-payoff games. Inf. Process. Lett. 106(1), 1–7 (2008)
11. Chatterjee, K., Henzinger, M., Krinninger, S., Nanongkai, D.: Polynomial-time algorithms for energy games with special weight structures. Algorithmica 70(3), 457–492 (2014)
12. Chatterjee, K., Jurdziński, M., Henzinger, T.A.: Quantitative stochastic parity games. In: SODA ’04, pp. 121–13. Society for Industrial and Applied Mathematics, Philadelphia (2004)
13. Calude, C.S., Jain, S., Khoussainov, B., Li, W., Stephan, F.: Deciding parity games in quasipolynomial time, to appear in Proceedings of the 49th ACM Symposium on Theory of Computing (STOC) (2017)
14. Eberenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Int. J. Game Theory 8, 109–113 (1979)
15. Gillette, D.: Stochastic games with zero stop probabilities. In: Tucker, A.W., Dreger, M., Wolfe, P. (eds.) Contribution to the Theory of Games III, in Annals of Mathematics Studies, vol. 39, pp. 179–187. Princeton University Press, Princeton (1957)
16. Gurvich, V., Karzanov, A., Khachiyan, L.: Cyclic games and an algorithm to find maximax cycle means in directed graphs. USSR Comput. Math. Phys. 28, 85–91 (1988)
17. Grätschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, New York (1988)
18. Halman, N.: Simple stochastic games, parity games, mean payoff games and discounted payoff games are all LP-type problems. Algorithmica 49(1), 37–50 (2007)
19. Sandberg, S., Björklund, H., Vorobyov, S.: A Combinatorial Strongly Sub-Exponential Strategy improvement Algorithm for Mean Payoff Games. DIMACS Technical Report 2004-05. DIMACS, Rutgers University (2004)
20. Jurdziński, M., Paterson, M., Zwick, U.: A deterministic subexponential algorithm for solving parity games. In: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 117–123. ACM, New York (2006)
21. Jurdziński, M.: Deciding the winner in parity games is in UP ∩ co-UP. Inf. Process. Lett. 68(3), 119–124 (1998)
22. Jurdziński, M.: Games for Verification: Algorithmic Issues, Ph.D. thesis. Department of Computer Science, University of Aarhus (2000)
23. Karp, R.M.: A characterization of the minimum cycle mean in a digraph. Discrete Math. 23, 309–311 (1978)
24. Khachiyan, L.: Polynomial algorithms in linear programming. USSR Comput. Math. Math. Phys. 20, 53–72 (1980)
25. Khachiyan, L.: Convexity and complexity in polynomial programming. In: Proceedings of the International Congress of Mathematicians, pp. 1569–1577. Warsa (1983)
26. Khachiyan, L.: Complexity of Convex Problems of Real and Integer Programming, Ph.D. thesis, The second thesis. Manuscript, Computer Center of the USSR Academy of Sciences (in Russian) (1984)
27. Karzanov, A.V., Lebedev, V.N.: Cyclical games with prohibition. Math. Program. 60, 277–293 (1993)
28. Littman, M.L.: Algorithm for Sequential Decision Making, CS-96-09, Ph.D. thesis. Department of Computer Science, Brown University (1996)
29. Liggett, T.M., Lippman, S.A.: Stochastic games with perfect information and time-average payoff. SIAM Rev. 4, 604–607 (1969)
30. Mine, H., Osaki, S.: Markovian Decision Process. American Elsevier Publishing Co., New York (1970)
31. Moulin, H.: Extension of two person zero sum games. J. Math. Anal. Appl. 5(2), 490–507 (1976)
32. Moulin, H.: Prolongement des jeux à deux joueurs de somme nulle. Bull. Soc. Math. Fr. Mem. 45, 5–111 (1976)
33. Pisaruk, N.N.: Mean cost cyclical games. Math. Oper. Res. 24(4), 817–828 (1999)
34. Raghavan, T.E.S., Filar, J.A.: Algorithms for stochastic games: a survey. Math. Methods Oper. Res. 35(6), 437–472 (1991)
35. Schewe, S.: From parity and payoff games to linear programming. In: Mathematical Foundations of Computer Science 2009, 34th International Symposium, MFCS 2009, Novy Smokovec, High Tatras, Slovakia. Proceedings, pp. 675–686 (2009)
36. Vorobyov, S.: Cyclic games and linear programming. Discrete Appl. Math. 156(11), 2195–2231 (2008)
37. Zwick, U., Paterson, M.: The complexity of mean payoff games on graphs. Theor. Comput. Sci. 158(1–2), 343–359 (1996)