Two Adversary Lower Bounds for Parity Games

Henrik Björklund & Sergei Vorobyov

Department of Information Technology
Uppsala University
Box 337, SE-751 05 Uppsala, Sweden

Technical report 2002-008
February 2002
ISSN 1404-3203

Abstract

By using the adversary arguments we settle the first exponential lower bounds for restricted classes of algorithms solving Parity Games. The first result applies to any algorithms that rely only on estimating values of vertices from the viewpoint of one player and ignore the game graph structure (a rough abstraction of different fix-point algorithms), the second settles the lower bound for a randomized algorithm that samples from the set of optimal counterstrategies (a popular idea used in many approaches).

Contents

1 Introduction

Parity Games is a well-known fundamental open problem in complexity theory with the intriguing and rare status of belonging to the complexity class NP∩coNP, with unknown but widely conjectured PTIME membership. Given a finite directed graph without sinks with vertices colored by natural numbers, two adversaries, Player 0 and Player 1, alternate moves in an infinite play. Player 0 tries to make sure that the largest color visited infinitely often is *even*, while Player 1 tries to make it *odd*. The Parity Games problem consists in determining the winner in such a game, which always exists, i.e., Parity Games are *determined*.

The Parity Games problem is known to be equivalent to the so-called *propositional μ-calculus model checking*, a fundamental problem in temporal logic, automata theory, and program verification. The modal μ-calculus is a powerful logic expressively subsuming virtually all known temporal logics of programs (Emerson 1997, Emerson 1990). Therefore, settling this problem represents both theoretical and practical interests.

Solving Parity Games means finding the so-called *optimal strategies* (or *Nash equilibria*), a problem whose complexity is a fundamental open question in complexity theory, as stressed by Papadimitriou (2001): *"The most interesting aspect of the Nash equilibrium concept to our community is that it is a most fundamental computational problem whose complexity is wide open. Suppose that two players have finitely many strategies each. Is there a polynomial algorithm for computing a (mixed) Nash equilibrium in such a game? [...] Together with factoring, the complexity of finding a Nash equilibrium is in my opinion the most important concrete open question on the boundary of P today."* Not surprisingly, the precise complexity characterization of Parity Games remains open.

Several approaches to solving Parity Games were extensively studied in the literature. The first one is based on the equivalence with the μ-calculus and computing fixpoints of monotonic operators in lattices, represented by Browne, Clarke, Jha, Long & Marrero (1997), Browne et al. (1997), Seidl (1996), Jurdzinski (2000); see also the survey (Emerson 1997). The second approach uses the iterative discrete strategy improvement algorithms (Vöge & Jurdzinski 2000). The third approach is based on a reduction to Mean Payoff Games problem (Puri 1995, Jurdziński 1998) and using a pseudopolynomial dynamic programming algorithm of Zwick & Paterson (1996). Other approaches are based on randomized simplex-like subexponential algorithms (Petersson & Vorobyov 2001*b*), on interior-point optimization schemes (Petersson & Vorobyov 2001*a*), or on pseudoboolean optimization of completely unimodal functions (Björklund, Petersson & Vorobyov 2001). Beffara & Vorobyov (2001) adapt and randomize an algorithm due to Gurvich, Karzanov & Khachiyan (1988) for solving the so-called Mean Payoff Games or Infinite Cyclic Games with Mean Payoff, a more general problem than Parity Games.

Despite all the conjectures, the Parity Games problem resists all the attacks, and all algorithms appear to have *exponential* worst-case complexity. For almost all algorithms explicit examples revealing their exponential behavior are constructed or there are no polynomial upper bounds in view. On the other hand, no nontrivial *lower bounds* for the Parity Games problem are currently known. This is clear, because settling a superpolynomial such bound will immediately imply P$\neq$NP.

However, obtaining nontrivial lower bounds seems plausible for *restricted* classes of algorithms. This approach appears useful, since all known algorithms rely on specific techniques, like fixpoint iteration or local search, and use limited means for solving games. Settling nontrivial lower bounds for such restricted classes will provide evidence for the inherent difficulty of the problem, and provably demonstrate the necessity of new or combined approaches. For comparison, this approach proved fruitful for settling exponential lower bounds for general purpose optimization, using the restricted black-box model of function evaluations only (i.e., not exploiting the structure of the function); see, e.g., (Hirsch, Papadimitriou & Vavasis 1989).

In this paper we present two first nontrivial, *exponential*, lower bounds for restricted algorithms based on adversary arguments. The first result in Section 3 shows that any algorithm that does not take the graph structure of the game into account and relies only on the *values* of vertices is necessarily *exponential*. To our knowledge, this is the first result of this kind. Although it is intuitively clear that a good algorithm should exploit the graph structure of the problem as much as possible, until now nobody proved that it is really necessary. Moreover, many really existing algorithms for games do not exploit or rely very weakly on the graph structure. Most of the algorithms enumerated above rely on 'values' of vertices under given strategies and proceed by iteratively improving such values. Although our result in its pure form does not directly apply to those algorithms (they actually combine a lot of tricky and ingenious, not so easy to explicitly postulate techniques), it is somehow indicative showing that graph structure should be taken into account more seriously by good algorithms.

Our second exponential lower bound in Section 4 applies to any *randomized* algorithm that exploits the structure of the game graph in a restricted way, by submitting trial strategies to probe the best possible counterstrategies of the second player. The aim of this probing consists in forcing the second player to reveal his best possible strategy. Once this best strategy is known, the algorithms terminates successfully solving the game. This is an attractive idea easy to implement because optimal counterstrategies for the second player against probe strategies can be computed in polynomial time by linear programming or graph algorithms. Intuitively, such an algorithm should converge fast (we prove the opposite), and in fact it is an abstraction of several iterative improvement algorithms for games. Some algorithms for Simple Stochastic Games (Hoffman & Karp 1966, Condon 1992, Condon 1993) and the Parity Games algorithms of Vöge & Jurdzinski (2000), Petersson & Vorobyov (2001*a*), Björklund et al. (2001) (loosely[1]) fall in this category.

It would be interesting and important to extend the two first lower bounds of this paper to broader and more realistic classes of algorithms for Parity Games.

2 Preliminaries

2.1 Parity Games

Let $G = (V, E)$ be a finite, directed graph without sinks, $V_0 \cup V_1$ be a partition of V, and c be a function from V to $\mathbb{N}$. Then the tuple (V_0, V_1, E, c) is called a Parity Game. The function c is called the *coloring function* of the game.

Given an initial vertex $v_0 \in V$, two parties, Player 0 and Player 1, start moving a token, first placed on v_0, along edges of the graph. If the vertex where the token is placed is in V_0, Player 0 chooses a move and if it is in V_1, then Player 1 moves. The game is infinite. Note that since G has no sinks, the players can never get stuck. The objective of Player 0 is to ensure that the maximal color of any vertex encountered infinitely often during the game is *even*, while Player 1 tries to ensure that it is *odd*. Formally, any play can be seen as an infinite sequence $P = v_0, v_1, v_2, \ldots$ of vertices, such that $(v_i, v_{i+1}) \in E$ for all $i \geq 0$. Define the set

$$Inf(P) = \{ \ col \mid card\{i \mid c(v_i) = col\} \ = \ \infty\}.$$

Then Player 0 wins iff the largest number in $Inf(P)$ is even.

A stationary strategy for a Player i is a selection of *exactly one* outgoing edge from each vertex in V_i. By following his stationary strategy a player

[1]We stress once again that we do not claim direct applicability of our results to specific algorithms which are not so easy to explain or formalize.

always selects a unique chosen edge from every vertex, independently of the history of the game.

Parity Games are known to be *determined*: for every game and every vertex one of the players always has a stationary winning strategy. This immediately implies NP∩coNP membership: it suffices to guess a winner and his winning strategy.

2.2 Adversary Arguments for Lower Bounds

Given a problem we want to prove a lower bound on the time required in the worst case by any algorithm from a given class that correctly solves the problem on all instances. The idea of adversary arguments for lower bounds is well known. We start an algorithm on a input that is initially *unspecified, except for its size*. Whenever the algorithm probes the input, a *malicious adversary* answers it in a way to keep the algorithm working as long as possible before producing a correct answer, while keeping consistency. If the algorithm stops producing the answer *before* probing the input a sufficient number of times, the adversary should be able to produce an actual input consistent with all algorithm's queries and showing that the algorithm is *incorrect*. Since this could have been the actual input for the algorithm, this establishes the required lower bound on the number of probes and hence on the running time of the algorithm.

Numerous applications of this techniques are well know, but *not for Parity Games*. Some lower bounds are known for *particular* algorithms, not for classes of algorithms, and no lower bounds were known for *randomized* algorithms; see our Section 5, where we settle the first such result.

A word of warning. Since proving an exponential lower bound on the problem in NP∩coNP would imply P≠NP, we should stress once again, that we are aiming at less ambitious proving lower bounds for *specific, restricted* classes of algorithms for Parity Games. Recall, for example, that the $\Omega(n \log(n))$ lower bound for sorting does not hold for non-comparison based algorithms.

3 Exponential Lower Bound for Structure Ignoring Algorithms

3.1 Algorithm Model

We start by presenting a hypothetical algorithm-adversary model by making the following rather strong assumptions on the information known and used by the Algorithms[2] while solving a game, and the rules obeyed by the malicious Adversary during interaction.

1. The Adversary generates a binary Parity Game[3] and communicates to the Algorithm full information about the game vertices (ownership and colors), but *does not reveal* the edges (successor relation) of the game graph. Throughout the interaction it is only known to the Algorithm that all vertices of Player 0 have outdegree at most two.

2. The Algorithm tries to find out an optimal strategy of Player 0 by presenting to the Adversary a sequence of probe strategies of Player 0 (bit vectors with 0 and 1 meaning 'go left or right').

3. The Algorithm may remember and use the full history of his previous queries and Adversary's answers.

4. The Adversary, given any probe strategy σ of Player 0 generated by the Algorithm, is obliged to give an *honest response*, by saying which vertices will be winning or losing and with which colors when Player 1 (knowing full information about the game) provides an *optimal response counterstrategy* $\tau(\sigma)$ against the probing strategy σ. Here 'optimality' means that Player 1 tries to win against σ in as many vertices as possible, with maximal possible odd colors visited infinitely often.

5. The Adversary tries to keep the Algorithm guessing as long as possible about the optimal strategy of Player 0 before producing a correct answer.

6. When the interaction ends, the Adversary must reveal his 'initially' generated game, consistent with all queries and answers made during the interaction process.

[2]Actually we define the whole class of algorithms here by allowing *any algorithm* satisfying these properties.

[3]I.e., the outdegree of every vertex is at most two; this is no generality loss.

3.2 Exponential Lower Bound

Theorem 3.1 *In the oracle model described above there is an Adversary that generates a parity game on $n+1$ vertices with vertices colored 0 and 1 only such that any Algorithm that makes less than $2^n - 1$ probe queries to the Adversary cannot discover the optimal strategy for Player 0.*

Thus any algorithms that does not take the structure of the game graph into consideration and relies only on the brute force strategy probing has an exponential running time lower bound. Although it is intuitively clear that any good algorithm should take the structure of the game graph into consideration, this is the first to our knowledge lower bound result for PARITY GAMEs of this kind. Different algorithms make different use of the knowledge about the structure of the game graph, and quite usually the use of this knowledge is not completely formalized or made explicit. Very often this information is quite limited or unspecified.

It would be interesting to make explicit and to formalize the knowledge different known algorithms for PARITY GAMEs make about the structure of the game graphs, and try to present stronger results similar to Theorem 3.1, by relaxing the constraints on the Algorithm in the oracle model of Section 3.1.

Proof of Theorem 3.1. We construct an explicit example of the game (for each $n \in \mathbb{N}$) and the Adversary. Consider the Parity Game in Figure 1. First notice that only Player 0 (moving from round vertices) has choices. It is clear that only one strategy is optimal for Player 0, namely the one which avoids moving to the vertex of color 1 altogether. Using this strategy, Player 0 can win in all but one vertex, and by using any other strategy he looses in all vertices. Thus there are 15 non-optimal strategies and only one optimal. The game can be straightforwardly extended to any number n of round vertices, yielding a game with 1 optimal strategy and $2^n - 1$ non-optimal ones. It is also important to notice that all strategies except the optimal one are equally bad, leading to a loss in every vertex with color 1 hit infinitely often, when Player 1 responds optimally. Thus non-optimal strategies are indistinguishable.

Assume that we have constructed an Algorithm (in the model of Section 3.1) that finds an optimal strategy for Player 0 in a game with $n+1$ vertices performing less than $2^n - 1$ queries.

Now we let a malicious Adversary respond with the vector consisting of $n+1$ ones to any probe strategy σ of Player 0 submitted by the Algorithm.

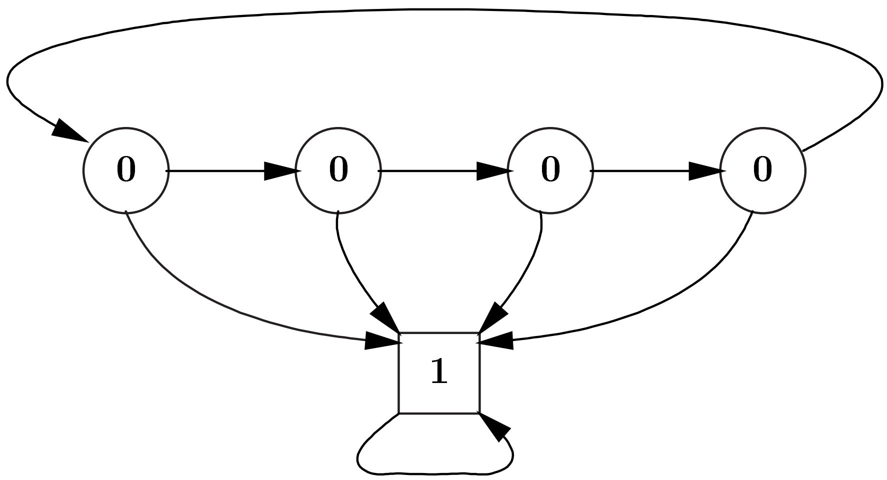

Figure 1: A Parity Game with only one optimal strategy for Player 0.

(This means that the value of every vertex – the color hit infinitely often when Player 1 responds with an optimal strategy $\tau(\sigma)$ to σ – is one.)

If the Algorithm terminates in fewer than $2^n - 1$ steps claiming having found an optimal strategy of Player 0, the Adversary proves it incorrect by revealing a game of the form presented in Figure 1. The Adversary numbers the vertices and labels the edges in such a way that the only optimal strategy is a strategy that has never been submitted by the Algorithm. Such a labeling always exists, since there are 2^n strategies in total and the Algorithm has only submitted at most $2^n - 2$ strategies. Thus no algorithm in the above model that uses less than $2^n - 1$ oracle queries can be correct. This settles the exponential lower bound on the PARITY GAMES problem in the model of algorithms from Section 3.1. $\square$

4 Exponential Lower Bound for a Randomized Optimal Response Algorithm

4.1 Optimal Response Algorithm-Adversary Model

In this section we consider a more 'clever' randomized algorithm for solving Parity Games, which knows the structure of the game graph. This algorithm tries to solve games by presenting probe strategies of Player 0 and asks the oracle[4] to produce one of the optimal response strategies for Player 1, uses those responses to improve its probes, and iterates towards constructing an optimal strategy of Player 0. We may assume that the algorithm keeps track of all strategies used for probing and never repeats itself. Consequently, it is correct by always finding an optimal strategy of Player 0. Although the only immediate upper bound for this algorithm is exponential (by exhaustion), it is tempting to think that this algorithm should perform well in practice by converging very fast, and several algorithms mentioned in the introduction exploit this idea. It can be seen as an abstraction (or a particular case tailored for Parity Games) of realistic algorithms, like Karp-Hoffman's for Simple Stochastic Games; see (Condon 1993). Another close example is the algorithm of Vöge & Jurdzinski (2000). Although we do not claim in this paper that our lower bound applies to the above algorithms (which are considerably more elaborate versions of our oversimplified scheme), our result precludes an easy proof of subexponential behavior of those algorithms, and may serve a basis of extending the lower bounds for more sophisticated algorithms.

More precisely we consider the following Algorithm-Adversary model:

1. The Algorithm probes the game by submitting trial strategies σ_i of Player 0.

2. To every trial strategy σ_i the Adversary is obliged to present an *optimal response* counterstrategy $\tau(\sigma_i)$ of Player 1 against the strategy σ_i.

   (Here we understand optimality as the ability of $\tau(\sigma_i)$ to win against σ_i in as many vertices as possible, with maximal possible odd colors, and when impossible, to lose with smallest possible even colors.)

3. The Algorithm *randomly* selects its next probe strategy σ_{i+1} among *optimal counterstrategies* against the previous optimal response $\tau(\sigma_i)$.

[4] Actually, the algorithm can substitute the oracle by computing optimal responses by itself in polynomial time.

Remark. It is important to note that:

1. If the Adversary is not required to respond with an optimal counterstrategy, then no Algorithm in this model is able to construct an optimal strategy of Player 0, since the Adversary may give the same trivially bad counterstrategy to any probe of the Algorithm.

2. If, on the other hand, the Adversary fairs poorly by *immediately* revealing an *optimal* strategy τ^* of Player 1, then the Algorithm stops in *just one round*. Indeed, an optimal response strategy $\sigma^* \equiv \sigma(\tau^*)$ is constructible in polynomial time, and when presented as a probe to Player 1, the latter cannot beat it in more vertices than τ^*.

3. The goal of the Algorithm therefore can be seen as forcing the Adversary to reveal an optimal strategy of Player 1 as early as possible, while the Adversary tries to keep it secret as long as possible. □

4.2 Exponential Space of Optimal Responses

As a crucial preparation step to the randomized lower bound in the next section, where we construct an exponential run of the above algorithm, we prove the following

Lemma 4.1 *For every $n \in \mathbb{N}$ there exists a Parity Game G on $2n + 1$ vertices colored in just two colors 0 and 1 such that:*

1. *there exists just one optimal strategy σ^* of Player 0 in G,*

2. *for any other strategy $\sigma \neq \sigma^*$ there exists an optimal counterstrategy $\tau(\sigma)$ of Player 1 such that the set $\Sigma(\tau(\sigma))$ of optimal counterstrategies $\sigma(\tau(\sigma))$ of Player 0 against $\tau(\sigma)$ has cardinality 2^{n-1}.*

Proof. Consider the Parity Game shown in Figure 2. Each vertex, except the leftmost one, has outdegree two. From all round vertices in the right part of the game graph Player 0 can force a win by moving right and always staying in this part of the graph. This is his unique optimal strategy σ^*. From the square vertices in the left part, however, Player 1 can win by going left to the leftmost vertex. Thus all vertices on the left are winning for Player 1.

Suppose that Player 0 picks an initial strategy σ different from the unique optimal σ^*. Then in at least one vertex v, he goes left, instead of staying

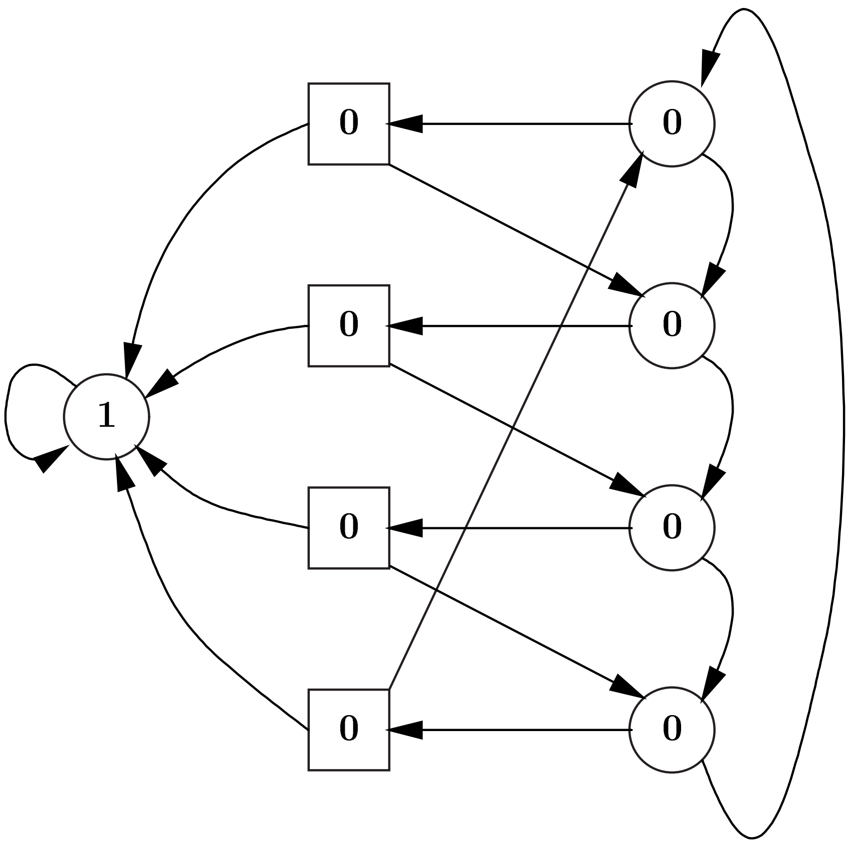

Figure 2: Game where optimal responses of Player 1 to non-optimal strategies of Player 0 do not reveal the optimal strategy of Player 1.

in the right part of the game. Now the Adversary constructs an optimal counterstrategy $\tau(\sigma)$ of Player 1 in the following way: in the vertex u that is a direct successor of v he goes left, to the loop over color 1, and from all other vertices he goes right. If the players use these strategies σ and $\tau(\sigma)$, then Player 1 wins in all vertices since there is path from each vertex, via v and u, to the 1-loop. Thus $\tau(\sigma)$ is indeed an optimal counterstrategy *against* σ. It is not *absolutely optimal*, however, because it loses in all but one square vertices to σ^*. On the other hand, when Player 1 uses $\tau(\sigma)$, Player 0 cannot win only from vertex u and from the vertex with color 1, but wins from all other vertices by simply changing his choice in vertex v, to avoid getting to u. In all other vertices, it does not matter what he does. Any choice in these vertices will avoid the vertex with color one, as long as Player 1 sticks to $\tau(\sigma)$. Thus in Figure 2 there are eight optimal counterstrategies to $\tau(\sigma)$, only one of which is truly optimal. The game can be generalized to have any number n of square vertices, and $n+1$ round vertices. Using the same argument, we can conclude that after Player 0 has guessed an incorrect initial strategy σ, Player 1 can construct a counterstrategy $\tau(\sigma)$ such that there are 2^{n-1} optimal counterstrategies to $\tau(\sigma)$, but only one truly optimal strategy for Player 0. This concludes the proof. $\square$

5 Exponential Counter-Counter Strategy Chain

In this section we extend the previous result and show that the expected running time of the Random Optimal Counterstrategy Algorithm from Section 4.1 (selecting its next probe strategy randomly from the set of optimal counterstrategies to the previous oracle response) is exponential. We do it by constructing an exponentially long sequence of probe strategies not revealing the optimal one. The proof is based on the fact that by Lemma 4.1 the Adversary does not reveal the optimal (always go left) strategy τ^* unless the Algorithm guesses the unique optimal (out of exponentially many) strategy σ^*. Thus, what the Algorithm actually does is uniform random sampling without return with success probability $1/2^{n-1}$. Its expected running time is therefore exponential.

Theorem 5.1 *In the model described in Section 4.1 there is a family of Parity Games on $2n+1$ vertices with colors 0 and 1, and Adversaries such that the expected running time of the Random Optimal Counterstrategy Algorithm is exponential.*

Proof. Consider again the Parity Game in Figure 2. We want to show that in this game (generalized to have n square and $n+1$ round vertices), we can construct a sequence $S = \sigma_0, \tau_0, \sigma_1, \tau_1, \sigma_2, \tau_2, ...$ of strategies for Player 0 and Player 1, with the following properties:

1. S is *exponentially long* in n.
2. For all $i > 0$, σ_i is an *optimal counterstrategy* of Player 0 against τ_{i-1}.
3. For all $i \geq 0$, τ_i is an *optimal counterstrategy* of Player 1 against σ_i.
4. All σ_i are *different.*
5. No strategy in the sequence is *optimal.*

We construct this sequence in the following way:

1. Let σ_0 be any suboptimal strategy for Player 0.
2. Given σ_i, we construct an optimal counterstrategy τ_i in the way described in Section 4.
3. As was shown in Section 4, for each τ_i constructed as above, there are 2^{n-1} *optimal counterstrategies* of Player 0 against τ_i, of which only one is *optimal*. For σ_{i+1} we can pick any one of them that has not appeared earlier in our sequence.

We do not demand that all τ_i are different. In fact they will repeat themselves when the sequence is constructed as above, since there are only n different strategies for Player 1 that can be constructed as in Section 4.

To show that S can indeed be exponentially long we consider a concrete example. We select two round vertices, v_i and v_j, and their square successors u_i and u_j. Let σ_0 be such that from v_i it prescribes going left, and construct τ_0 so that it prescribes going left from u_i. The next strategy for Player 0 must go right from v_i, so we pick one that goes left in v_j and construct a τ_1 that goes left from u_j. In the following sequence we alternate between strategies for Player 0 that go left in v_i and right in v_j and strategies that go right in v_i and left in v_j. Since there are $n-2$ other vertices where Player 0 has a choice, and the choices in these vertices do not matter, there are 2^{n-2} different strategies of each kind, for a total of 2^{n-1} different strategies for Player 0. Thus our sequence can have 2^{n-1} strategies for Player 0 without repeating itself and without containing an optimal strategy. For Player 1 we only alternate between two different strategies (the one that goes left only in u_i and the one that only goes left in u_j). $\square$

6 Conclusions

We presented two first adversary lower bounds for the restricted, yet representative, classes of algorithms for solving the Parity Games problem. These lower bounds clearly indicate that a successful polynomial-time algorithm for the problem should exploit the structure of the game graph and cannot simply rely on vertex values. Our second result clearly demonstrates that random sampling from the set of optimal counterstrategies, although quite appealing and used in several approaches, provably does not help. We plan to reinforce and sharpen these results and apply them to more realistic classes of algorithms.

References

Beffara, E. & Vorobyov, S. (2001), Adapting Gurvich-Karzanov-Khachiyan's algorithm for parity games: Implementation and experimentation, Technical Report 2001-020, Department of Information Technology, Uppsala University. `http://www.it.uu.se/research/reports/`.

Björklund, H., Petersson, V. & Vorobyov, S. (2001), Experiments with iterative improvement algorithms on completely unimodal hypercubes, Technical Report 2001-017, Information Technology/Uppsala University. `http://www.its.uu.se/research/reports/`.

Browne, A., Clarke, E. M., Jha, S., Long, D. E. & Marrero, W. (1997), 'An improved algorithm for the evaluation of fixpoint expressions', *Theor. Comput. Sci.* **178**, 237–255. Preliminary version in CAV'94, LNCS'818.

Condon, A. (1992), 'The complexity of stochastic games', *Information and Computation* **96**, 203–224.

Condon, A. (1993), 'On algorithms for simple stochastic games', *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **13**, 51–71.

Emerson, E. A. (1990), Temporal and modal logic, *in* J. van Leeuwen, ed., 'Handbook of Computer Science', Vol. B, Elsevier, pp. 996–1072.

Emerson, E. A. (1997), Model checking and the Mu-calculus, *in* N. Immerman & P. G. Kolaitis, eds, 'DIMACS Series in Discrete Mathematics', Vol. 31, pp. 185–214.

Gurvich, V. A., Karzanov, A. V. & Khachiyan, L. G. (1988), 'Cyclic games and an algorithm to find minimax cycle means in directed graphs', *U.S.S.R. Computational Mathematics and Mathematical Physics* **28**(5), 85–91.

Hirsch, M. D., Papadimitriou, C. H. & Vavasis, S. A. (1989), 'Exponential lower bounds for finding Brower fixed points', *Journal of Complexity* **5**, 379–416.

Hoffman, A. J. & Karp, R. M. (1966), 'On nonterminating stochastic games', *Management Science* **12**(5), 359–370.

Jurdziński, M. (1998), 'Deciding the winner in parity games is in UP$\cap$co-UP', *Information Processing Letters* **68**, 119–124.

Jurdzinski, M. (2000), Small progress measures for solving parity games, *in* '17th STACS', Vol. 1770 of *Lect. Notes Comput. Sci.*, Springer-Verlag, pp. 290–301.

Papadimitriou, C. (2001), Algorithms, games, and the internet, *in* 'ACM Annual Symposium on Theory of Computing', ACM, pp. 749–753.

Petersson, V. & Vorobyov, S. (2001*a*), Interior-point approach to parity games, *in* 'IEEE Annual Symposium on Logic in Computer Science (LICS'2001)', Boston, Massachusetts. Short communication.

Petersson, V. & Vorobyov, S. (2001*b*), 'A randomized subexponential algorithm for parity games', *Nordic Journal of Computing* **8**, 324–345. Preliminary version: Technical Report 2001-001, Department of Information Technology, Uppsala Unitersity. Also: Proceedings of the 12th Nordic Workshop on Programming Theory, Bergen , Norway, November 2000.

Puri, A. (1995), Theory of hybrid systems and discrete events systems, PhD thesis, EECS Univ. Berkeley.

Seidl, H. (1996), 'Fast and simple nested fixpoints', *Information Processing Letters* **59**(3), 303–308.

Vöge, J. & Jurdzinski, M. (2000), A discrete strategy improvement algorithm for solving parity games, *in* 'CAV'00: Computer-Aided Verification', Vol. 1855 of *Lect. Notes Comput. Sci.*, Springer-Verlag, pp. 202–215.

Zwick, U. & Paterson, M. (1996), 'The complexity of mean payoff games on graphs', *Theor. Comput. Sci.* **158**, 343–359.

Recent technical reports from the Department of Information Technology

2001-018	Emad Abd-Elrady: *An adaptive grid point RPEM algorithm for harmonic signal modeling*
2001-019	Wendy Kress and Jonas Nilsson: *Boundary conditions and estimates for the linearized Navier-Stokes equations on staggered grids*
2001-020	Emmanuel Beffara and Sergei Vorobyov: *Adapting Gurvich-Karzanov-Khachiyan's Algorithm for Parity Games: Implementation and Experimentation*
2001-021	Inger Boivie: *Usability and Design Decisions in Software Development*
2001-022	Pierre Flener, Alan Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel, Justin Pearson and Toby Walsh: *Symmetry in Matrix Models*
2001-023	Pierre Flener, Alan Frisch, Brahim Hnich, Zeynep Kiziltan, Ian Miguel and Toby Walsh: *Matrix Modelling*
2001-024	Larisa Beilina, Klas Samuelsson and Krister Åhlander: *A hybrid method for the wave equation*
2001-025	Emmanuel Beffara and Sergei Vorobyov: *Is Randomized Gurvich-Karzanov-Khachiyan's Algorithm for Parity Games Polynomial?*
2001-026	Jan Gulliksen and Inger Boivie: *Usability Throughout the Entire Software Development Lifecycle - A Summary of the INTERACT 2001 Workshop*
2001-027	K. Mahata, T. Söderström, M. Mossberg, L. Hillström and S. Mousavi: *On the use of flexural wave propagation experiments for identification of complex modulus*
2001-028	Magnus Berggren: *Wireless communication in telemedicine using Bluetooth and IEEE 802.11b*
2001-029	Henrik Lundgren, David Lundberg, Johan Nielsen, Erik Nordström and Christian Tschudin: *A Large-scale Testbed for Reproducible Ad hoc Protocol Evaluations*
2001-030	Jakob Engblom, Andreas Ermedahl and Friedhelm Stappert: *Validating a Worst-Case Execution Time Analysis Method for an Embedded Processor*
2002-001	Sônia M. Gomes and Bertil Gustafsson: *Combining Wavelets with Finite Differences: Consistency Analysis*
2002-002	S. Seipel and M. Lindkvist: *Methods and application of interactive 3D computer graphics in anthropology*
2002-003	Ken Mattsson, Magnus Svärd and Jan Nordström: *Stable and accurate artificial dissipation*
2002-004	Mats Löfberg, Jan Gulliksen and Torsten Norlander: *Användare i systemutvecklingsprocessen - ett fenomenologiskt perspektiv*
2002-005	Kajsa Ljungberg, Sverker Holmgren and Örjan Carlborg: *Efficient Kernel Algorithms for QTL Mapping Problems*
2002-007	F. Edelvik, G. Ledfelt, P. Lötstedt and D. J. Riley: *An Unconditionally Stable Subcell Model for Arbitrarily Oriented Thin Wires in the FETD Method*
2002-008	Henrik Björklund and Sergei Vorobyov: *Two Adversary Lower Bounds for Parity Games*

UPPSALA
UNIVERSITET

February 2002
ISSN 1404-3203
http://www.it.uu.se/