Synthesising Strategy Improvement and Recursive Algorithms for Solving 2.5 Player Parity Games

Ernst Moritz Hahn[1], Sven Schewe[2], Andrea Turrini[1], and Lijun Zhang[1]

[1] State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China
[2] University of Liverpool, Liverpool, UK

Abstract. 2.5 player parity games combine the challenges posed by 2.5 player reachability games and the qualitative analysis of parity games. These two types of problems are best approached with different types of algorithms: strategy improvement algorithms for 2.5 player reachability games and recursive algorithms for the qualitative analysis of parity games. We present a method that—in contrast to existing techniques—tackles both aspects with the best suited approach and works exclusively on the 2.5 player game itself. The resulting technique is powerful enough to handle games with several million states.

1 Introduction

Parity games are non-terminating zero sum games between two players, Player 0 and Player 1. The players move a token along the edges of a finite graph without sinks. The vertices are *coloured*, i.e. labelled with a priority taken from the set of natural numbers. The infinite sequence of vertices visited by the token is called the run of a graph, and each run is coloured according to the minimum priority that appears infinitely often on the run. A run is winning for a player if the parity of its colour agrees with the parity of the player.

Parity games come in two flavours: games with random moves, also called 2.5 player games, and games without random moves, called 2 player games. For 2 player games, the adversarial objectives of the two players are to ensure that the lowest priority that occurs infinitely often is even (for Player 0) and odd (for Player 1), respectively. For 2.5 player games, the adversarial objectives of the two players are to maximise the likelihood that the lowest priority that occurs infinitely often is even resp. odd.

Solving parity games is the central and most expensive step in many model checking [1,16,20,34,48], satisfiability checking [34,43,46,48], and synthesis [39,44] methods. As a result, efficient algorithms for 2 player parity games have been studied intensively [3–5,19,21,22,31,32,34,36–38,40–42,45,47,49].

Parity games with 2.5 players have recently attracted attention [6–10,17,18,26,50]. This attention, however, does not mean that results are similarly rich or similarly diverse as for 2 player games. Results on the existence of pure strategies and on approximation algorithms [17,50] are decades younger than similar results for 2 player games, while algorithmic solutions [7,9] focus on strategy improvement techniques only.

The qualitative counterpart of 2.5 player games, where one of the players has the goal to win almost surely while the other one wants to win with a non-zero chance, can be reduced to 2 player parity games, cf. [12] or attacked directly on the 2.5 player game with recursive algorithm [30]. The more interesting quantitative analysis can be approached through a reduction to 2.5 player reachability games [2], which can then be attacked with strategy improvement algorithms [14, 22, 36, 40, 45]. Alternatively, entangled strategy improvement algorithms can also run concurrently the 2.5 player parity game directly (for the quantitative aspects) and on a reduction to 2 player parity games (for the qualitative aspects) [7, 9]. (Or, likewise, run on the larger game with an ordered quality measure that gives preference to the likelihood to win and uses the progress measure from [4] or [47] as a tie-breaker.)

This raises the question if strategy improvement techniques can be directly applied on 2.5 player parity games, especially as such games are memoryless determined and therefore satisfy a main prerequisite for the use of strategy improvement algorithms. The short answer is that strategy algorithms for 2.5 player parity games simply do not work. Classical strategy improvement algorithms follow a joint pattern. They start with an arbitrary strategy f for one of the players (say Player 0). This strategy f maps each vertex of Player 0 to a successor, and thus resolves all moves of Player 0. This strategy is then *improved* by changing the strategy f at positions, where it is *profitable* to do so. The following steps are applied repeatedly until there is no improvement in Step 2.

1. Evaluate the simpler game resulting from fixing f.
2. Identify all changes to f that, when applied once[1], lead to an improvement.
3. Obtain a new strategy f' from f by selecting some subset of these changes.

So where does this approach go wrong? The first step works fine. After fixing a strategy for Player 0, we obtain a 1.5 player parity game, which can be solved efficiently with standard techniques [15].

It is also not problematic to identify the profitable switches in the second step. The winning probability for the respective successor vertex provides a natural measure for the profitability of a switch. We will show in Section 5 that, as usual for strategy improvement, any combination of such profitable switches will lead to an improvement.

The problem arises with the optimality guarantees. Strategy improvement algorithms guarantee that a strategy that cannot be improved is optimal. In the next paragraph, we will see an example, where this is not the case. Moreover, we will see that it can be necessary to change several decisions in a strategy f in order to obtain an improvement, something which is against the principles of strategy improvement.

1.1 An illustrating example

Consider the example 2.5 player parity game $\mathcal{P}_e$ depicted in Figure 1. Square vertices are controlled by Player 0, while triangular ones are controlled by Player 1. In circular vertices, a random successor vertex is chosen with the given probability. In v_w, Player 0 wins with certainty (and therefore in particular almost surely), while she loses with certainty in v_l. In $v_{0.55}$ (or $v_{0.95}$), Player 0 wins with probability 0.55 (or 0.95). For

[1] In classic strategy improvement algorithms, the restriction to implementing a change once is made to keep it easy to identify the improvements. Such changes will also lead to an improvement when applied repeatedly.

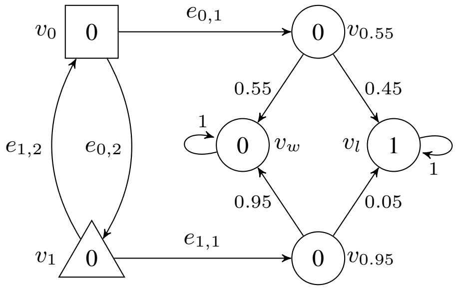

Fig. 1. A probabilistic parity game $\mathcal{P}_e$.

the nodes v_0 and v_1, we can see that the mutually *optimal strategy* for Player 0 and Player 1 are to play $e_{0,2}$ and $e_{1,1}$, respectively. Player 0 therefore wins with probability 0.95 when the game starts in v_0 and both players play optimally.

1.2 Naive strategy iteration

Strategy iteration algorithms start with an arbitrary strategy, and use an *update rule* to get profitable switches. These are edges, where the new target vertex has a higher probability of reaching the winning region (when applied once) compared to the current vertex. As usual with strategy improvement, any combination of profitable switches leads to a strictly better strategy for Player 0. We illustrate that, if done naively, it may lead to values that are only locally maximal. Assume that initially Player 0 chooses the edge $e_{0,1}$ from v_0, then the best counter strategy of Player 1 is to choose $e_{1,2}$ from v_1. The winning probability for Player 0 under these strategies is 0.55.

In strategy iteration, an update rule allows a player to switch actions only if the switching offers some *improvement*. Since by switching to the edge $e_{0,2}$ Player 0 would obtain the same winning probability, no strategy iteration can be applied, and the algorithm terminates with a sub-optimal solution.

Let us try to get some insights from this problem. Observe that Player 1 can entrap the play in the left vertices v_0 and v_1 when Player 0 chooses the edge $e_{0,2}$, such that the almost sure winning region of Player 0 cannot be reached. However, this comes to the cost of losing almost surely for Player 1, as the dominating colour on the resulting run is 0. Broadly speaking, Player 0 must find a strategy that maximises her chance of reaching her almost sure winning regions, but only under the constraint that the counter strategy of Player 1 does not introduce new almost sure winning regions for Player 0.

1.3 Solutions from the literature

In the literature, two different solutions to this problem have been discussed. Neither of these solutions works fully on the game graph of the 2.5 player parity game. Instead, one of them uses a reduction to reachability games through a simple gadget construction [2], while the other uses strategy improvement on two levels, for the qualitative update described above, and for an update within subgames of states that have the same value [7, 9]; this requires to keep a pair of entangled strategies.

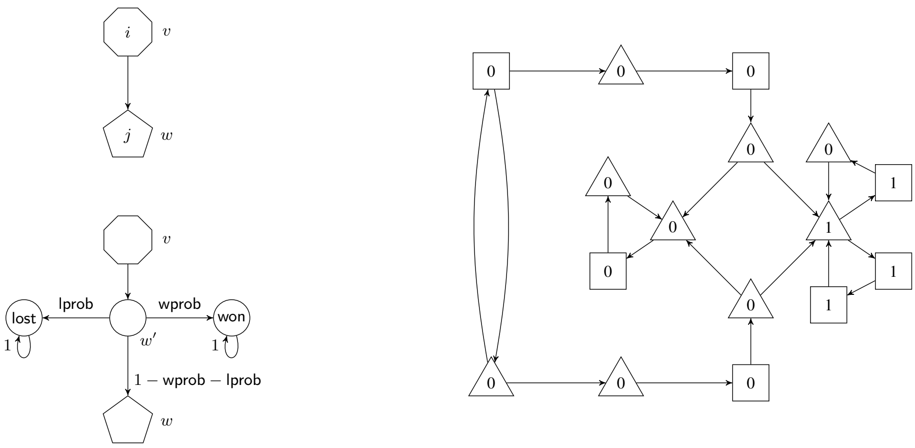

Fig. 2. Left: the gadget construction from [12] (Figure 2a). Right: The qualitative game resulting from the game from Figure 1 when using the gadget construction from [12] (Figure 2b).

Gadget construction for a reduction to reachability games. In [2], it is shown that 2.5 player parity games can be solved by reducing them to 2.5 player reachability games and solving them, e.g. by using a strategy improvement approach. For this reduction, one can use the simple gadgets shown in Figure 2a. There, when a vertex is passed by, the token goes to an accepting sink with probability wprob and to a losing sink with probability lprob, both depending on the priority of the node (and continues otherwise as in the parity game). For accordingly chosen $\mathsf{wprob}, \mathsf{lprob}$, any optimal strategy for this game is an optimal strategy for the parity game. To get this guarantee, however, the termination probabilities have to be very small indeed. In [2], they are constructed from the expression $(n!^2 2^{2n+3} M^{2n^2})^{-1}$ where n is the number of vertices and M is an integer depending on the probabilities occurring in the model. Unfortunately, these small probabilities render this approach very inefficient and introduce numerical instability.

Classic strategy improvement for 2.5 player parity games. In [7, 9], the concept of strategy improvement algorithms has been extended to 2.5 player parity games. To overcome the problem that the natural quality measure—the likelihood of winning—is not fine enough, this approach constructs classical 2 player games played on translations of the value classes (the set of vertices with the same likelihood of winning). These subgames are translated using a gadget construction similar to the one used for qualitative solutions for 2.5 player to a solution to 2 player games from [12] (Figure 2a). This results in the 2 player game shown in the right part of Figure 2 (Figure 2b).

The strategy improvement algorithm keeps track of 'witnesses $\omega = (\pi, \overline{\pi}_Q)$', which consists of a strategy π on the 2.5 player parity game, and a strategy $\overline{\pi}_Q$ defined on the 2 player game Q obtained from this 2.5 player game using the gadget construction from [12]. The strategies are entangled in that π is the translation[2] of $\overline{\pi}_Q$. That is, the strategies have to concur on the nodes of Player 0 from the 2.5 player game, and each

[2] In the notation of [7, 9], $\pi = \mathsf{Tr}_{\text{almost}}(\overline{\pi}_Q)$.

update on π (resp. $\overline{\pi}_Q$) on the decisions from these vertices will translate to an update on the strategy of $\overline{\pi}_Q$ (resp. π) on the same vertices.

The valuation of one of these vertices is an ordered pair, consisting of the chance of obtaining the parity objective as the primary measure, and the value obtained in the quantitative game restricted to the individual value classes (vertices with the same chance of obtaining the parity objective) as a secondary measure [7, 9].

1.4 Novel strategy iteration algorithm

We show that we can apply strategy improvement techniques with two different update rules directly on the 2.5 player game. The first rule is a standard update rule for increasing the chance of reaching the almost sure winning region. As we have seen in the example, this rule would not necessarily find the optimum: it would not find the improvement from edge $e_{0,1}$ to $e_{0,2}$. To overcome this problem, we introduce a second rule that handles the problem that Player 1 can reduce the chances of reaching the almost sure winning region of Player 0 by playing a strategy that leads to a larger almost sure winning region for Player 0. This step uses a reduction to the *qualitative* evaluation of these games. Player 0 changes her strategy in a way that she would win on the subgame that consists only of the edges of Player 0 and Player 1 that are *neutral*. For both players, these are the edges that lead to successor states with the same chance of winning under the current strategy. If this provides a larger almost sure winning region for Player 0 than f, then update f in this *new winning region* accordingly leads to a strictly better strategy f'.

While the first rule alone is not powerful enough, the two rules together provide the guarantee that a strategy that cannot be improved by either of them is optimal.

Note that the second rule is a non-standard rule for strategy improvement. Not only does it not rely on an improvement that is obtained when a change is applied once, it also requires to apply a fixed set of changes (in the new region) in one step for correctness. This is quite unusual for strategy improvement algorithms, where the combination of updates selected is irrelevant for correctness.

A further significant difference to the method from [7, 9] is that we do not have to revert to solving transformed games. Instead, we use the new generalisation of McNaughton's algorithm to the qualitative solution of 2.5 player parity games [30]. This method seems to maintain the good practical performance known for classic recursive techniques, which have proven to be much faster than strategy improvement for the qualitative analysis of parity games [25]. A consequence of this choice is that we solve the qualitative games completely when there is no progress through the naive update step, which reduces the number of times that qualitative updates have to be considered.

This way, we use strategy improvement for the quantitative part of the analysis, where it has its strengths, while leaning on a variation [30] of McNaughton's algorithm [21, 37, 49] for the qualitative part of the analysis, where prior research suggests that recursive algorithms outperform strategy improvement [25].

Note that our quality measure strategy improvement is the same as the primary measure used in classical strategy improvement for 2.5 player parity games [7, 9]. Different from that approach, we do not need to resort to gadget constructions for progressing within value classes, but can overcome the lack of progress w.r.t. the primary measure through invoking a performant algorithm for solving 2.5 player games quantitatively [30].

1.5 Organisation of the Paper

We first introduce the standard terms and concepts in Section 2. We then recall the strategy improvement algorithms in Section 3, describe our algorithm in Section 4, show its correctness in Section 5, and offer an experimental evaluation in Section 6.

2 Terms and Concepts

A *probability distribution* over a finite set A is a function $\mu\colon A \to [0,1] \cap \mathbb{Q}$ with $\sum_{a\in A} \mu(a) = 1$. We denote by $Distr(A)$ the set of probability distributions over A.

Definition 1. *An* arena *is a tuple* $\mathfrak{A} = (V_0, V_1, V_r, E, \mathsf{prob})$*, where*

- V_0*,* V_1*, and* V_r *are three finite disjoint sets of* vertices *owned by the three players: Player* 0*, Player* 1*, and Player* random*, respectively. Let* $V \overset{\text{def}}{=} V_0 \cup V_1 \cup V_r$*;*
- $E \subseteq V \times V$ *is a set of* edges *such that* (V, E) *is a* sinkless *directed graph, i.e. for each* $v \in V$ *there exists* $v' \in V$ *such that* $(v, v') \in E$*; for* $\sigma \in \{0, 1, r\}$ *we let* $E_\sigma \overset{\text{def}}{=} E \cap (V_\sigma \times V)$*.*
- $\mathsf{prob}\colon V_r \to Distr(V)$ *is the* successor distribution function*. We require that for each* $v \in V_r$ *and each* $v' \in V$*,* $\mathsf{prob}(v)(v') > 0$ *if and only if* $(v, v') \in E$*.*

If $V_0 = \emptyset$ or $V_1 = \emptyset$, we call $\mathfrak{A}$ a *Markov decision process* (MDP) or 1.5 player game. If both $V_0 = V_1 = \emptyset$, we call $\mathfrak{A}$ a *Markov chain* (MC). Given an arena $\mathfrak{A} = (V_0, V_1, V_r, E, \mathsf{prob})$, we define the following concepts.

- A *play* is an infinite sequence $\pi = v_0v_1v_2v_3\ldots$ such that $(v_i, v_{i+1}) \in E$ for all $i \in \mathbb{N}$. We define $\pi(i) \overset{\text{def}}{=} v_i$. We denote by $\mathsf{Play}(\mathfrak{A})$ the set of all plays of $\mathfrak{A}$.
- For $\sigma \in \{0, 1\}$, a (pure memoryless) *strategy* f_σ of Player σ is a mapping $f_\sigma\colon V_\sigma \to V$ from the vertices V_σ of Player σ to their successor states, i.e. for each $v \in V_\sigma$, $(v, f_\sigma(v)) \in E$. We denote the set of Player 0 and 1 strategies by Strats_0 and Strats_1, respectively.
- Given a strategy f_0 for Player 0, we define the *induced MDP* as $\mathfrak{A}_{f_0} = (\emptyset, V_1, V_r \cup V_0, E_{f_0}, \mathsf{prob}_{f_0})$ with $E_{f_0} \overset{\text{def}}{=} (E \setminus V_0 \times V) \cup \{\, (v, f_0(v)) \mid v \in V_0 \,\}$ and

$$\mathsf{prob}_{f_0}(v)(v') \overset{\text{def}}{=} \begin{cases} \mathsf{prob}(v)(v') & \text{if } v{\in}V_r, \\ 1 & \text{if } v{\in}V_0 \text{ and } v'{=}f_0(v), \\ 0 & \text{otherwise,} \end{cases}$$

  and similarly for Player 1.
- Given strategies f_0, f_1 for Player 0 and Player 1, respectively, we denote by $\mathfrak{A}_{f_0,f_1} \overset{\text{def}}{=} (\mathfrak{A}_{f_0})_{f_1}$ the *induced MC* of the strategies.

– If $\mathfrak{A}$ is a MC, we denote by $\mathbf{P}^{\mathfrak{A}}(v)\colon \Sigma^{\mathfrak{A}} \to [0,1]$ the uniquely induced [33] probability measure on $\Sigma^{\mathfrak{A}}$, the σ-algebra on the *cylinder sets* of the plays of $\mathfrak{A}$, under the condition that the initial node is v, where, for a finite prefix $\pi' = v_0 v_1 \dots v_n$ of a play π, the probability of the cylinder set $C_{\pi'}$ of π' is defined as $\mathbf{P}^{\mathfrak{A}}(v)(C_{\pi'}) = \prod_{i=0}^{n-1} \mathsf{prob}(v_i)(v_{i+1})$ if $v_0 = v$, 0 otherwise. For a generic arena $\mathfrak{A}$ and strategies f_0 and f_1 of Player 0 and Player 1, respectively, we let $\mathbf{P}^{\mathfrak{A}}_{f_0,f_1}(v) \stackrel{\text{def}}{=} \mathbf{P}^{\mathfrak{A}_{f_0,f_1}}(v)$.

Definition 2. *A* 2.5 player game*, also referred to as* Markov game *(MG), is a tuple* $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{win})$*, where* $\mathfrak{A} = (V_0, V_1, V_r, E, \mathsf{prob})$ *is an arena and* $\mathsf{win} \subseteq \mathsf{Play}(\mathfrak{A})$ *is the* winning condition *for Player* 0*, the set of plays for which Player* 0 *wins.*

The notions of plays, strategies, induced 1.5 player games, etc. extend to 2.5 player games by considering their underlying arena.

We consider two types of winning conditions, reachability and parity objectives.

Definition 3. *A 2.5 player* reachability game *is a 2.5 player game* $\mathcal{P}$ *in which the winning condition* win *is defined by a* target set $\mathsf{R} \subseteq V$*. Then, we have* $\mathsf{win} = \{\, \pi \in \mathsf{Play}(\mathcal{P}) \mid \exists i \geq 0 : \pi(i) \in \mathsf{R} \,\}$*. For 2.5 player reachability games, we also use the notation* $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{R})$.

Definition 4. *A* 2.5 player parity game *(MPG) is a 2.5 player game* $\mathcal{P}$ *in which the winning condition* win *is defined by the* priority function $\mathsf{pri}\colon V \to \mathbb{N}$ *mapping each vertex to a natural number. We call the image of* pri *the set of* priorities *(or:* colours*), denoted by* $\mathcal{C}$*. Note that, since* V *is finite,* $\mathcal{C}$ *is finite as well. We extend* pri *to plays, using* $\mathsf{pri}\colon \pi \mapsto \liminf_{i \to \infty} \mathsf{pri}(\pi(i))$*. Then, we have* $\mathsf{win} = \{\, \pi \in \mathsf{Play}(\mathcal{P}) \mid \mathsf{pri}(\pi)$ *is even* $\}$*. For 2.5 player parity games, we also use the notation* $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{pri})$*. We denote with* $|\mathcal{P}|$ *the size of a 2.5 player parity game, referring to the space its overall representation takes.*

Note that in the above discussion we have defined strategies as mappings from vertices of the respective player to successor vertices. More general definitions of strategies exist that e.g. use randomised choices (imposing a probability distributions over the edges chosen) or take the complete history of the game so far into account. However, it is known that, for finite 2.5 player parity and reachability games, the simple pure memoryless strategies we have introduced above suffice to obtain mutually optimal infima and suprema [12].

We also use the common intersection and subtraction operations on directed graphs for arenas and games: given an MG $\mathcal{P}$ with arena $\mathfrak{A} = (V_0, V_1, V_r, E, \mathsf{prob})$,

– $\mathcal{P} \cap V'$ denotes the MG $\mathcal{P}'$ we obtain when we restrict the arena $\mathfrak{A}$ to $\mathfrak{A} \cap V' \stackrel{\text{def}}{=} (V_0 \cap V', V_1 \cap V', V_r \cap V', E \cap (V' \times V'), \mathsf{prob}{\restriction_{V' \cap V_r}})$,
– for $E' \supseteq E_r$, we denote by $\mathcal{P} \cap E'$ the MG $\mathcal{P}'$ we obtain when restricting arena $\mathfrak{A}$ to $\mathfrak{A} \cap E' \stackrel{\text{def}}{=} (V_0, V_1, V_r, E \cap E', \mathsf{prob})$.

Note that the result of such an intersection may or may not be substochastic or contain sinks. While we use these operations freely in intermediate constructions, we make sure that, whenever they are treated as games, they have no sinks and are not substochastic.

Definition 5. *Let $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{win})$ be a 2.5 player game, and let f_0 and f_1 be two strategies for player 0 and 1, respectively. The* value $\mathsf{val}^{\mathcal{P}}_{f_0,f_1} : V \to [0,1]$ *is defined as*

$$\mathsf{val}^{\mathcal{P}}_{f_0,f_1}(v) \stackrel{\text{def}}{=} \mathbf{P}^{\mathcal{P}}_{f_0,f_1}(v)(\{\, \pi \in \mathsf{Play}(\mathcal{P}) \mid \pi \in \mathsf{win} \,\}).$$

We also define

$$\begin{aligned} \mathsf{val}^{\mathcal{P}}_{f_0}(v) &\stackrel{\text{def}}{=} \inf_{f_1' \in \mathsf{Strats}_1} \mathsf{val}^{\mathcal{P}}_{f_0,f_1'}(v), \\ \mathsf{val}^{\mathcal{P}}_{f_1}(v) &\stackrel{\text{def}}{=} \sup_{f_0' \in \mathsf{Strats}_0} \mathsf{val}^{\mathcal{P}}_{f_0',f_1}(v), \\ \mathsf{val}^{\mathcal{P}}(v) &\stackrel{\text{def}}{=} \sup_{f_0' \in \mathsf{Strats}_0} \inf_{f_1' \in \mathsf{Strats}_1} \mathsf{val}^{\mathcal{P}}_{f_0',f_1'}(v). \end{aligned}$$

We write $\mathsf{val}^{\mathcal{P}}_{f'} \geq \mathsf{val}^{\mathcal{P}}_{f}$ if, for all $v \in V$, $\mathsf{val}^{\mathcal{P}}_{f'}(v) \geq \mathsf{val}^{\mathcal{P}}_{f}(v)$ holds, and $\mathsf{val}^{\mathcal{P}}_{f'} > \mathsf{val}^{\mathcal{P}}_{f}$ if $\mathsf{val}^{\mathcal{P}}_{f'} \geq \mathsf{val}^{\mathcal{P}}_{f}$ and $\mathsf{val}^{\mathcal{P}}_{f'} \neq \mathsf{val}^{\mathcal{P}}_{f}$ hold.

Definition 6. *Given a vertex $v \in V$, a strategy f_σ for Player σ is called v-*winning *if, starting from v, Player σ wins almost surely in the MDP defined by f_σ (that is, $\mathsf{val}^{\mathcal{P}}_{f_\sigma}(v) = 1-\sigma$). For $\sigma \in \{0,1\}$, a vertex v in V is v-winning for Player σ if Player σ has a v-winning strategy f_σ. We call the set of v-winning vertices for Player σ the* winning region *of Player σ, denoted W_σ. Note for $v \in W_0$, $\mathsf{val}^{\mathcal{P}}(v) = 1$, whereas for $v \in W_1$ we have $\mathsf{val}^{\mathcal{P}}(v) = 0$.*

3 Strategy Improvement

A strategy improvement algorithm takes a memoryless strategy f of one player, in our case of Player 0, and either infers that the strategy is optimal, or offers a family $\mathcal{I}_f$ of strategies, such that, for all strategies $f' \in \mathcal{I}_f$, $\mathsf{val}^{\mathcal{P}}_{f'} > \mathsf{val}^{\mathcal{P}}_{f}$ holds.

The family $\mathcal{I}_f$ is usually given through *profitable switches*. In such a case, $\mathcal{I}_f$ is defined as follows.

Definition 7. *Given a 2.5 player game $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{win})$ and a strategy f for Player 0, the* profitable switches, *denoted $\mathsf{profit}(\mathcal{P}, f)$, for Player 0 are the edges that offer a strictly higher chance of succeeding (under the given strategy). That is, $\mathsf{profit}(\mathcal{P}, f) = \{\, (v,v') \in E_0 \mid \mathsf{val}^{\mathcal{P}}_{f}(v') > \mathsf{val}^{\mathcal{P}}_{f}(v) \,\}$. We also define the unprofitable switches accordingly as $\mathsf{loss}(\mathcal{P}, f) = \{\, (v,v') \in E_0 \mid \mathsf{val}^{\mathcal{P}}_{f}(v') < \mathsf{val}^{\mathcal{P}}_{f}(v) \,\}$.*

$\mathcal{I}_f$ is the set of strategies that can be obtained from f by applying one or more profitable switches to f: $\mathcal{I}_f = \{\, f' \in \mathsf{Strats}_0 \mid f' \neq f$ and $\forall v \in V_0 : f'(v) = f(v)$ or $(v, f'(v)) \in \mathsf{profit}(\mathcal{P}, f) \,\}$.

Strategy improvement methods can usually start with an arbitrary strategy f_0, which is then updated by selecting some $f_{i+1} \in \mathcal{I}_{f_i}$ until $\mathcal{I}_{f_i}$ is eventually empty. This f_i is then guaranteed to be optimal. The update policy with which the profitable switch or switches are selected is not relevant for the correctness of the method, although it does impact on the performance and complexity of the algorithms. In our implementation, we use a 'greedy switch all' update policy, that is we perform any switch we can perform and change the strategy to the locally optimal switch.

For 2.5 player reachability games, strategy improvement algorithms provide optimal strategies.

Theorem 1 (cf. [14]). *For a 2.5 player reachability game $\mathcal{P}$, a strategy improvement algorithm with the profitable switches / improved strategies as defined in Definition 7 terminates with an optimal strategy for Player* 0*.*

In the strategy improvement step, for all $v \in V$ and all $f' \in \mathcal{I}_f$, it holds that $\mathsf{val}^{\mathcal{P}}_{f'}(v) = \mathsf{val}^{\mathcal{P}}_{f'}\big(f'(v)\big) \geq \mathsf{val}^{\mathcal{P}}_{f}\big(f(v)\big) = \mathsf{val}^{\mathcal{P}}_{f}(v)$. Moreover, strict inequality is obtained at some vertex in V. As we have seen in the introduction, this is not the case for 2.5 player parity games: in the example from Figure 1, for a strategy f with $f(v_0) = v_{0.55}$, the switch from edge $e_{0,1}$ to $e_{0,2}$ is not profitable. Note, however, that it is not unprofitable either.

4 Algorithm

We observe that situations where the naive strategy improvement algorithm described in the previous section gets stuck are *tableaux*: no profitable switches are available. However, switches that are *neutral* in that applying them once would neither lead to an increased nor to a decreased likelihood of winning can still lead to an improvement, and can even happen that combinations of such neutral switches are required to obtain an improvement. As usual with strategy improvement algorithms, neutral switches cannot generally be added to the profitable switches: not only would one lose the guarantee to improve, one can also reduce the likelihood of winning when applying such changes.

Overcoming this problem is the main reason why strategy improvement techniques for MPG would currently have to use a reduction to 2.5 player reachability games (or other reductions), with the disadvantages discussed in the introduction. We treat these tableaux directly and avoid reductions. We first make formal what neutral edges are.

Definition 8. *Given a 2.5 player game $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{win})$ and a strategy f for Player* 0*, we define the set of* neutral edge $\mathsf{neutral}(\mathcal{P}, f)$ *as follows:*

$$\mathsf{neutral}(\mathcal{P}, f) \stackrel{\text{def}}{=} E_r \cup \{\, (v, v') \in E_0 \cup E_1 \mid \mathsf{val}^{\mathcal{P}}_{f}(v') = \mathsf{val}^{\mathcal{P}}_{f}(v) \,\}.$$

Based on these neutral edges, we define an update policy on the subgame played only on the neutral edges. The underlying idea is that, when Player 0 can win in case the game only uses neutral edges, then Player 1 will have to try to break out. He can only do this by changing his decision from one of his states in a way that is profitable for Player 0.

Definition 9. *Given a 2.5 player game $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{win})$ and a strategy f for Player* 0*, we define the* neutral subgame *of $\mathcal{P}$ for f as $\mathcal{P}' = \mathcal{P} \cap \mathsf{neutral}(\mathcal{P}, f)$. Based on $\mathcal{P}'$ we define the set $\mathcal{I}'_f$ of* additional strategy improvements *as follows.*

Let W_0 and W_0' be the winning regions of Player 0 on $\mathcal{P}$ and $\mathcal{P}'$, respectively. If $W_0 = W_0'$, then $\mathcal{I}_f' = \emptyset$. Otherwise, let $\mathcal{W}$ be the set of strategies that are v-winning for Player 0 on $\mathcal{P}'$ for all vertices $v \in W_0'$. Then we set

$$\mathcal{I}_f'' = \left\{ f_0 \in \mathsf{Strats}_0 \;\middle|\; \begin{array}{r} \exists f_w \in \mathcal{W} : \forall v \in W_0' : f_0(v) = f_w(v) \\ \textit{and } \forall v \notin W_0' : f_0(v) = f(v) \end{array} \right\},$$
$$\mathcal{I}_f' = \{ f' \in \mathcal{I}_f'' \mid \forall v \in W_0 : f'(v) = f(v) \}.$$

We remark that $W_0 \subseteq W_0'$ always holds. Intuitively, we apply a qualitative analysis on the neutral subgame, and if the winning region of Player 0 on the neutral subgame is larger than her winning region on the full game, then we use the new winning strategy on the new part of the winning region. Intuitively, this forces Player 1 to leave this area eventually (or to lose almost surely). As he cannot do this through neutral edges, the new strategy for Player 0 is superior over the old one.

Example 1. Consider again the example MPG $\mathcal{P}_e$ from Figure 1 and the strategy such that $f_0(v_0) = v_{0.55}$. Under this strategy, $\mathsf{neutral}(\mathcal{P}_e, f_0) = E_r \cup \{(v_0, v_{0.55}), (v_0, v_1), (v_1, v_0)\}$; the resulting neutral subgame $\mathcal{P}_e'$ is the same as $\mathcal{P}_e$ except for the edge $e_{1,1}$. In $\mathcal{P}_e'$, the winning region W_0' is $W_0' = \{v_0, v_1, v_w\}$, while the original region was $W_0 = \{v_w\}$. The two sets $\mathcal{I}_{f_0}'$ and $\mathcal{I}_{f_0}''$ contain only the strategy f_0' such that $f_0'(v_0) = v_1$. In order to avoid to lose almost surely in W_0', Player 1 has to change his strategy from $f_1(v_1) = v_0$ to $f_1'(v_1) = v_{0.95}$ in $\mathcal{P}_e$. Consequently, strategy f_0' is superior to f_0: the resulting winning probability is not 0.55 but 0.95 for v_0 and v_1.

Note that using $\mathcal{I}_f'$ or $\mathcal{I}_f''$ in the strategy iteration has the same effect. Once a run has reached W_0 in the neutral subgame, it cannot leave it. Thus, changing the strategy f_0 from $\mathcal{I}_f''$ to a strategy f' with $f'(v) = f(v)$ for $v \in W_0$ and $f'(v) = f_0(v)$ for $v \notin W_0$ will not change the chance of winning: $\mathsf{val}_{f_0}^{\mathcal{P}'} = \mathsf{val}_{f'}^{\mathcal{P}'}$ and $\mathsf{val}_{f_0}^{\mathcal{P}} = \mathsf{val}_{f'}^{\mathcal{P}}$. This also implies $\mathcal{I}_f'' \neq \emptyset \Rightarrow \mathcal{I}_f' \neq \emptyset$, since $\mathcal{I}_f'$ contains all strategies that belong to $\mathcal{I}_f''$ and that agree with f only on the original winning region W_0. Using $\mathcal{I}_f'$ simplifies the proof of Lemma 1, but it also emphasises that one does not need to re-calculate the strategy on a region that is already winning.

Our extended strategy improvement algorithm applies updates from either of these constructions until no further improvement is possible. That is, we can start with an arbitrary Player 0 strategy f_0 and then apply $f_{i+1} \in \mathcal{I}_{f_i} \cup \mathcal{I}_{f_i}'$ until $\mathcal{I}_{f_i} = \mathcal{I}_{f_i}' = \emptyset$. We will show that therefore f_i is an optimal Player 0 strategy.

For the algorithm, we need to calculate $\mathcal{I}_{f_i}$ and $\mathcal{I}_{f_i}'$. Calculating $\mathcal{I}_{f_i}$ requires only to solve 1.5 player parity games [15], and we use IscasMC [27,28] to do so. Calculating $\mathcal{I}_{f_i}'$ requires only qualitative solutions of neutral subgame $\mathcal{P}'$. For this, we apply the algorithm from [30].

A more algorithmic representation of our algorithm with a number of minor design decisions is provided in the arXiv version [29] of this paper. The main design decision is to favour improvements from $\mathcal{I}_{f_i}$ over those from $\mathcal{I}_{f_i}'$. This allows for calculating $\mathcal{I}_{f_i}'$ only if $\mathcal{I}_{f_i}$ is empty. Starting with calculating $\mathcal{I}_{f_i}$ first is a design decision, which is slightly arbitrary. We have made it because solving 1.5 player games quantitatively is cheaper than solving 2.5 player games qualitatively and we believe that the guidance for the search is, in practice, better in case of quantitative results. Likewise, we have implemented a 'greedy switch all' improvement strategy, simply because this is believed to behave well in practice. We have, however, not collected evidence for either decision and acknowledge that finding a good update policy is an interesting future research.

5 Correctness

5.1 Correctness proof in a nutshell

The correctness proof combines two arguments: the correctness of *all* basic strategy improvement algorithms for reachability games and a reduction from 2.5 player parity games to 2.5 player reachability games with arbitrarily close winning probabilities for similar strategy pairs. In a nutshell, if we approximate close enough, then three properties hold for a game $\mathcal{P}$ and a strategy f of Player 0:

1. all 'normal' strategy improvements of the parity game correspond to strategy improvements in the reachability game (Corollary 2);
2. if Player 0 has a larger winning region W_0' in the neutral subgame (cf. Definition 9) for $P \cap \mathsf{neutral}(\mathcal{P}, f)$ than for $\mathcal{P}_f$, then replacing f by a winning strategy in $\mathcal{I}_f'$ leads to an improved strategy in the reachability game (Lemma 1); and
3. if neither of these two types of strategy improvements are left, then a strategy improvement step on the related 2.5 player reachability game will not lead to a change in the winning probability on the 2.5 player parity game (Lemma 2).

5.2 Two game transformations

In this subsection we discuss two game transformations that change the likelihood of winning only marginally and preserve the probability of winning, respectively. The first transformation turns 2.5 player parity games into 2.5 player reachability games such that a strategy that is an optimal strategy for the reachability game is also optimal for the parity game (cf. [2]).

Definition 10. *Let* $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{pri})$*,* $\varepsilon \in (0,1)$*, and* $n \in \mathbb{N}$*. We define the 2.5 player reachability game* $\mathcal{P}_{\varepsilon,n} = (V_0, V_1, V_r'', E'', \mathsf{prob}', \{\mathsf{won}\})$ *with*

- $V_r'' = V_r \cup V' \cup \{\mathsf{won}, \mathsf{lost}\}$*, where (i)* V' *contains primed copies of the vertices; for ease of notation, the copy of a vertex* v *is referred to as* v' *in this construction; (ii)* won *and* lost *are fresh vertices; they are a winning and a losing sink, respectively;*
- $E' = \{\, (v, w') \mid (v, w) \in E \,\} \cup \{(\mathsf{won}, \mathsf{won}), (\mathsf{lost}, \mathsf{lost})\}$*;*
- $E'' = E' \cup \{\, (v', v) \mid v \in V \,\} \cup \{\, (v', \mathsf{won}) \mid v \in V \,\} \cup \{\, (v', \mathsf{lost}) \mid v \in V \,\}$*;*
- $\mathsf{prob}'(v)(w') = \mathsf{prob}(v)(w)$ *for all* $v \in V_r$ *and* $(v, w) \in E$*;*
- $\mathsf{prob}'(v')(\mathsf{won}) = \mathsf{wprob}\big(\varepsilon, n, \mathsf{pri}(v)\big)$*,*
- $\mathsf{prob}'(v')(\mathsf{lost}) = \mathsf{lprob}\big(\varepsilon, n, \mathsf{pri}(v)\big)$*,*
- $\mathsf{prob}'(v')(v) = 1 - \mathsf{wprob}\big(\varepsilon, n, \mathsf{pri}(v)\big) - \mathsf{lprob}\big(\varepsilon, n, \mathsf{pri}(v)\big)$ *for all* $v \in V$*, and*
- $\mathsf{prob}'(\mathsf{won})(\mathsf{won}) = \mathsf{prob}'(\mathsf{lost})(\mathsf{lost}) = 1.$

where $\mathsf{lprob}, \mathsf{wprob} \colon (0,1) \times \mathbb{N} \times \mathbb{N} \to [0,1]$ *are two functions with* $\mathsf{lprob}(\varepsilon, n, c) + \mathsf{wprob}(\varepsilon, n, c) \leq 1$ *for all* $\varepsilon \in (0,1)$ *and* $n, c \in \mathbb{N}$.

Intuitively, this translation replaces all the vertices by the gadgets from Figure 2a.

Note that $\mathcal{P}_{\varepsilon,n}$ and $\mathcal{P}$ have similar memoryless strategies. By a slight abuse of the term, we say that a strategy f_σ of Player σ on $\mathcal{P}_{\varepsilon,n}$ is *similar* to her strategy f'_σ on $\mathcal{P}$ if $f'_\sigma\colon v \mapsto f_\sigma(v)'$ holds, i.e. when v is mapped to w by f_σ, then v is mapped to w' by f'_σ.

Theorem 2 (cf. [2]). *Let $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{pri})$ be a 2.5 player parity game. Then, there exist $\varepsilon \in (0,1)$ and $n \geq |\mathcal{P}|$ such that we can construct $\mathcal{P}_{\varepsilon,n}$ and the following holds: for all strategies $f_0 \in \mathsf{Strats}_0$, $f_1 \in \mathsf{Strats}_1$, and all vertices $v \in V$, it holds that $\left|\mathsf{val}^{\mathcal{P}}_{f_0,f_1}(v) - \mathsf{val}^{\mathcal{P}_{\varepsilon,n}}_{f'_0,f'_1}(v)\right| < \varepsilon$, $\left|\mathsf{val}^{\mathcal{P}}_{f_0,f_1}(v) - \mathsf{val}^{\mathcal{P}_{\varepsilon,n}}_{f'_0,f'_1}(v')\right| < \varepsilon$, $\left|\mathsf{val}^{\mathcal{P}}_{f_0}(v) - \mathsf{val}^{\mathcal{P}_{\varepsilon,n}}_{f'_0}(v)\right| < \varepsilon$, $\left|\mathsf{val}^{\mathcal{P}}_{f_0}(v) - \mathsf{val}^{\mathcal{P}_{\varepsilon,n}}_{f'_0}(v')\right| < \varepsilon$, $\left|\mathsf{val}^{\mathcal{P}}_{f_1}(v) - \mathsf{val}^{\mathcal{P}_{\varepsilon,n}}_{f'_1}(v)\right| < \varepsilon$, and $\left|\mathsf{val}^{\mathcal{P}}_{f_1}(v) - \mathsf{val}^{\mathcal{P}_{\varepsilon,n}}_{f'_1}(v')\right| < \varepsilon$, where f'_0 resp. f'_1 are similar to f_0 resp. f_1.*

The results of [2] are stronger in that they show that the probabilities grow sufficiently slow for the reduction to be polynomial, but we use this construction only for correctness proofs and do not apply it in our algorithms. For this reason, existence is enough for our purpose. As [2] does not contain a theorem that directly makes the statement above, we have included a simple construction (without tractability claim) with a correctness proof in the arXiv version [29] of this paper.

We will now introduce a second transformation that allows us to consider changes in the strategies in many vertices at the same time.

Definition 11. *Let $\mathcal{P} = (V_0, V_1, V_r, E, \mathsf{prob}, \mathsf{win})$ and a region $R \subseteq V$. Let $\mathcal{F}_R = \{\, f\colon R \cap V_0 \to V \mid \forall v \in R : \big(v, f(v)\big) \in E \,\}$ denote the set of memoryless strategies for Player 0 restricted to R. The transformation results in a parity game $\mathcal{P}^R = (V'_0, V'_1, V'_r, E', \mathsf{prob}', \mathsf{pri}')$ such that*

- *$V''_0 = V_0 \cup R$, $V'''_0 = (V_0 \cap R) \times \mathcal{F}_R$, and $V'_0 = V''_0 \cup V'''_0$;*
- *$V''_1 = V_1 \setminus R$, $V'''_1 = (V_1 \cap R) \times \mathcal{F}_R$, and $V'_1 = V''_1 \cup V'''_1$;*
- *$V''_r = V_r \setminus R$, $V'''_r = (V_r \cap R) \times \mathcal{F}_R$, and $V'_r = V''_r \cup V'''_r$;*
- *$E' = \{\, (v,w) \in E \mid v \in V \setminus R \,\} \cup \{\, (v,(v,f)) \mid v \in R \text{ and } f \in \mathcal{F}_R \,\} \cup \{\, ((v,f),(w,f)) \mid v,w \in R,\ (v,w) \in E \text{ and either } v \notin V_0 \text{ or } f(v) = w \,\} \cup \{\, ((v,f),w) \mid v \in R,\ w \notin R,\ (v,w) \in E \text{ and either } v \notin V_0 \text{ or } f(v) = w \,\}$;*
- *$\mathsf{prob}'(v)(w) = \mathsf{prob}(v)(w)$, $\mathsf{prob}'\big((v,f)\big)(w) = \mathsf{prob}(v)(w)$, and $\mathsf{prob}'\big((v,f)\big)\big((w,f)\big) = \mathsf{prob}(v)(w)$; and*
- *$\mathsf{pri}'(v) = \mathsf{pri}(v)$ for all $v \in V$ and $\mathsf{pri}'\big((v,f)\big) = \mathsf{pri}(v)$ otherwise.*

Intuitively, the transformation changes the game so that, every time R is entered, Player 0 has to fix her memoryless strategy in the game. The fact that in the resulting game the strategy f for Player 0 is fixed entering R is due to the jump from the original vertex v to (v,f) whenever $v \in R$. Once in R, either the part v of (v,f) is under the control of Player 1 or Player random, i.e. $v \notin V_0$, so it behaves as in $\mathcal{P}$, or the next state w (or (w,f) if $w \in R$) is the outcome of f, i.e. $w = f(v)$.

It is quite obvious that this transformation does not impact on the likelihood of winning. In fact, Player 0 can simulate every memoryless strategy $f : V_0 \to V$ by playing a strategy $f_R : V'_0 \to V'$ that copies f outside of R (i.e. for each $v \in V_0 \setminus R$, $f_R(v) = f(v)$) and moves to the $f \restriction_R$ (i.e. f with a preimage restricted to R) copy from states in R (i.e. for each $v \in V_0 \cap R$, $f_R(v) = (v, f \restriction_R)$): there is a one-to-one correspondence between playing in $\mathcal{P}$ with strategy f and playing in $\mathcal{P}^R$ with strategy f_R when starting in V.

Theorem 3. *For all $v \in V$, all $R \subseteq V$, and all memoryless Player 0 strategies f, $\mathsf{val}_f^{\mathcal{P}}(v) = \mathsf{val}_{f_R}^{\mathcal{P}^R}\big((v, f \restriction_R)\big)$, $\mathsf{val}^{\mathcal{P}}(v) = \sup_{f \in \mathsf{Strats}_0(\mathcal{P})} \mathsf{val}^{\mathcal{P}^R}\big((v, f \restriction_R)\big)$, and $\mathsf{val}^{\mathcal{P}}(v) = \mathsf{val}^{\mathcal{P}^R}(v)$ hold.*

5.3 Correctness proof

For a given game $\mathcal{P}$, we call an $\varepsilon \in (0,1)$ *small* if it is at most $\frac{1}{5}$ of the smallest difference between all probabilities of winning that can occur on any strategy pair for any state in any game $\mathcal{P}^R$ for any $R \subseteq V$. For every small ε, we get the following corollary from Theorem 2.

Corollary 1 (preservation of profitable and unprofitable switches). *Let $n \geq |\mathcal{P}|$, f be a Player 0 strategy for $\mathcal{P}$, f' be the corresponding strategy for $\mathcal{P}_{\varepsilon,n}$, $\varepsilon \in (0,1)$ be small, $v \in V$, $w = f(v)$, and $(v,u) \in E$. Then $\mathsf{val}_f^{\mathcal{P}}(u) > \mathsf{val}_f^{\mathcal{P}}(w)$ implies $\mathsf{val}_{f'}^{\mathcal{P}_{\varepsilon,n}}(u) > \mathsf{val}_{f'}^{\mathcal{P}_{\varepsilon,n}}(w')$, and $\mathsf{val}_f^{\mathcal{P}}(u) < \mathsf{val}_f^{\mathcal{P}}(w)$ implies $\mathsf{val}_{f'}^{\mathcal{P}_{\varepsilon,n}}(u) < \mathsf{val}_{f'}^{\mathcal{P}_{\varepsilon,n}}(w')$.*

It immediately follows that all combinations of profitable switches can be applied, and will lead to an improved strategy: for small ε, a profitable switch for f_i from $f_i(v) = w$ to $f_{i+1}(v) = u$ implies $\mathsf{val}_{f_i}^{\mathcal{P}}(u) \geq \mathsf{val}_{f_i}^{\mathcal{P}}(w) + 5\varepsilon$ since by definition, we have that $\mathsf{val}_{f_i}^{\mathcal{P}}(u) > \mathsf{val}_{f_i}^{\mathcal{P}}(w)$ (as the switch is profitable); in particular, $\mathsf{val}_{f_i}^{\mathcal{P}}(u) = \mathsf{val}_{f_i}^{\mathcal{P}}(w) + \delta$ with $\delta \in \mathbb{R}^{>0}$; since $\varepsilon \leq \frac{1}{5}\delta$, we have that $\mathsf{val}_{f_i}^{\mathcal{P}}(u) \geq \mathsf{val}_{f_i}^{\mathcal{P}}(w) + 5\varepsilon$. The triangular inequalities provided by Theorem 2 imply that $\mathsf{val}_{f_i'}^{\mathcal{P}_{\varepsilon,n}}(u') \geq \mathsf{val}_{f_i'}^{\mathcal{P}_{\varepsilon,n}}(w') + 3\varepsilon$, since $\big|\mathsf{val}_{f_i}^{\mathcal{P}} - \mathsf{val}_{f_i'}^{\mathcal{P}_{\varepsilon,n}}\big| < \varepsilon$. Consequently, since under f_{i+1}' we have that $\mathsf{val}_{f_{i+1}'}^{\mathcal{P}_{\varepsilon,n}}(v') = \mathsf{val}_{f_i'}^{\mathcal{P}_{\varepsilon,n}}(u')$, it follows that $\mathsf{val}_{f_{i+1}'}^{\mathcal{P}_{\varepsilon,n}}(v) \geq \mathsf{val}_{f_i'}^{\mathcal{P}_{\varepsilon,n}}(v) + 3\varepsilon$, and, using triangulation again, we get $\mathsf{val}_{f_{i+1}}^{\mathcal{P}}(v) \geq \mathsf{val}_{f_i}^{\mathcal{P}}(v) + \varepsilon$. Thus, we have the following corollary:

Corollary 2. *Let $\mathcal{P}$ be a given 2.5 player parity game, and f_i be a strategy with profitable switches ($\mathsf{profit}(\mathcal{P}, f_i) \neq \emptyset$). Then, $\mathcal{I}_{f_i} \neq \emptyset$, and for all $f_{i+1} \in \mathcal{I}_{f_i}$, $\mathsf{val}_{f_{i+1}}^{\mathcal{P}} > \mathsf{val}_{f_i}^{\mathcal{P}}$.*

We now turn to the case that there are no profitable switches for f in the game $\mathcal{P}$. Corollary 1 shows that, for the corresponding strategy f' in $\mathcal{P}_{\varepsilon,n}$, all profitable switches lie within the neutral edges for f in $\mathcal{P}$, provided f has no profitable switches.

We expand the game by fixing the strategy of Player 0 for the vertices in $R \cap V_0$ for a region $R \subseteq V$. The region we are interested in is the winning region of Player 0 in the neutral subgame $\mathcal{P} \cap \mathsf{neutral}(P, f)$. The game is played as follows.

For every strategy $f_R \colon R \cap V_0 \to V$ such that $\big(r, f_R(r)\big) \in E$ holds for all $r \in R$, the game has a copy of the original game intersected with R, where the choices of Player 0 on the vertices in R are fixed to the single choice defined by the respective strategy f_R. We define $\|\mathcal{P}\| = \max\{\, |\mathcal{P}^R| \mid R \subseteq V \,\}$.

We consider the case where the almost sure winning region of Player 0 in the neutral subgame $\mathcal{P}' = \mathcal{P} \cap \mathsf{neutral}(\mathcal{P}, f_i)$ is strictly larger than her winning region in $\mathcal{P}_{f_i}$.

Lemma 1. *Let $\mathcal{P}$ be a given 2.5 player parity game, and f_i be a strategy such that the winning region W_0' for Player 0 in the neutral subgame $\mathcal{P}'=\mathcal{P}\cap\mathsf{neutral}(\mathcal{P}, f_i)$ is strictly larger than her winning region W_0 in $\mathcal{P}_{f_i}$. Then $\mathcal{I}'_{f_i}\neq\emptyset$ and, $\forall f_{i+1}\in\mathcal{I}'_{f_i}$, $\mathsf{val}^{\mathcal{P}}_{f_{i+1}}>\mathsf{val}^{\mathcal{P}}_{f_i}$.*

Proof. The argument is an extension of the common argument for strategy improvement made for the modified reachability game. We first recall that the strategies in $\mathcal{I}'_{f_i}$ differ from f_i only on the winning region W_0' of Player 0 in the neutral subgame $\mathcal{P}'$. Assume that we apply the change *once*: the first time W_0' is entered, we play the new strategy, and after it is left, we play the old strategy. If the reaction of Player 1 is to stay in W_0', Player 0 will win almost surely in $\mathcal{P}$. If he leaves it, the value is improved due to the fact that Player 1 has to take a disadvantageous edge to leave it.

Consider the game $\mathcal{P}^{W_0'}$ and fix $f_{i+1} \in \mathcal{I}'_{f_i}$. Using Theorem 3, this implies that, when first in a state $v \in W_0'$, Player 0 moves to (v, f_{i+1}) for some $f_{i+1} \in \mathcal{I}'_{f_i}$, then the likelihood of winning is either improved or 1 for any counter strategy of Player 1. For all $v \in W_0' \setminus W_0$, this implies a strict improvement. For an $n \geq \|\mathcal{P}\|$ and a small ε, we can now follow the same arguments as for the Corollaries 1 and 2 on $\mathcal{P}^{W_0'}$ to establish that $\mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_{i+1})_{W_0'}} > \mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_i)_{W_0'}}$ holds, where the inequality is obtained through the same steps: $\mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_i)_{W_0'}}\big((v, f_{i+1}|_{W_0})\big) > \mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_i)_{W_0'}}(v)$ implies $\mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_i)_{W_0'}}\big((v, f_{i+1}|_{W_0})\big) \geq \mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_i)_{W_0'}}(v) + 5\varepsilon$; this implies $\mathsf{val}^{\mathcal{P}^{W_0'}_{\varepsilon,n}}_{(f_i)_{W_0'}}\big((v, f_{i+1}|_{W_0})'\big) \geq \mathsf{val}^{\mathcal{P}^{W_0'}_{\varepsilon,n}}_{(f_i)_{W_0'}}(v) + 3\varepsilon$; and this implies $\mathsf{val}^{\mathcal{P}^{W_0'}_{\varepsilon,n}}_{(f_{i+1})_{W_0'}}(v) = \mathsf{val}^{\mathcal{P}^{W_0'}_{\varepsilon,n}}_{(f_{i+1})_{W_0'}}\big((v, f_{i+1}|_{W_0})'\big) \geq \mathsf{val}^{\mathcal{P}^{W_0'}_{\varepsilon,n}}_{(f_i)_{W_0'}}(v) + 3\varepsilon$ and we finally get $\mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_{i+1})_{W_0'}}(v) = \mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_{i+1})_{W_0'}}\big((v, f_{i+1}|_{W_0})\big) > \mathsf{val}^{\mathcal{P}^{W_0'}}_{(f_i)_{W_0'}}(v)$.

With Theorem 3, we obtain that $\mathsf{val}^{\mathcal{P}}_{f_{i+1}} > \mathsf{val}^{\mathcal{P}}_{f_i}$ holds. □

Let us finally consider the case where there are no profitable switches for Player 0 in $\mathcal{P}_{f_i}$ and her winning region on the neutral subgame $\mathcal{P} \cap \mathsf{neutral}(\mathcal{P}, f_i)$ coincides with her winning region in $\mathcal{P}_{f_i}$.

Lemma 2. *Let $\mathcal{P}$ be an MPG and f_i be a strategy such that the set of profitable switches is empty and the neutral subgame $\mathcal{P} \cap \mathsf{neutral}(\mathcal{P}, f_i)$ has the same winning region for Player 0 as her winning region in $\mathcal{P}_{f_i}$ ($\mathcal{I}_{f_i} = \mathcal{I}'_{f_i} = \emptyset$). Then, every individual profitable switch in the reachability game $\mathcal{P}_{\varepsilon,n}$ from f_i to f_{i+1} implies $\mathsf{val}^{\mathcal{P}}_{f_{i+1}} = \mathsf{val}^{\mathcal{P}}_{f_i}$ and $\mathsf{neutral}(\mathcal{P}, f_{i+1}) = \mathsf{neutral}(\mathcal{P}, f_i)$.*

Proof. When there are no profitable switches in the parity game $\mathcal{P}$ for f_i, then all profitable switches in the reachability game $\mathcal{P}_{\varepsilon,n}$ for f_i (if any) must be within the set of neutral edges $\mathsf{neutral}(\mathcal{P}, f_i)$ in the parity game $\mathcal{P}$. We apply one of these profitable switches at a time. By our definitions, this profitable switch is neutral in the 2.5 player parity game.

Taking this profitable (in the reachability game $\mathcal{P}_{\varepsilon,n}$ for a small ε and some $n \geq \|\mathcal{P}\|$) switch will improve the likelihood of winning for Player 0 in the reachability game. By our definition of ε, this implies that the likelihood of winning cannot be decreased on any position in the parity game.

To see that the quality of the resulting strategy cannot be higher for Player 0 in the 2.5 player parity game, recall that Player 1 can simply follow his optimal strategy on the neutral subgame. The likelihood of winning for Player 0 is the likelihood of reaching her winning region, and this winning region has not changed. Moreover, consider the evaluation of the likelihood of reaching this winning region: since by fixing the strategy for Player 1 the resulting game is an MDP, such an evaluation can be obtained by solving a linear programming problem (cf. the arXiv version [29] for more details). The old minimal non-negative solution to the resulting linear programming problem is a solution to the new linear programming problem, as it satisfies all constraints.

Putting these arguments together, likelihood of winning in the parity game is not altered in any vertex by this change. Hence, the set of neutral edges is not altered. $\square$

This lemma implies that *none* of the subsequently applied improvement steps applied on the 2.5 player reachability game has any effect on the quality of the resulting strategy on the 2.5 player parity game. Together, the above lemmas and corollaries therefore provide the correctness argument.

Theorem 4. *The algorithm is correct.*

Proof. Lemma 2 shows that, when $\mathcal{I}_{f_i}$ and $\mathcal{I}'_{f_i}$ are empty (i.e. when the algorithm terminates), then the updates in the related 2.5 player reachability game will henceforth (and thus until termination) not change the valuation for the 2.5 player parity game. With Theorems 1 and 2 and our selection of small ε, it follows that f_i is an optimal strategy. The earlier lemmas and corollaries in this subsection show that every strategy $f_{i+1} \in \mathcal{I}_{f_i} \cup \mathcal{I}'_{f_i}$ satisfies $\mathsf{val}^{\mathcal{P}}_{f_{i+1}} > \mathsf{val}^{\mathcal{P}}_{f_i}$. Thus, the algorithm produces strategies with strictly increasing quality in each step until it terminates. As the game is finite, then also the set of strategies is finite, thus the algorithm will terminate after finitely many improvement steps with an optimal strategy. $\square$

As usual with strategy improvement algorithms, we cannot provide good bounds on the number of iterations. As reachability games are a special case of 2.5 player games, all selection rules considered by Friedmann [23,24] will have exponential lower bounds.

6 Implementation and experimental results

We have written a prototypical implementation for the approach of this paper, which accepts as input models in the same format as the probabilistic model checker PRISM-GAMES [13], an extension of PRISM [35] to stochastic Markov games. As case study, we consider an extension of the robot battlefield presented in [30], consisting of $n \times n$ square tiles, surrounded by a solid wall; four marked zones $\mathrm{zone}_1, \dots, \mathrm{zone}_4$ at the corners, each of size 3×3; and two robots, R_0 and R_1, acting in strict alternation. Each tile can be occupied by at most one robot at a time. When it is the turn of a robot, this robot can move as follows: decide a direction and move one tile forward; decide a direction and attempt to move two tiles forward. In the latter case, the robot moves two tiles forward with 50% probability, but only one tile forward with 50% probability. If the robot would run into a wall or into the other robot, it stops at the tile

Table 1. Robots analysis: different reachability properties

property	n	b	MPG vertices	MPG colours	$p_{destr} = 0.1$ $p_{\max}$	$p_{destr} = 0.1$ t_{sol}	$p_{destr} = 0.3$ $p_{\max}$	$p_{destr} = 0.3$ t_{sol}	$p_{destr} = 0.5$ $p_{\max}$	$p_{destr} = 0.5$ t_{sol}	$p_{destr} = 0.7$ $p_{\max}$	$p_{destr} = 0.7$ t_{sol}	$p_{destr} = 0.9$ $p_{\max}$	$p_{destr} = 0.9$ t_{sol}
Reachability $\langle\langle R_0\rangle\rangle\mathcal{P}_{\max=?}[\ \mathbf{F}\mathrm{zone}_1 \wedge \mathbf{F}\mathrm{zone}_2 \wedge \mathbf{F}\mathrm{zone}_3 \wedge \mathbf{F}\mathrm{zone}_4]$	7	1	663 409	2	0.9614711	33	0.8178044	22	0.6247858	22	0.3961410	21	0.1384328	23
	7	2	1 090 537	2	0.9244309	56	0.6742610	66	0.4017138	57	0.1708971	58	0.0230085	52
	7	3	1 517 665	2	0.8926820	89	0.5793073	87	0.2995397	77	0.0953904	86	0.0060025	68
	7	4	1 944 793	2	0.8667039	112	0.5385632	109	0.2409219	96	0.0649772	107	0.0026513	85
	7	5	2 371 921	2	0.8571299	147	0.5062357	144	0.2167625	127	0.0506530	140	0.0019157	112
Ordered Reachability $\langle\langle R_0\rangle\rangle\mathcal{P}_{\max=?}[\mathbf{F}(\mathrm{zone}_1 \wedge \mathbf{F}\,\mathrm{zone}_2)]$	8	1	528 168	2	0.9613511	23	0.8176058	19	0.6246643	21	0.3962011	20	0.1384974	19
	8	2	868 986	2	0.9243652	35	0.6999023	44	0.4522051	35	0.2083732	42	0.0320509	40
	8	3	1 209 804	2	0.9091132	62	0.6538475	71	0.3643938	56	0.1352710	60	0.0131408	58
	8	4	1 550 622	2	0.9013742	91	0.6200998	91	0.3316778	72	0.1168758	74	0.0097312	71
	8	5	1 891 440	2	0.8977303	113	0.6031945	108	0.3207408	90	0.1138603	88	0.0093679	83
Reach-Avoid $\langle\langle R_0\rangle\rangle\mathcal{P}_{\max=?}[\ \neg\mathrm{zone}_1\, \mathbf{U}\, \mathrm{zone}_2 \wedge \neg\mathrm{zone}_4\, \mathbf{U}\, \mathrm{zone}_2 \wedge \mathbf{F}\mathrm{zone}_3]$	9	1	833 245	4	0.9447793	46	0.8005413	31	0.6125397	35	0.3914531	25	0.1372075	24
	9	2	1 370 827	4	0.9095579	81	0.6824329	52	0.4411181	61	0.2089446	49	0.0302023	45
	9	3	1 908 409	4	0.8972146	108	0.6375883	68	0.3792906	84	0.1444959	71	0.0106721	66
	9	4	2 445 991	4	0.8936231	148	0.6221536	93	0.3478172	117	0.1158094	103	0.0051508	89
	9	5	2 983 573	4	0.8918034	172	0.6162166	109	0.3366050	136	0.1010400	120	0.0035468	105
Reachability $\langle\langle R_0\rangle\rangle\mathcal{P}_{\max=?}[\ \mathbf{F}\mathrm{zone}_1 \wedge \mathbf{F}\mathrm{zone}_2 \wedge \mathbf{F}\mathrm{zone}_3 \wedge \mathbf{F}\mathrm{zone}_4]$	10	1	3 307 249	2	0.9614711	186	0.8178044	141	0.6247858	142	0.3961410	142	0.1384328	141
	10	2	5 440 429	2	0.9244267	296	0.6755372	414	0.4017718	374	0.1665626	732	0.0207851	615
	10	3	7 573 609	2	0.8931881	570	0.5742127	572	0.2864117	509	0.0847474	1019	0.0043153	861
	10	4	9 706 789	2	0.8676441	530	0.5239018	794	0.2248369	735	0.0479367	1396	0.0009959	1610
	10	5	11 839 969	2	0.8503684	968	0.4885654	980	0.1866995	971	0.0305890	1708	—TO—	

before the obstacle. Robot R_1 can also shoot R_0 instead of moving, which is destroyed with probability p_{destr}^d where p_{destr} is the probability of destroying the robot and d is the Euclidean distance between the two robots. Once destroyed, R_0 cannot move any more. We assume that we are in control of R_0 but cannot control the behaviour of R_1. Our goal is to maximise, under any possible behaviour of R_1, the probability of fulfilling a certain objective depending on the zones, such as repeatedly visiting all zones infinitely often, visiting the zones in a specific order, performing such visits without entering other zones in the meanwhile, and so on. As an example, we can specify that the robot eventually reaches each zone by means of the probabilistic LTL (PLTL) formula $\langle\langle R_0\rangle\rangle\mathcal{P}_{\max=?}[\bigwedge_{i=1,\dots,4} \mathbf{F}\,\mathrm{zone}_i]$ requiring to maximise the probability of satisfying $\bigwedge_{i=1,\dots,4} \mathbf{F}\,\mathrm{zone}_i$ by controlling R_0 only.

The machine we used for the experiments is a 3.6 GHz Intel Core i7-4790 with 16GB 1600 MHz DDR3 RAM of which 12GB assigned to the tool; the timeout has been set to 30 minutes. We have applied our tool on a number of properties that require the robot R_0 to visit the different zones in a certain order. In Table 1 we report the performance measurements for these properties. Column "property" shows the PLTL formula we consider, column "n" the width of the battlefield instance, and column "b" the number of bullets R_1 can shoot. For the "MPG" part, we present the number of "vertices" of the resulting MPG and the number of "colours". In the remaining columns, for each value of "p_{destr}", we report the achieved maximum probability "$p_{\max}$" and the time "t_{sol}" in seconds needed to solve the game. Note that we cannot compare to PRISM-GAMES because it does not support general PLTL formulas, and we are not aware of other tools to compare with.

As we can see, the algorithm performs quite well on MPGs with few million states. It is worth mentioning that a large share of the time spent is due to the evaluation of the 1.5 player parity games in the construction of the profitable switches. For instance, such an evaluation required 137 seconds out of 172 for the case $n = 9$, $b = 5$, and $p_{destr} = 0.1$. Since a large part of these 1.5 player games are similar, we are investigating how to avoid the repeated evaluation of similar parts to reduce the running time. Generally, all improvements in the quantitative solution of 1.5 player parity games and the qualitative solution of 2.5 player parity games will reduce the running time of our algorithm.

7 Discussion

We have combined a recursive algorithm for the quantitative solution of 2.5 player parity games with a strategy improvement algorithm, which lifts these results to the qualitative solution of 2.5 player parity games. This shift is motivated by the significant acceleration in the qualitative solution of 2.5 player parity games: while [11] scaled to a few thousand vertices, [30] scales to tens of millions of states. This changes the playing field and makes qualitative synthesis a realistic target. It also raises the question if this technique can be incorporated smoothly into a quantitative solver.

Previous approaches [7, 9] have focused on developing a progress measure that allows for joining the two objective. This has been achieved in studying strategy improvement techniques that give preference to the likelihood of winning, and overcome stalling by performing strategy improvement on the larger qualitative game from [12] on the value classes.

This approach was reasonable at the time, where the updates benefited from memorising the recently successful strategies on the qualitative game. Moreover, focussing on value classes keeps the part of the qualitative game under consideration small, which is a reasonable approach when the cost of qualitative strategy improvement is considered significant. Building on a fast solver for the qualitative analysis, we can afford to progress in larger steps.

The main advancement, however, is as simple as it is effective. We use strategy improvement where it has a simple direct meaning (the likelihood to win), and we do not use it where the progress measure is indirect (progress measure within a value class). This has allowed us to transfer the recent performance gains from qualitative solutions of 2.5 player parity games [30] to their quantitative solution.

The difference in performance also explains the difference in the approach regarding complexity. Just as the deterministic subexponential complexity of solving 2.5 player games qualitatively is not very relevant in [30] (as this approach would be very slow in practice), the expected subexponential complexity in [9] is bought by exploiting a random facet method, which implies that only one edge is updated in every step. From a theoretical angle, these complexity considerations are interesting. From a practical angle, however, strategy improvement algorithms that use multiple switches in every step are usually faster and therefore preferable.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (Grants No. 61472473, 61532019, 61550110249, 61550110506), by the National 973 Program (No. 2014CB340701), the CDZ project CAP (GZ 1023), the CAS Fellowship for International Young Scientists, the CAS/SAFEA International Partnership Program for Creative Research Teams, and the EPSRC grant EP/M027287/1.

References

1. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *JACM*, 49(5):672–713, 2002.
2. Daniel Andersson and Peter Bro Miltersen. The complexity of solving stochastic games on graphs. In *ISAAC*, volume 5878 of *LNCS*, pages 112–121, 2009.
3. Dietmar Berwanger, Anuj Dawar, Paul Hunter, and Stephan Kreutzer. DAG-width and parity games. In *STACS*, pages 524–436, 2006.
4. Henrik Björklund and Sergei Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. *DAM*, 155(2):210–229, 2007.
5. Anca Browne, Edmund M. Clarke, Somesh Jha, David E. Long, and Wilfredo R. Marrero. An improved algorithm for the evaluation of fixpoint expressions. *TCS*, 178(1–2):237–255, 1997.
6. Krishnendu Chatterjee. The complexity of stochastic müller games. *Inf. Comput.*, 211:29–48, 2012.
7. Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. The complexity of quantitative concurrent parity games. In *SODA*, pages 678–687. SIAM, 2006.
8. Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. Strategy improvement for concurrent reachability and turn-based stochastic safety games. *J. Comput. Syst. Sci.*, 79(5):640–657, 2013.
9. Krishnendu Chatterjee and Thomas A. Henzinger. Strategy improvement and randomized subexponential algorithms for stochastic parity games. In *Proc. of STACS*, volume 3884 of *Lecture Notes in Computer Science*, pages 512–523. Springer, 2006.
10. Krishnendu Chatterjee and Thomas A. Henzinger. Strategy improvement for stochastic rabin and streett games. In Christel Baier and Holger Hermanns, editors, *CONCUR 2006 - Concurrency Theory, 17th International Conference, CONCUR 2006, Bonn, Germany, August 27-30, 2006, Proceedings*, volume 4137 of *Lecture Notes in Computer Science*, pages 375–389. Springer, 2006.
11. Krishnendu Chatterjee, Thomas A. Henzinger, Barbara Jobstmann, and Arjun Radhakrishna. Gist: A solver for probabilistic games. In *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, volume 6174 of *Lecture Notes in Computer Science*, pages 665–669. Springer, 2010.
12. Krishnendu Chatterjee, Marcin Jurdziński, and Thomas A. Henzinger. Quantitative stochastic parity games. In *SODA'04*, pages 121–130, 2004.
13. Taolue Chen, Vojtech Forejt, Marta Z. Kwiatkowska, David Parker, and Aistis Simaitis. PRISM-games: A model checker for stochastic multi-player games. In *TACAS*, volume 7795 of *LNCS*, pages 185–191, 2013.
14. Anne Condon. On algorithms for simple stochastic games. In *Advances in Computational Complexity Theory*, pages 51–73, 1993.
15. Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.
16. Luca de Alfaro, Thomas A. Henzinger, and Rupak Majumdar. From verification to control: Dynamic programs for omega-regular objectives. In *LICS*, pages 279–290, 2001.
17. Luca de Alfaro and Rupak Majumdar. Quantitative solution of omega-regular games. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 675–683. ACM, 2001.
18. Luca de Alfaro and Rupak Majumdar. Quantitative solution of omega-regular games. *J. Comput. Syst. Sci.*, 68(2):374–397, 2004.

19. E. Allen Emerson and Charanjit S. Jutla. Tree automata, μ-calculus and determinacy. In *FOCS*, pages 368–377, 1991.
20. E. Allen Emerson, Charanjit S. Jutla, and A. Prasad Sistla. On model-checking for fragments of μ-calculus. In *CAV*, volume 697 of *LNCS*, pages 385–396, 1993.
21. E. Allen Emerson and Chin-Laung Lei. Efficient model checking in fragments of the propositional μ-calculus. In *LICS*, pages 267–278, 1986.
22. John Fearnley. Non-oblivious strategy improvement. In *LPAR*, pages 212–230, 2010.
23. Oliver Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *LICS*, pages 145–156, 2009.
24. Oliver Friedmann, Thomas Dueholm Hansen, and Uri Zwick. A subexponential lower bound for the random facet algorithm for parity games. In *SODA*, pages 202–216, 2011.
25. Oliver Friedmann and Martin Lange. Solving parity games in practice. In *ATVA*, volume 5799 of *LNCS*, pages 182–196, 2009.
26. Hugo Gimbert and Florian Horn. Solving simple stochastic games with few random vertices. *LMCS*, 5(2), 2009.
27. Ernst Moritz Hahn, Guangyuan Li, Sven Schewe, Andrea Turrini, and Lijun Zhang. Lazy probabilistic model checking without determinisation. In *CONCUR*, volume 42 of *LIPIcs*, pages 354–367, 2015.
28. Ernst Moritz Hahn, Yi Li, Sven Schewe, Andrea Turrini, and Lijun Zhang. IscasMC: A web-based probabilistic model checker. In *FM*, volume 8442 of *LNCS*, pages 312–317, 2014.
29. Ernst Moritz Hahn, Sven Schewe, Andrea Turrini, and Lijun Zhang. Synthesising strategy improvement and recursive algorithms for solving 2.5 player parity games. arxiv:1607.01474.
30. Ernst Moritz Hahn, Sven Schewe, Andrea Turrini, and Lijun Zhang. A simple algorithm for solving qualitative probabilistic parity games. In *CAV*, volume 9780 of *LNCS*, pages 291–311, 2016.
31. Marcin Jurdziński. Small progress measures for solving parity games. In *STACS*, volume 1770 of *LNCS*, pages 290–301, 2000.
32. Marcin Jurdziński, Mike Paterson, and Uri Zwick. A deterministic subexponential algorithm for solving parity games. *SIAM Journal on Computing*, 38(4):1519–1532, 2008.
33. John G. Kemeny, J. Laurie Snell, and Anthony W. Knapp. *Denumerable Markov Chains*. D. Van Nostrand Company, 1966.
34. Dexter Kozen. Results on the propositional μ-calculus. *TCS*, 27:333–354, 1983.
35. Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV*, volume 6806 of *LNCS*, pages 585–591, 2011.
36. Walter Ludwig. A subexponential randomized algorithm for the simple stochastic game problem. *Inf. Comput.*, 117(1):151–155, 1995.
37. Robert McNaughton. Infinite games played on finite graphs. *Ann. Pure Appl. Logic*, 65(2):149–184, 1993.
38. Jan Obdržálek. Fast mu-calculus model checking when tree-width is bounded. In *CAV*, volume 2725 of *LNCS*, pages 80–92, 2003.
39. Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *Journal of Logical Methods in Computer Science*, 3(3:5), 2007.
40. Anuj Puri. *Theory of hybrid systems and discrete event systems*. PhD thesis, Computer Science Department, University of California, Berkeley, 1995.
41. Sven Schewe. An optimal strategy improvement algorithm for solving parity and payoff games. In *CSL*, volume 5213 of *LNCS*, pages 368–383, 2008.
42. Sven Schewe. Solving parity games in big steps. *Journal of Computer and System Sciences*, 84:243–262, 2017.
43. Sven Schewe and Bernd Finkbeiner. Satisfiability and finite model property for the alternating-time μ-calculus. In *CSL*, volume 4207 of *LNCS*, pages 591–605, 2006.

44. Sven Schewe and Bernd Finkbeiner. Synthesis of asynchronous systems. In *LOPSTR*, volume 4407 of *LNCS*, pages 127–142, 2006.
45. Sven Schewe, Ashutosh Trivedi, and Thomas Varghese. Symmetric strategy improvement. In *ICALP*, volume 9135 of *LNCS*, pages 388–400, 2015.
46. Moshe Y. Vardi. Reasoning about the past with two-way automata. In *ICALP*, volume 1443 of *LNCS*, pages 628–641, 1998.
47. Jens Vöge and Marcin Jurdziński. A discrete strategy improvement algorithm for solving parity games (Extended abstract). In *CAV*, volume 1855 of *LNCS*, pages 202–215, 2000.
48. Thomas Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bull. Soc. Math. Belg.*, 8(2), 2001.
49. Wieslaw Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *TCS*, 200(1-2):135–183, 1998.
50. Wieslaw Zielonka. Perfect-information stochastic parity games. In *FOSSACS*, volume 2987 of *LNCS*, pages 499–513, 2004.