Theoretical Computer Science 560 (2014) 235–250

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The discrete strategy improvement algorithm for parity games and complexity measures for directed graphs

Felix Canavoi, Erich Grädel, Roman Rabinovich [*,1]

RWTH Aachen University, Germany

A R T I C L E I N F O

Available online 9 January 2014

Keywords:
Parity games
Graph complexity measures
Treewidth

A B S T R A C T

The problem whether winning regions and wining strategies for parity games can be computed in polynomial time is a major open problem in the field of infinite games, which is relevant for many applications in logic and formal verification. For some time the discrete strategy improvement algorithm due to Jurdziński and Vöge had been considered to be a candidate for solving parity games in polynomial time. However, it has recently been proved by Oliver Friedmann that this algorithm requires super-polynomially many iteration steps, for all popular local improvements rules, including switch-all (also with Fearnley's snare memorisation), switch-best, random-facet, random-edge, switch-half, least-recently-considered, and Zadeh's Pivoting rule.

We analyse the examples provided by Friedmann in terms of complexity measures for directed graphs such as treewidth, DAG-width, Kelly-width, entanglement, directed pathwidth, and cliquewidth. It is known that for every class of parity games on which one of these parameters is bounded, the winning regions can be efficiently computed. It turns out that with respect to almost all of these measures, the complexity of Friedmann's counterexamples is bounded, and indeed in most cases by very small numbers. This analysis strengthens in some sense Friedmann's results and shows that the discrete strategy improvement algorithm is even more limited than one might have thought. Not only does it require super-polynomial running time in the general case, where the problem of polynomial-time solvability is open, it even has super-polynomial time lower bounds on natural classes of parity games on which efficient algorithms are known.

1. Introduction

Parity games are a family of infinite games on directed graphs. They are played by two players, player 0 and player 1, whose moves consist in shifting a pebble from a vertex to a vertex along edges. Vertices have marks indicating the player to move, and priorities (colours) from a finite set of natural numbers. A player wins a finite play if it ends in a vertex that belongs to the opponent and has no outgoing edges. Otherwise the players construct an infinite play and thus an infinite sequence of colours. If the greatest infinitely often appearing colour is even, player 0 wins, otherwise player 1 wins.

Parity games are important for several reasons. Many classes of games arising in practical applications admit reductions to parity games (over larger game graphs). This is not only the case for games modelling reactive systems, with winning conditions specified in some temporal logic or in monadic second-order logic over infinite paths (S1S), for Muller games, but also for games with partial information appearing in the synthesis of distributed controllers. Further, parity games arise

* Corresponding author.
E-mail addresses: canavoi@logic.rwth-aachen.de (F. Canavoi), graedel@logic.rwth-aachen.de (E. Grädel), rabinovich@logic.rwth-aachen.de (R. Rabinovich).

1 This work was partially supported by ESF, www.esf.org.

0304-3975/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.tcs.2013.12.021

as the model-checking games for *fixed-point logics* such as the modal μ-calculus or LFP, the extension of first-order logic by least and greatest fixed-points. Conversely, winning regions of parity games (with a bounded number of priorities) are definable in both LFP and the μ-calculus. Parity games are also of crucial importance in the analysis of structural properties of fixed-point logics.

From an algorithmic point of view parity games are highly intriguing as well. It is an immediate consequence of the *positional determinacy* of parity games, that their winning regions can be decided in NP $\cap$ Co-NP. In fact, it was proved in [13] that the problem is in UP$\cap$Co-UP, where UP denotes the class of NP-problems with unique witnesses. The best known deterministic algorithm has complexity $n^{O(\sqrt{n})}$ [15]. For parity games with a number d of priorities the progress measure lifting algorithm by Jurdziński [14] computes winning regions in time $O(dm \cdot (2n/(d/2))^{d/2}) = O(n^{d/2+O(1)})$, where m is the number of edges, giving a polynomial-time algorithm when d is bounded. The two approaches can be combined to achieve a worst-case running time of $O(n^{d/3+O(1)})$ for solving parity games with d priorities, with $d = \sqrt{n}$ (see [1, Chapter 3]).

Although the question whether parity games are in general solvable in PTIME is still open, there are efficient algorithms that solve parity games in special cases, where the structural complexity of the underlying directed graphs, measured by numerical graph parameters, is low. These include parity games of bounded treewidth [18], bounded entanglement [3,4], bounded DAG-width [2], bounded Kelly-width [12], or bounded cliquewidth [19].

One algorithm that, for a long time, had been considered as a candidate for solving parity games in polynomial time is the *discrete strategy improvement algorithm* by Jurdziński and Vöge [16]. The basic idea behind the algorithm is to take an arbitrary initial strategy for Player 0 and improve it step by step until an optimal strategy is found. The algorithm is parametrised by an *improvement rule*. Indeed, there are many possibilities to improve the current strategy at any iteration step, and the improvement rule determines the choice that is made. Popular improvement rules are switch-all, switch-best, random-facet, random-edge, switch-half and Zadeh's Pivoting rule. Although it is open whether there is an improvement rule that results in a polynomial worst-case runtime of the strategy improvement algorithm, Friedmann [8] was able to show that there are super-polynomial lower bounds for all popular improvement rules mentioned above. For each of these rules, Friedmann constructed a family of parity games on which the strategy improvement algorithm requires super-polynomial running time.

In this paper we analyse the examples provided by Friedmann in terms of complexity measures for directed graphs. It turns out that with respect to most of these measures, the complexity of Friedmann's counterexamples is bounded, and indeed in most cases by very small numbers. This analysis strengthens in some sense Friedmann's results and shows that the discrete strategy improvement algorithm is even more limited than one might have thought. Not only does it require super-polynomial running time in the general case, where the problem of polynomial-time solvability is open, it even has super-polynomial lower time bounds on natural classes of parity games on which efficient algorithms are known.

2. The strategy improvement algorithm

We assume that the reader is familiar with basic notions and terminology on parity games. We shall now briefly discuss the discrete strategy improvement algorithm and the different improvement rules that parametrise it. For the purpose of this paper a precise understanding of the algorithm is not needed. The idea of the strategy improvement algorithm is that one can compute an optimal strategy of a player by starting with an arbitrary initial strategy and improve it step by step, depending on a discrete valuation of plays and strategies, and on a rule that governs the choices of local changes (switches) of the current strategy.

It is well-known that parity games are determined by positional strategies, i.e. strategies which at each position just select one of the outgoing edges, independent of the history of a play. The discrete valuation defined by Jurdziński and Vöge [16] measures how good a play is for Player 0 in a more refined way than just winning or losing. Given a current strategy one can then, at each position of Player 0, consider the possible local changes, i.e. the switches of the outgoing edges, and select a locally best possibility. Rules that describe how to combine such switches in one improvement step are called *improvement rules* and parametrise the algorithm.

The *switch-all* or *locally optimising* rule [16] regards each vertex independently and performs the best possible switch for every vertex. In other words, for every vertex, it computes the best improvement of the strategy *at that vertex* assuming that the strategy remains unchanged at other vertices. However, the switch is done simultaneously at each vertex. The *switch-best* or the *globally optimising rule* takes cross-effects of improving switches into account and applies in every iteration step a best possible combination of switches.

The *random-edge* rule applies a single improving switch at some vertex chosen randomly and the improvement rule *switch-half* applies an improving switch at every vertex with probability 1/2. The *random-facet* rule chooses randomly an edge e leaving a Player 0 vertex and computes recursively a winning strategy σ^* on the graph without e. If taking e is not an improvement, σ^* is optimal, otherwise σ^* switched to e is the new initial strategy. The *least-entered* rule switches at a vertex at that the least number of switches has been performed so far. Cunningham's *least-recently-considered* or *round-robin* rule fixes an initial ordering on all Player 0 vertices first, and then selects the next vertex to switch at in a round-robin manner. Fearnley introduced *snare memorisation* in [7]. It can be seen as an extension of a basic improvement rule by a snare rule that memorises certain structures of a game to avoid reoccurring patterns.

All local improvement rules discussed here can be computed in polynomial time [16,21]. Hence the running time of the algorithm on a game depends primarily on the number of improvement steps. In a series of papers and in his dissertation,