Randomized Strategy Improvement Algorithm for Parity Games

Sergei Vorobyov (vorobyov@csd.uu.se)
Computing Science Department, Box 337,
Uppsala University, 751 05 Uppsala, SWEDEN

1 Parity Games

Parity Games are infinite two-person full-information antagonistic games played on finite bipartite directed graphs without sinks. Vertices of a game graph are assigned natural indices (colors) from 0 to k. Bob and Alice alternate moves starting from the initial vertex. Bob (resp. Alice) wins if he (resp. she) can guarantee that in any infinite play the maximal vertex index hit infinitely often is even (resp. odd), independently of how clever the opponent is. It is known that the winner always has a positional (stationary, memoryless) winning strategy, i.e., a selection of exactly one outgoing edge from each his/her vertex. The problem of determining a winner in a parity games is equivalent, wrt. polynomial-time reductions, to the propositional μ-calculus model checking, (Emerson, Jutla & Sistla 1993).

Both problems are known to belong to the complexity class $\mathrm{NP} \cap \mathrm{coNP}$ (Emerson et al. 1993), but despite all efforts and conjectures it is still unknown whether the problem is in PTIME. The best currently known algorithms run in time $O(k^2 \cdot (n+1)^{\lfloor k/2 \rfloor + 1})$ (Browne, Clarke, Jha, Long & Marrero 1997), i.e., exponential in the number of indices k labeling vertices (equivalently, in the number of μ-ν alternations in a μ-calculus formula).

The aim of this short communication is to describe an implementation and computer experiments with the randomized stepwise improvement strategy, which converges quite fast (typically in a few iterations) to the winning strategy for one of the players.

2 Randomized Stepwise Improvement Strategy

We start by choosing a random strategy σ_0 for Bob and then iterate, by selecting a random counter-strategy (if any) σ_i for Alice when i is odd and for Bob when i is even, which wins against σ_{i-1}, wherever possible (choices in vertices from which one of the players definitely has a winning strategy are not altered). Such a random counter-strategy is computed by a modification of Karp's algorithm (polynomial) for finding maximum weighted cycles in strongly connected components when the strategy of one of the players is fixed. Roughly, for a randomly chosen SSC C reachable from the initial vertex, random vertex $a \in C$ of index j (with j even iff i is even), and its random successor b, we check whether there exists a path from b to a with maximum index $\leq j$. When constructing such paths we also randomize, and pick random edges involved in a winning cycle from a to a. Randomized counter-strategies are needed in order to avoid infinite loops repeating the same strategy (possible with a small probability) and foil 'bad' inputs.

Although we currently do not provide full explanation of the good practical behavior of the randomized algorithm sketched above, we can show that under certain assumptions, e.g., when the number of colors is quite large, $k = \Omega(\sqrt{n})$, it outperforms the best previously known algorithms for the problem. In fact, it is *subexponential*, in contrast with exponential previous algorithms, and uses a low-degree polynomial space.

The program is implemented in Java 2 SDK v1.2.2, runs under MS Windows'98 on Dell Latitude notebook with 466MHz Intel Celeron processor and 256MB of memory. The program is tested on randomly generated parity games (uniform distribution of successors, each vertex gets an index uniformly distributed in $[0..k]$). For randomly generated 25 games on 50 vertices with indices from 0 to 10 the number of iterations to obtain a winning strategy and a winner are as follows: 149, 54, 3, 20, 13, 56, 30, 19, 17, 16, 55, 67, 1, 17, 30, 39, 3, 51, 44, 9, 6, 51, 21, 10, 72. By the time of the conference we plan to provide more substantial experimental data and theoretical justification of the algorithm.

References

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.

Emerson, E. A., Jutla, C. & Sistla, A. P. (1993), On model-checking for fragments of μ-calculus, *in* 'Computer Aided Verification, Proc. 5th Int. Conference', Vol. 697, Lect. Notes Comput. Sci., pp. 385–396.