Information Processing Letters 114 (2014) 382–386

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

A note on the approximation of mean-payoff games ☆

Raffaella Gentilini

Department of Mathematics and Computer Science, University of Perugia, Via Vanvitelli 1, I-06123, Perugia, Italy

ARTICLE INFO

Article history:
Received 18 June 2013
Received in revised form 17 January 2014
Accepted 12 February 2014
Available online 26 February 2014
Communicated by A. Muscholl

Keywords:
Mean-payoff games
Approximation algorithms

ABSTRACT

We consider the problem of designing approximation schemes for the values of mean-payoff games. It was recently shown that (1) mean-payoff with rational weights scaled on $[-1, 1]$ admit additive fully-polynomial approximation schemes, and (2) mean-payoff games with positive weights admit relative fully-polynomial approximation schemes. We show that the problem of designing additive/relative approximation schemes for general mean-payoff games (i.e. with no constraint on their edge-weights) is P-time equivalent to determining their exact solution.

1. Introduction

Two-player mean-payoff games are played on weighted graphs[1] with two types of vertices: in player-0 vertices, player 0 chooses the successor vertex from the set of outgoing edges; in player-1 vertices, player 1 chooses the successor vertex from the set of outgoing edges. The game results in an infinite path through the graph. The long-run average of the edge-weights along this path, called the *value* of the play, is won by player 0 and lost by player 1.

The *decision problem* for mean-payoff games asks, given a vertex z and a threshold $\mu \in \mathbb{Q}$, if player 0 has a strategy to win a value at least μ when the game starts in z. The *value problem* consists in computing the maximal (rational) value that player 0 can achieve from each vertex v of the game. The associated *(optimal) strategy synthesis problem* is to construct a strategy for player 0 that secures the maximal value.

Mean-payoff games have been first studied by Ehrenfeucht and Mycielski in [1] and Moulin in [23,24] on the special class of weighted bipartite graphs.[2] In particular, in [1] it was shown that memoryless (or positional) strategies suffice to achieve the optimal value. This result entails that the decision problem for these games lies in NP ∩ coNP [2,30], and it was later shown to belong to[3] UP ∩ coUP [20]. Despite many efforts [13,14,18,22,25,30–32], no polynomial-time algorithm for the mean-payoff game problems is known so far.

Beside such a theoretically engaging complexity status, mean-payoff games have plenty of applications, especially in the synthesis, analysis and verification of reactive (non-terminating) systems. Many natural models of such systems include quantitative information, and the corresponding question requires the solution of quantitative games, like mean-payoff games [5,11,16,33]. Concrete examples of applications include finite-window online string matching [30], streaming editing between two regular languages [4], embedded controller synthesis [5,33], various kinds of scheduling and selection with limited storage [10,30]. Mean-payoff games can even be used for solving the max-plus algebra $Ax = Bx$ problem, which in turn has further applications [14]. Last but not least, mean-payoff

☆ This work was partially supported by INdAM-GNCS.
E-mail address: raffaella.gentilini@dmi.unipg.it.

[1] In which every edge has a positive/negative (rational) weight.

[2] Boros et al. [7] gave recently polynomial reductions from general mean-payoff games to mean-payoff games on bipartite graphs. In [12] Chatterjee et al. defined further reductions from the bipartite case to the complete bipartite case.

[3] The complexity class UP is the class of problems recognizable by unambiguous polynomial time nondeterministic Turing machines [26]. Obviously P ⊆ UP ∩ coUP ⊆ NP ∩ coNP.

http://dx.doi.org/10.1016/j.ipl.2014.02.010
0020-0190/

games have tight connections with important problems in game theory and logic. For instance, parity games [17] and the model-checking problem for the modal μ-calculus [21] are poly-time reducible to mean-payoff games [15], and it is a long-standing open question to know whether these problems are in P.

In [30], Zwick and Paterson defined the first pseudopolynomial algorithm for mean-payoff games,[4] improving on purely exponential solutions proposed in [25,31]. The algorithm in [30] solves the decision problem (resp. value problem) for mean-payoff games in $\mathcal{O}(|E| \cdot |V|^2 \cdot W)$ steps (resp. in $\mathcal{O}(|E| \cdot |V|^3 \cdot W)$ steps), where $|V|$ is the number of vertices in the arena of the game, $|E|$ is the number of edges, and W is the maximum (absolute) weight labeling an edge. Coding mean-payoff games as energy games [8,9], the authors of [22] provided better pseudopolynomial mean-payoff algorithms, running in time $\mathcal{O}(|E| \cdot |V| \cdot W)$ for the corresponding decision problem, and $\mathcal{O}(|E| \cdot |V|^2 \cdot W \cdot (\log |V| + \log W))$ for the value problem. In [3,18], the authors defined a *randomized* algorithm which is both subexponential and pseudopolynomial. In [19], Halman proved that simple stochastic games [13] can be formulated as an LP-type problem, an abstract generalization of linear programming problems.[5] Such a formulation leads to further strongly subexponential (randomized) algorithms for simple stochastic games and mean-payoff games.

Recently, the authors of [6,27] showed that the pseudopolynomial procedures in [22,25,30] can be used to design (fully) polynomial value approximation schemes for certain classes of mean-payoff games: namely, mean-payoff games with positive (integer) weights or rational weights with absolute value less or equal to 1. In this paper, we consider the problem of extending such positive approximation results for general mean-payoff games, i.e. mean-payoff games with weights arbitrary shifted/scaled on the line of rational numbers.

2. Preliminaries and definitions

2.1. Game graphs

A *game graph* is a tuple $\Gamma = (V, E, w, \langle V_0, V_1 \rangle)$ where $G^\Gamma = (V, E, w)$ is a weighted graph and $\langle V_0, V_1 \rangle$ is a partition of V into the set V_0 of player-0 vertices and the set V_1 of player-1 vertices. An *infinite game* on Γ is played for infinitely many rounds by two players moving a pebble along the edges of the weighted graph G^Γ. In the first round, the pebble is on some vertex $v \in V$. In each round, if the pebble is on a vertex $v \in V_i$ $(i = 0, 1)$, then player i chooses an edge $(v, v') \in E$ and the next round starts with the pebble on v'. A *play* in the game graph Γ is an infinite sequence $p = v_0 v_1 \dots v_n \dots$ such that $(v_i, v_{i+1}) \in E$ for all $i \geqslant 0$. A *strategy* for player i $(i = 0, 1)$ is a function $\sigma : V^* \cdot V_i \to V$, such that for all finite paths $v_0 v_1 \dots v_n$ with $v_n \in V_i$, we have $(v_n, \sigma(v_0 v_1 \dots v_n)) \in E$. A *strategy-profile* is a pair of strategies $\langle \sigma_0, \sigma_1 \rangle$, where σ_0 (resp. σ_1) is a strategy for player 0 (resp. player 1). We denote by Σ_i $(i = 0, 1)$ the set of strategies for player i. A strategy σ for player i is *memoryless* if $\sigma(p) = \sigma(p')$ for all sequences $p = v_0 v_1 \dots v_n$ and $p' = v'_0 v'_1 \dots v'_m$ such that $v_n = v'_m$. We denote by Σ_i^M the set of memoryless strategies of player i. A play $v_0 v_1 \dots v_n \dots$ is *consistent* with a strategy σ for player i if $v_{j+1} = \sigma(v_0 v_1 \dots v_j)$ for all positions $j \geqslant 0$ such that $v_j \in V_i$. Given an initial vertex $v \in V$, the *outcome* of the *strategy profile* $\langle \sigma_0, \sigma_1 \rangle$ in v is the (unique) play $\mathsf{outcome}^\Gamma(v, \sigma_0, \sigma_1)$ that starts in v and is consistent with both σ_0 and σ_1. Given a memoryless strategy π_i for player i in the game Γ, we denote by $G^\Gamma(\pi_i) = (V, E_{\pi_i}, w)$ the weighted graph obtained by removing from G^Γ all edges (v, v') such that $v \in V_i$ and $v' \neq \pi_i(v)$.

2.2. Mean-payoff games

A *mean-payoff game* (MPG) [1] is an infinite game played on a game graph Γ where player 0 wins a payoff value defined as the long-run average weights of the play, while player 1 loses that value. Formally, the payoff value of a play $v_0 v_1 \dots v_n \dots$ in Γ is

$$\mathsf{MP}(v_0 v_1 \dots v_n \dots) = \liminf_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} w(v_i, v_{i+1}).$$

The value *secured* by a strategy $\sigma_0 \in \Sigma_0$ in a vertex v is

$$\mathsf{val}^{\sigma_0}(v) = \inf_{\sigma_1 \in \Sigma_1} \mathsf{MP}\big(\mathsf{outcome}^\Gamma(v, \sigma_0, \sigma_1)\big)$$

and the *(optimal) value* of a vertex v in a mean-payoff game Γ is

$$\mathsf{val}^\Gamma(v) = \sup_{\sigma_0 \in \Sigma_0} \inf_{\sigma_1 \in \Sigma_1} \mathsf{MP}\big(\mathsf{outcome}^\Gamma(v, \sigma_0, \sigma_1)\big).$$

We say that σ_0 is *optimal* if $\mathsf{val}^{\sigma_0}(v) = \mathsf{val}^\Gamma(v)$ for all $v \in V$. Secured value and optimality are defined analogously for strategies of player 1. Ehrenfeucht and Mycielski [1] show that mean-payoff games are *memoryless determined*, i.e., memoryless strategies are sufficient for optimality and the optimal (maximum) value that player 0 can secure is equal to the optimal (minimum) value that player 1 can achieve.

Theorem 2.1. *(See [1].) For all MPG $\Gamma = (V, E, w, \langle V_0, V_1 \rangle)$ and for all vertices $v \in V$, we have*

$$\begin{aligned} \mathsf{val}^\Gamma(v) &= \sup_{\sigma_0 \in \Sigma_0} \inf_{\sigma_1 \in \Sigma_1} \mathsf{MP}\big(\mathsf{outcome}^\Gamma(v, \sigma_0, \sigma_1)\big) \\ &= \inf_{\sigma_1 \in \Sigma_1} \sup_{\sigma_0 \in \Sigma_0} \mathsf{MP}\big(\mathsf{outcome}^\Gamma(v, \sigma_0, \sigma_1)\big), \end{aligned}$$

and there exist two memoryless strategies $\pi_0 \in \Sigma_0^M$ and $\pi_1 \in \Sigma_1^M$ such that

$$\mathsf{val}^\Gamma(v) = \mathsf{val}^{\pi_0}(v) = \mathsf{val}^{\pi_1}(v).$$

[4] I.e., polynomial in the number of vertices $|V|$, the number of edges $|E|$, and the maximal absolute weight W, rather than in the binary representation of W.

[5] Further connections between mean-payoff games and linear programming are established in [29].