Smoothed Analysis of Deterministic Discounted and Mean-Payoff Games

Bruno Loff
LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal

Mateusz Skomra
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

Abstract

We devise a policy-iteration algorithm for deterministic two-player discounted and mean-payoff games, that runs in polynomial time with high probability, on any input where each payoff is chosen independently from a sufficiently random distribution and the underlying graph of the game is ergodic.

This includes the case where an arbitrary set of payoffs has been perturbed by a Gaussian, showing for the first time that deterministic two-player games can be solved efficiently, in the sense of smoothed analysis.

More generally, we devise a *condition number* for deterministic discounted and mean-payoff games played on ergodic graphs, and show that our algorithm runs in time polynomial in this condition number.

Our result confirms a previous conjecture of Boros et al., which was claimed as a theorem [18] and later retracted [19]. It stands in contrast with a recent counter-example by Christ and Yannakakis [24], showing that Howard's policy-iteration algorithm does *not* run in smoothed polynomial time on *stochastic* single-player mean-payoff games.

Our approach is inspired by the analysis of random optimal assignment instances by Frieze and Sorkin [39], and the analysis of bias-induced policies for mean-payoff games by Akian, Gaubert and Hochart [6].

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases Mean-payoff games, discounted games, policy iteration, smoothed analysis

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.147

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: `https://arxiv.org/abs/2402.03975`

Funding Funded by the European Union (ERC, HOFGA, 101041696). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. Also supported by FCT through the LASIGE Research Unit, ref. UIDB/00408/2020 and ref. UIDP/00408/2020.

Acknowledgements MS would like to thank Xavier Allamigeon, Stéphane Gaubert, and Ricardo D. Katz for many useful discussions on mean-payoff games, policy iteration, and the operator approach, for exchanging ideas about the problem of smoothed analysis, for their remarks on a preliminary version of this paper, and for being a perpetual source of friendship and inspiration.

1 Extended Abstract

1.1 A history of discounted and mean-payoff games

John von Neumann proved his minimax theorem in 1928, founding game theory by showing the existence of optimal strategies in zero-sum matrix games. In 1953, Lloyd Shapley [75] considered what happened if two players repeatedly played a zero-sum matrix game. The overall game proceeds as follows. We have n states, and to each state $i \in [n]$ corresponds a zero-sum matrix game G_i. At each round, the two players are in some state $i \in [n]$ and play the corresponding game G_i, with each player simultaneously choosing an action out of a finite set of possible actions. The two players' choice of actions determines not just the payoff, but also the state at the next round. This is repeated ad-infinitum.

In these games, randomness is possible in two different ways. First, the state at the next round can be chosen stochastically, or deterministically, based on the current state and on the players' chosen actions. This gives us two variants: *stochastic* games, and *deterministic* games, with the latter being a special case of the former. Second, the players' choice of action can itself be *pure* (a single action), or *mixed* (a distribution over the possible actions).

In an infinite game such as this there are two natural ways of determining the winning player. In the *discounted* variant, payoffs received at round t are multiplied by a *discount factor* of γ^t (for some $0 \leqslant \gamma < 1$), and we wish to know the total discounted payoff in the limit as the number of rounds goes to infinity. This is equivalent to saying that the game is forced to stop after every round with probability $1 - \gamma$, and asking for the expected payoff at the limit. In the *mean-payoff* variant, we measure the liminf or limsup, as the number of rounds goes to infinity, of the average payoff received so far (i.e. total payoff divided by the number of rounds). Shapley [75] proved the existence of a value and optimal (mixed) strategies for the stochastic, discounted variant.

Concurrently to Shapley's work, Bellman [16] studied a class of problems which he termed *Markov Decision Processes* (MDPs). MDPs model decision making when the result of one's actions can be partly random, and they can be seen as *single-player* variant of stochastic games. At each round, the player finds himself in a given state out of a finite number of states, and chooses an action. Depending on his choice he receives a payoff, and transitions to a different state. This transition can be either deterministic or stochastic. The player's goal is to maximize the discounted payoff or mean payoff at the limit, as the number of rounds goes to infinity. Bellman provided a method to find an optimal pure strategy in the discounted variant.

In both MDPs and in discounted games the optimal strategies can be made *memoryless*, in that the choice of what to do only depends on the current state i, and not on the past history. In the general case of discounted stochastic games where the players play simultaneously, the optimal memoryless strategies must be mixed. In the case of MDPs, the optimal memoryless strategy can further be made *pure*.

As for the mean-payoff variant, Gillette [42] gave an example of a mean-payoff two-player game, where the players play simultaneously at each round, whose optimal strategies cannot be memoryless.[1] With this in mind, Gillette introduced a *turn-based* variant of Shapley's infinite game, where two players play in turns. At each round, the game is in some state i, and one of the players (depending on i) chooses an action, which determines the next state (stochastically or deterministically) and a resulting payoff. One player is trying to maximize the payoff at the limit, and the other tries to minimize it. Gillette claimed that turn-based two-player games have a value and that optimal strategies exist for both players which are both memoryless and pure. Gillette's proof was actually wrong, but it was later corrected by Liggett and Lippman [60], so the statement *is* true. It also implies the corresponding statement for the mean-payoff variant of MDPs, as a special case.

[1] In fact, it was only in the 1980s [63] that simultanous-move mean-payoff games were proven to have a value. For every $\varepsilon > 0$, optimal (not memoryless) strategies exist for each player ensuring that the payoff is ε-close to the value.

A pure, memoryless strategy for such games is called a *policy*, and it is a finite object: one can represent it as a finite function $\sigma : \mathsf{States} \to \mathsf{Actions}$ specifying the chosen action at each state. In this paper, we will not concern ourselves with simultaneous two-player games, and only consider single-player games and turn-based two-player games. We will use the informal nomenclature "deterministic/stochastic single-player/two-player discounted/mean-payoff game" to denote each of the eight variants of (non-simultaneous) games just mentioned. Let us also use the term "discounted and mean-payoff games" to refer to these games as whole.

1.2 Algorithms

The above results show that all eight variants have a value, and that optimal strategies are policies, hence finite objects. It then makes sense to consider the algorithmic problem of *solving* such a game: given as input a specification of the game with rational weights (and with rational discount factor, if applicable), compute the value of the game, and optimal policies for the players.[2] The study of discounted and mean-payoff games has always been accompanied by the development of algorithms for solving them. Most algorithms for solving discounted and mean-payoff games can be broadly classified into three families: value-iteration algorithms[3], algorithms for MDPs that use linear programming[4], and, of particular importance to us, policy iteration algorithms.

Policy iteration algorithms have been invented for solving all variants of games described above. These algorithms maintain a policy in memory, and proceed by repeatedly modifying the policy, so that its quality improves monotonically according to some measure, until it can no longer be improved, at which point the measure must guarantee that we have found an optimal strategy for both players.

The first policy iteration algorithm was invented by Howard [52], and finds an optimal strategy for deterministic (and some stochastic) Markov Decision Processes. This was later extended by Denardo and Fox [30] to work on all stochastic MDPs. The method was first extended to two-player mean-payoff games by Hoffman and Karp [50], and two-player discounted games by Denardo [29], with later developments by many other authors [70, 67, 25, 40, 69, 26]. A good historical overview with more technical details appears in [3], where a policy iteration algorithm first appeared that can handle all the variants of mean-payoff games. In the case of discounted games, an optimal strategy can be found in time polynomial in $\frac{1}{1-\gamma}$ [78, 47]. Otherwise, for mean-payoff games, or for discount factors γ exponentially close to 1, no upper-bound is known on the number of iterations, significantly better than the number of policies, which is exponential in the number n of states. More precisely, the best upper-bound on the number of iterations is $2^{\tilde{O}(\sqrt{n})}$ [46, 48].

1.3 Policy iteration versus the simplex method

When one first studies policy iteration algorithms, one gets a sense of familiarity, as if policy iteration algorithms are analogous to the simplex method for linear programming. The intuitive sense is that the choice of policy plays the same role as the choice of basic feasible solution in the simplex method, with a change in policy being analogous to a pivot operation.

2 It can be shown that the value of such a game with rational weights (and discount factor) is a rational number of comparable size.

3 The first algorithm ever invented was a value iteration algorithm [16]. For a modern value-iteration algorithm for single-player games, see [76], which contains a historical overview in Section 2. For two players see, e.g., [79, 54, 23, 13, 8].

4 See, e.g., Chapter 2 of [37], or various sections of [68].

In fact, in some cases, this analogy can be formally established. It is possible to express a MDP by a particular linear program, and in this particular case the connection is perfect: a simplex pivoting rule gives us a policy iteration algorithms for MDPs, and any policy iteration algorithm that switches a single node at a time gives us a pivoting rule for applying simplex on this particular program.

As a result, many known counter-examples for the simplex method, showing that certain pivoting rules require an exponential number of pivots, were devised by first finding examples of MDPs for which certain policy-iteration algorithms need an exponential number of iterations, and then *translating* the counter-example to work for the simplex algorithm, by the above connection [38, 33].

More broadly, it turns out that deterministic two-player mean-payoff games are exactly equivalent to tropical linear programming, i.e., solving systems of "linear" inequalities over the tropical $(\min, +)$ semiring. This was first explicitly shown by Akian, Gaubert, and Guterman [4], strengthening earlier connections between these problems that were made in the literature on tropical algebra (such as [40, 32, 58]) and in works on scheduling problems [64].[5]

Furthermore, tropical linear programs can be reduced to linear programs over the non-Archimedean field of convergent generalized power series [31, 11].[6] This characterization has been exploited to show that, if there exists a strongly-polynomial-time pivoting rule for the simplex algorithm, where the choice of basis element to pivot is semialgebraic in a certain technical sense (and this is the case for many pivoting rules), then the entire algorithm can be tropicalized, to get a polynomial-time algorithm for deterministic two-player mean-payoff games [10].

The analogy between policy iteration and the simplex method is also seen in practice. The aforementioned counter-examples show that policy-iteration algorithms run in exponential time in the worst-case. And yet, various benchmarks have shown that policy-iteration algorithms are very efficient at solving real-world instances, both for single-player [41, 27, 59] and two-player games [32, 21]. This difference between worst-case and real-life performance is also what happens with the simplex method. And in both cases it begs the question: *why?*

1.4 Smoothed analysis

In the case of the simplex method, the generally accepted explanation was proposed by Spielman and Teng [77]. They have shown that, if one takes any linear programming instance $\max\{c \cdot x \mid A \cdot x \geqslant b\}$ of dimension n, and perturbs each entry of A, b and c by a Gaussian with mean 0 and standard deviation $\frac{1}{\phi}$, then the simplex method, with a particular choice of pivoting rule, will solve the resulting perturbed system in time $\text{poly}(\phi \cdot n)$ [77, 28]. It is then reasonable to expect the simplex method to work efficiently on real-world instances, since they incorporate real-world data which is prone to such perturbations. It was this result of Spielman and Teng that founded the area of *smoothed analysis*, where one studies the efficiency of algorithms on such perturbed inputs.

[5] Stochastic two-player mean-payoff games, on the other hand, are equivalent to tropical semidefinite programming [14].

[6] A linear program over such a field can be thought of as a parametric family of linear programs over $\mathbb{R}$. It follows from the above reduction that deterministic mean-payoff games can be encoded as linear programs with coefficients of exponential bit-length. Such an encoding was first derived by Schewe [73], without reference to non-Archimedean fields.

The question then naturally follows: are policy-iteration algorithms efficient in the sense of smoothed analysis?

Recent evidence seems to indicate that no, they are not. The first policy-iteration algorithm for single-player games (MDPs), by Howard [52], determines for the current policy $\sigma : \mathsf{States} \to \mathsf{Actions}$, and for each state i of the game, if a local improvement is possible: *would a different choice of action at i improve the value of the game when starting at i, if the game were to be played according to σ at every other state?* The algorithm then changes the action $\sigma(i)$ at every state i where such a local improvement is possible, to the best possible local improvement. This is sometimes called *Howard's policy iteration*, or the *greedy all-switches* rule.

Last year, Christ and Yannakakis [24] showed a remarkable lower bound. They showed that $2^{\Omega(n^c)}$ iterations are necessary on a certain family of stochastic MDPs (single-player games), even when the payoffs are perturbed. In fact, the lower bound holds not only probabilistically, where each payoff is independently perturbed by a Gaussian with standard deviation $\frac{1}{\mathrm{poly}(n)}$, but even adversarially, where each payoff is perturbed by any value within $\pm\frac{1}{\mathrm{poly}(n)}$.

It is surprising that such a bound can be proven at all. However, their result only holds for *stochastic* games, and does not necessarily apply to deterministic games, where it has been previously conjectured that Howard's rule is efficient [49]. Also, this result shows that a particular way of improving the policy, the greedy all-switches rule, does not give us an efficient algorithm (in the sense of smoothed analysis). This is analogous to saying that a particular pivoting rule in the simplex algorithm is not efficient, and does not exclude the possibility that other ways of improving the policy might work.

Exploiting any one of these caveats could in principle allow for a smoothed analysis for policy iteration. Our main result exploits both: we show that for *deterministic* two-player games, a *slightly different* policy-improvement method will be efficient, in the sense of smoothed analysis.

▶ **Theorem 1** (our main theorem). *There exists a policy-iteration algorithm for solving n-state deterministic two-player (discounted or mean-payoff) games played on ergodic graphs, which runs in time $\mathrm{poly}(\phi \cdot n)$ with high probability, on an input where normalized payoffs in $[-1, 1]$ have been independently perturbed by a Gaussian with mean 0 and standard deviation $\frac{1}{\phi}$.*

It should be emphasized that the lower bound of Christ and Yannakakis holds even for the single-player case, and our result could be contrasted with theirs, even if we only had proved it for the single-player case. However, our policy-iteration algorithm (our upper-bound) works even for two-player games, which are much harder. Our policy-improvement rule is similar to the greedy all-switches rule, except that the choice of switches is allowed to depend on an additional parameter (a discount factor) which evolves over time.

1.5 Computational complexity

Our result should also be contrasted with the case of the simplex algorithm for linear programming. Recall that we *do* know polynomial-time algorithms for linear programming, it is only the simplex algorithm which fails to run efficiently in the worst case. However, it should be emphasized that we do *not* know any polynomial-time algorithms for solving two-player discounted or mean-payoff games.

Indeed, solving one-player discounted and mean-payoff games reduces to linear programming, so we have polynomial-time algorithms. But the complexity of solving *two-player* discounted and mean-payoff games is one of the great unsolved problems in computational

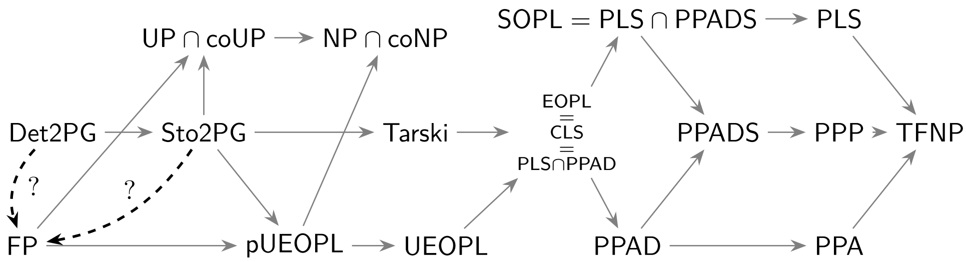

Figure 1 A hierarchy of NP search problems. Det2PG and Sto2PG refer to deterministic, respectively stochastic, two-player games. Arrows denote inclusion or containment. By inclusion of a search problem in the classes UP ∩ coUP and NP ∩ coUP of decision problems, we mean that the problem of deciding each bit of the unique answer can be computed in these classes.

complexity, first posed by Gurvich, Karzanov, and Khachiyan [44]. For any of the two-player variants, the respective decision problem (what is the i-th bit of the value) is in UP∩coUP [56], and the search problem (find an optimal strategy) is in the computational complexity class UEOPL (*Unique End-of-Potential-Line*) [53, 35]. In fact it is in the promise version of this class, which we denote pUEOPL. Even within the class pUEOPL, the problem of solving two-player discounted and mean-payoff games seems to be only a special, simple case: the sought optimal policies can be obtained from the unique fixed point of a simple monotone operator. This places the search problem in the class Tarski. The two complexity classes pUEOPL and Tarski sit at the bottom of a large hierarchy of complexity classes [34, 43]. This hierarchy stratifies the broad class TFNP of NP *search problems* [55, 62, 66]. See Figure 1.

In this sense, the problem of solving two-player discounted and mean-payoff games is the simplest known problem in NP, which is not yet known to be solvable in polynomial time (or even in time $2^{n^{o(1)}}$). For any problem which is not known to be in P, one may ask the question: *is the problem still hard on a random instance?* Many hard problems have been conjectured to have this property, of being hard to solve even for a random instance. This is the case for SAT [74, 36] and subset sum [72], but also for other, not necessarily NP-hard, problems in NP, such as lattice problems [2]. There is a broad belief that natural problems which are hard, remain hard on random inputs.[7] In contrast, our main theorem generalizes to show that solving deterministic two-player games is easy, for any sufficiently random input distribution.

▶ **Theorem 2** (generalization). *Consider distributions on n-state deterministic two-player discounted or mean-payoff games played on an ergodic graph, where each payoff is chosen independently according to a (not necessarily identical) distribution with mean in $[-1, 1]$ and standard deviation $\leqslant \frac{1}{\phi}$ and with probability density functions satisfying $f(y) \leqslant \phi$ for all $y \in \mathbb{R}$. (This includes, for example, payoffs perturbed by a Gaussian, or payoffs sampled from a uniform interval of length $\geqslant \frac{1}{\phi}$.)*

There exists a policy-iteration algorithm which runs in time $\mathrm{poly}(n, \phi)$*, with high probability, on games sampled according to any such distribution.*

[7] The distributions which are considered hard must be chosen carefully to avoid trivial cases, e.g. CNF formulas with too many or too few clauses, but there are often simple and natural distributions.

1.6 A previous approach and our approach

A first, naive attempt at proving Theorem 1 could proceed along the same lines as the Mulmuley, Vazirani and Vazirani's isolation lemma [65]. We think of what happens to the game when all of the payoffs of all the actions are fixed, except for one, which is sampled independently according to some distribution as above. Let us suppose that the free payoff is for an action of some state i. As it turns out, when every other payoff is fixed, the value of the game at state i is a piecewise linear function of the free payoff, and one can then hope that it has few break points. If this were indeed the case, that there were only poly(n) break points, one could then argue similarly to the MVV isolation lemma, to show that approximating the random payoffs using $O(\log(n))$ bits of precision is enough to isolate the linear piece (in the piecewise linear function). From this it would follow that optimal policies for the truncated payoffs are also optimal for the untruncated payoffs. One could then invoke a pseudo-polynomial-time value-iteration algorithm [79] on the truncated payoffs, and this would run in polynomial time.

The conference version of the paper of Boros, Elbassioni, Fouz, Gurvich, Makino and Manthey [18] outlines a similar proof strategy. Among other results, a result similar to our Theorem 1 was claimed [18, Theorem 4.6]. Their proof works for the one-player case, and the authors claimed, without a careful proof, that the two-player case also follows. This claim was later retracted in the journal version [19].[8] Indeed, it turns out that the two-player case is significantly more subtle.

In the one-player case it can be shown that for every action there exist at most n break points, and hence an isolation lemma can be proven. One can then conjecture, for the two-player case, a poly(n) upper-bound on the number of break points. As it turns out, this conjecture is wrong. An exponential example can be created using the construction of [17] that was also used in [12] to prove that interior point methods for linear programming are not strongly polynomial. This construction gives us a deterministic two-player game with n states such that, leaving the payoffs of all but one of actions fixed, as the free payoff varies between -1 and 1, the value of the game is a piecewise-linear function with $2^{\Omega(n)}$ break points.

So what do we do instead? One natural thing to try is to show that the number of break points *is* poly(n), with very high probability, for randomly-chosen payoffs. This could well be true, and an argument in the style of the MVV isolation lemma would then follow. But we were unable to show it.

Instead, our results depend on a deeper analysis of deterministic two-player games. We also prove an isolation lemma, but using an approach different to MVV. Instead of attempting to isolate an optimal policy among all possible policies, we show that sufficiently random payoffs, with high probability, isolate a Blackwell-optimal policy. Blackwell-optimal policies are policies that arise in discounted games with discount factor γ close to 1. Blackwell-optimal policies are part of a family of policies which are induced by an object called a *bias*. Not all optimal policies are Blackwell-optimal, or even bias-induced. Nonetheless, every two-player discounted game has a Blackwell-optimal policy [60]. Furthermore, there exist policy-iteration algorithms for finding a Blackwell-optimal policy [57, 51] (that are inefficient in the worst case).

[8] In their paper, the breakpoints are chosen so that between any two breakpoints the value of the optimal strategy and the value of the second-best strategy are sufficiently far apart. This works for the single player-case, but the argument we just presented, where we only keep track of the number of break points in the value function, is simpler and also works.

We are then able to show that, if the payoffs are sufficiently random, then with high probability it will happen that there is a unique bias-induced policy, which must then be the Blackwell-optimal policy. Furthermore, from the proof of this theorem we devise a *condition number* $\Delta(r)$, associating a number in $[0, +\infty]$ to any given choice r of payoffs. The uniqueness proof generalizes to show that a deterministic two-player game with sufficiently random payoffs will have a small condition number ($\Delta(r) \leqslant \mathrm{poly}(n)$) with high probability.

This is a condition number in the same sense as the known condition numbers that govern the complexity of algorithms for solving linear equations, semidefinite feasibility, *etc*, and broadly measure the *inverse-distance to ill-posedness* (see [20]). Although, strictly speaking, our condition number does not measure inverse-distance to some set, it does have the property that $\Delta(r)$ is finite if and only if the game with payoffs r has a unique bias-induced policy, and that every payoff $\tilde{r} \in B_\infty(r, \delta)$, within a ball of size $\delta \leqslant \frac{1}{\mathrm{poly}(\Delta(r))}$ around r, will also have a unique bias-induced policy. So, at least intuitively, we can think of $\Delta(r)$ as an inverse-distance between r and the set of games with multiple bias-induced policies.

Finally, it can be shown that, taking a discount rate $\gamma \geqslant 1 - \frac{1}{\mathrm{poly}(n,\Delta)}$ (i.e., sufficiently close to 1), the only optimal policy is the Blackwell-optimal policy. We can then use the results of [78, 47] to obtain a policy iteration algorithm for finding the Blackwell optimal policy in time $\mathrm{poly}(n, \Delta)$. This algorithm runs in polynomial smoothed time, because, as mentioned above, sufficiently random payoffs have small condition number. This includes any fixed choice of payoffs that has been randomly perturbed by a Gaussian.

1.7 Related work

There is not a lot of work on the complexity of discounted and mean-payoff games with random payoffs. Besides the papers of Boros et al. [18, 19] and Christ and Yannakakis [24], which we mentioned above, we only know of a paper by Mathieu and Wilson [61]. They do not provide an algorithm, but they analyze the distribution of the value of a deterministic single-player mean-payoff game (deterministic MDP) played on a complete graph with i.i.d. exponentially distributed payoffs. Our algorithm will also work on such a payoff distribution.

A paper of Allamigeon, Benchimol, and Gaubert [9] analyzes efficiency in a different random model. The shadow vertex rule is known to be efficient on average under any distribution over linear feasibility problems, which is symmetric up to changing of the sign in each linear inequality [1]. Allamigeon et al. tropicalize this result, to show that a certain tropical analogue of the shadow vertex rule will solve deterministic two-player mean-payoff games in a bipartite graph in expected polynomial time, if the distribution over the payoff matrix is invariant under transposition (this is the tropical analogue of the above symmetry). In particular, if the payoffs of some given fixed input obey the same symmetry, their algorithm runs in polynomial time. Of course, in general, perturbed payoffs need not be symmetric in this way.

It was a paper of Frieze and Sorkin [39] that gave us the first idea of how to approach the problem. Frieze and Sorkin analyse the gap between optimal and second-optimal assignment in the assignment problem, using a bound on the reduced costs of the associated linear program at the optimum solution [39, Theorem 3]. In the simplex algorithm, the reduced cost works as a gradient, telling us the improvement in the objective function obtained by changing the current basic solution in a given direction. Frieze and Sorkin show that, at an optimum basic solution of a random assignment problem, every reduced cost is large, which implies that there is a large difference between the optimum and second-best solution. This large difference implies that the optimum solution is stable under perturbations. In the case of deterministic two-player games, biases will play the role of dual variables, which allows us define an analogous notion of reduced costs at the optimal solution. We then show, analogously, that, with high probability for a random instance, every reduced cost is large at the optimum policy, which also implies stability. Our condition number is the (normalized) inverse of the smallest reduced cost.

Our analysis of discounted and mean-payoff games is based on the operator approach to study these games. Using this approach, Akian, Gaubert, and Hochart [6] have previously shown that a generic two-player mean-payoff game has a unique bias (which must then equal the Blackwell bias). More precisely, they show that for any stochastic or deterministic two-player mean-payoff game, the set of payoffs where the bias is not unique has measure zero. We give a more precise version of this result, for deterministic two-player mean-payoff games, by showing that the policies induced by the unique bias are also unique. This further allows us the measure "how far from having multiple bias-induced policies" is a given choice of payoffs.

The operator approach was also used by Allamigeon, Gaubert, Katz and Skomra [13], who define a condition number for the value iteration algorithm. Computer experiments indicate that value iteration converges quickly for random games [14], which strongly suggests that the condition number of [13] is small for random games, but there is currently no formal proof of this claim. Even though our condition number and the one from [13] are based on the bias vector, it is not clear how these two conditions numbers compare to each other. In particular, we do not know if the value iteration algorithm has polynomial smoothed complexity and we leave this problem as an open question.

2 Technical summary

For the sake of simplicity, we restrict our attention to deterministic mean-payoff games played on an *ergodic* weighted directed graph $\vec{\mathcal{G}} = ([n], E, r)$, where $|E| = m$ and $[n]$ is split into vertices controlled by players Max and Min, $[n] = V_{\mathrm{Max}} \uplus V_{\mathrm{Min}}$. The ergodicity is taken in the sense of [45, 5, 7]: a graph is called ergodic if the value of any mean-payoff game played on this graph is independent of the initial state of the game.[9] A typical example of such a graph is a complete bipartite graph, in which the bipartition is formed by $V_{\mathrm{Max}}, V_{\mathrm{Min}}$. Ergodic graphs are representative for the difficulty of mean-payoff games, because solving games on general graphs reduces to solving games on complete bipartite graphs [22]. We note however that it is not clear if this reduction can be done in the smoothed analysis setting. We leave the problem of extending our results to non-ergodic graphs as a question for future research.

The weights r_{ij} of the edges in our model are chosen randomly: we suppose that (r_{ij}) are independent absolutely continuous variables with densities f_{ij}. We further suppose that weights are normalized – $\mathbb{E}(r_{ij}) \in [-1, 1]$ – and that there exists a number $\phi > 0$ such that $f_{ij}(y) \leqslant \phi$ for all i, j, y and $\mathrm{Var}(r_{ij}) \leqslant 1/\phi^2$. As an example, if the weights r are taken by perturbing some fixed weights $\bar{r}_{ij} \in [-1, 1]$ by Gaussian noise, so that $r_{ij} \sim \mathcal{N}(\bar{r}_{ij}, \rho^2)$, then we can take $\phi \coloneqq 1/\rho$.

Under the ergodicity assumption, it is known [44, 7] that the following *ergodic equation* has a solution $(\lambda, u) \in \mathbb{R}^{n+1}$ for all choices of weights:

$$\begin{cases} \forall i \in V_{\mathrm{Max}}, \ \lambda + u_i = \max_{(i,j)\in E}\{r_{ij} + u_j\}\,, \\ \forall i \in V_{\mathrm{Min}}, \ \lambda + u_i = \min_{(i,j)\in E}\{r_{ij} + u_j\}\,. \end{cases}$$

[9] This is a notion similar to strong connectivity, but for two-player games. Intuitively, a graph is ergodic if no player can play in such a way as to force the game to get stuck on a sub-graph.

Furthermore, the number λ is unique and it is the value of the game (which does not depend on the initial state because of ergodicity)[10]. The vector $u \in \mathbb{R}^n$, called a *bias*, is never unique because the set of solutions contains at least one line: we can always add a constant to all coordinates of u. In general, the set of biases may consist of more than one line. We say that a pair of policies $\sigma\colon V_{\mathrm{Max}} \to V$ (of Max) and $\tau\colon V_{\mathrm{Min}} \to V$ (of Min) is *bias-induced* if there is a bias u such that the edges used by σ, τ achieve the maxima and minima in the ergodic equation. Bias-induced policies are optimal [44], but not every optimal policy is bias-induced. To study the behavior of random games, we introduce the sets

$$\begin{aligned}\mathcal{P}^{\sigma,\tau} := \{r \in \mathbb{R}^m \colon &(\sigma,\tau) \text{ is the only pair of bias-induced policies} \\ &\text{in the MPG with weights } r\},\end{aligned}$$

for any pair (σ, τ) such that the resulting graph $\vec{\mathcal{G}}^{\sigma,\tau}$ has a single directed cycle. We denote by $\varXi$ the set of all such pairs of policies. We also put $\mathcal{U} := \cup \mathcal{P}^{\sigma,\tau}$. Using the techniques from [6] we are then able to show the following proposition. This proposition strengthens [6, Theorem 3.2] for deterministic games by showing that each maximum and minimum in the ergodic equation is generically achieved by a single edge.

▶ **Proposition 3** (cf. [6, Theorem 3.2]). *The sets $\mathcal{P}^{\sigma,\tau}$ are open polyhedral cones. Moreover, these cones are disjoint and $\mathbb{R}^m \setminus \mathcal{U}$ is included in a finite union of hyperplanes. In particular, this set has Lebesgue measure zero. Furthermore, if $r \in \mathcal{U}$, then the ergodic equation has a single solution (up to adding a constant to the bias), and each maximum and minimum in the ergodic equation is achieved by a single edge.*

This motivates the introduction of the following *condition number $\varDelta$*, which measures the difference between the edge that achieves a maximum or minimum in the ergodic equation and the “second best” edge, relatively to the spread of the weights around the value:

▶ **Definition 4.** *Given $r \in \mathcal{U}$, we put*

$$\varDelta(r) := \frac{\max\{|r_{ij} - \lambda| \colon (i,j) \in E\}}{\min\{|r_{ij} - \lambda + u_j - u_i| \colon (i,j) \in E, r_{ij} - \lambda + u_j - u_i \neq 0\}}\,.$$

When defined in such a way, the condition number does not change when the weights are multiplied by a positive constant, or when the same constant is added to all the weights.

To analyze the behavior of random games, we introduce the following random variables. If i is a vertex controlled by Min, then for any edge $(i, j) \in E$ we put

$$\begin{aligned}Z_{ij} = \inf\{x \in \mathbb{R} \colon &(x, r_{-ij}) \in \mathcal{U} \text{ and the MPG with weights } (x, r_{-ij}) \text{ has a pair} \\ &\text{of bias-induced policies } (\sigma,\tau) \in \varXi \text{ such that } \tau(i) \neq j\}\,.\end{aligned}$$

Here, (x, r_{-ij}) is the vector obtained from r by replacing the ijth coordinate with x. We analogously define the variables Z_{ij} for vertices controlled by Max, changing inf to sup. Since the variable Z_{ij} does not depend on r_{ij}, we get the estimate

▶ **Lemma 5.** *For any $\alpha > 0$, $\mathbb{P}(\exists ij, |r_{ij} - Z_{ij}| \leqslant \alpha) \leqslant 2\alpha m\phi$.*

---

[10] This is a fundamental result which appeared already in [44]: in the paper’s only theorem, $p(v)$ is the value and $c'_{ij} = r_{ij} + u_j - u_i$ are the payoffs modified by u. In the reference [7], the existence of λ and u is line (iii) of the much more general Theorem 2.1, which applies to additive eigenvectors of additively homogeneous monotone operators.

Furthermore, the variables Z_{ij} are related to the ergodic equation in the following way.

▶ **Lemma 6.** *Suppose that $r \in \mathcal{P}^{\sigma,\tau}$ for some $(\sigma,\tau) \in \Xi$. Then, for every $(i,j) \in E$ that is not used in (σ,τ) we have $Z_{ij} = \lambda + u_i - u_j$.*

The two lemmas above improve the conclusion of Proposition 3: not only each maximum and minimum in the ergodic equation is achieved by a single edge, but with high probability the difference between the best edge and the second best edge is large. In particular, this shows that bias-induced policies do not change when the random weights are truncated, and it gives an estimate of the condition number.

▶ **Theorem 7.** *Let $\delta := 1/(4n(2n+1)m\phi)$. Then, with probability at least $1 - 1/n$, the whole ℓ_∞ ball $B_\infty(r,\delta)$ is included in a single polyhedron $\mathcal{P}^{\sigma,\tau}$.*

▶ **Theorem 8.** *Random mean payoff games are well conditioned with high probability. More formally, for every $\varepsilon > 0$ we have $\mathbb{P}\big(\mathit{\Delta} \geqslant \frac{8m}{\varepsilon}(\phi + \sqrt{\frac{2m}{\varepsilon}})\big) \leqslant \varepsilon$.*

To propose an algorithm that exploits the condition number, we use the fact that every mean-payoff game has a pair of *Blackwell-optimal* policies, i.e., policies that are optimal for all discount factors γ close to 1. Such policies are induced by the *Blackwell bias*, which is defined as $u^* := \lim_{\gamma \to 1}(\lambda^{(\gamma)} - \lambda)/(1-\gamma)$, where $\lambda^{(\gamma)}$ is the value of the discounted game with discount factor γ. We then show that, for well-conditioned games, the Blackwell-optimal policies can be already found when the discount factor is low.

▶ **Theorem 9.** *Suppose that $r \in \mathcal{P}^{\sigma,\tau}$ and fix $1 > \gamma > 1 - \frac{1}{6n^2 \mathit{\Delta}(r)}$. Then, (σ,τ) is the unique pair of optimal policies in the discounted game with discount factor γ.*

Combining Theorems 8 and 9 with the results of [78, 47] showing that policy iteration has polynomial complexity for discount factor $\gamma < 1 - \frac{1}{\mathrm{poly}(n)}$, we get our final result.[11]

▶ **Theorem 10.** *The greedy-all switches policy iteration rule combined with increasing discount factor solves random instances of deterministic discounted or mean-payoff games in polynomial smoothed complexity.*

In the theorem above, "polynomial smoothed complexity" is defined as in [15, 71]: there exists a polynomial $\mathrm{poly}(x_1, x_2, x_3, x_4)$ such that for all $\varepsilon \in]0,1]$ the probability that the number of iterations of our algorithm exceeds $\mathrm{poly}(n, m, \phi, \frac{1}{\varepsilon})$ is at most ε.

3 Conclusion and future work

We gave an analysis of two-player discounted and mean-payoff games, that led to a condition number, and a policy-iteration algorithm which is efficient on well-conditioned inputs. We showed that random inputs are well-conditioned with high probability. A few remarks are in order.

1. Our techniques work for two-player games played on ergodic graphs. In non-ergodic graphs, the value λ_i is not necessarily the same at each vertex i. A folklore reduction, appearing for example in [22], shows that computing the value vector of a non-ergodic

[11] To wit: since the pair of optimal policies is unique, we can find them using an algorithm for discount factor $\gamma < 1 - \frac{1}{\mathrm{poly}(n,\phi)}$, and the same policies will be optimal for any higher γ and also for the mean-payoff game.

game reduces to computing the value of an ergodic game. So one can ask if our algorithms can be used on non-ergodic games. The answer is not obvious. The reduction proceeds in rounds, where in the first round one finds, say, the largest coordinate λ_i of the value vector, and then discards the node i (which requires some care) and repeats. Now, if one takes a non-ergodic game $\vec{\mathcal{G}}$ with sufficiently random payoffs, and applies this reduction, the resulting game is sufficiently random at the first round, but it is not clear what happens in the succeeding rounds. So, as far as we can tell, the question remains open: *Do deterministic two-player discounted and mean-payoff games have polynomial smoothed complexity, when played on non-ergodic graphs?* A possible way of answering this question is by doing an analysis of Blackwell-optimal policies in the non-ergodic case, similar to what we have done here for the ergodic case.

2. Allamigeon, Gaubert, Katz and Skomra [13] show that a certain value-iteration algorithm runs efficiently on all ergodic instances with value λ bounded away from zero. They use $\frac{\max_i u_i - \min_i u_i}{|\lambda|}$ as a condition number. Can we use their result to show that value iteration has polynomial smoothed complexity? I.e., is a sufficiently-random instance well-conditioned as per their condition number? This was the central question left unanswered in their paper, and we tried to solve it, or provide a counter-example, but have so far failed to do so.
3. Our policy-iteration rule is not one of the standard rules (Howard, lexicographic, RandomFacet, *etc*). Do these standard rules also have polynomial smoothed complexity on deterministic two-player games? How about other "combinatorial" algorithms?
4. Can we extend our results to *stochastic* two-player games? The counter-example of Christ and Yannakakis shows that the Howard all-switches rule does not have polynomial smoothed complexity on stochastic two-player games. This seems to indicate that the stochastic setting is more delicate. On the other hand, our policy iteration rule is different to Howard's. So one could tentatively ask: is there a smoothed counter-example to the Howard rule also in the deterministic (say, two-player) setting? This would show that our policy-iteration rule cannot be replaced by the Howard rule.
5. How about other problems in UEOPL? Some of these problems are combinatorial, and do not seem to be amenable to smoothed analysis. But one can consider, for example, the P-Matrix Linear Complementarity Problem (P-LCP, see [35, Section 4.3]), and ask: *does it have polynomial smoothed complexity?* More broadly speaking, one can make the conjecture that *every problem in* UEOPL *becomes easy under a suitable notion of perturbation.* This conjecture is broad and imprecise, but it might be an interesting starting point for further research.

References

1 Ilan Adler, Richard M. Karp, and Ron Shamir. A simplex variant solving an $m \times d$ linear program in $O(\min(m^2, d^2))$ expected number of pivot steps. *Journal of Complexity*, 3(4):372–387, 1987.

2 Miklós Ajtai. Generating hard instances of the short basis problem. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–9, 1999.

3 M. Akian, J. Cochet-Terrasson, S. Detournay, and S. Gaubert. Policy iteration algorithm for zero-sum multichain stochastic games with mean payoff and perfect information, 2012. arXiv:1208.0446.

4 M. Akian, S. Gaubert, and A. Guterman. Tropical polyhedra are equivalent to mean payoff games. *Int. J. Algebra Comput.*, 22(1):125001 (43 pages), 2012. doi:10.1142/S0218196711006674.

5 M. Akian, S. Gaubert, and A. Hochart. Ergodicity conditions for zero-sum games. *Discrete Contin. Dyn. Syst.*, 35(9):3901–3931, 2015. doi:10.3934/dcds.2015.35.3901.

6 M. Akian, S. Gaubert, and A. Hochart. Generic uniqueness of the bias vector of finite zero-sum stochastic games with perfect information. *J. Math. Anal. Appl.*, 457:1038–1064, 2018. doi:10.1016/j.jmaa.2017.07.017.

7 M. Akian, S. Gaubert, and A. Hochart. A game theory approach to the existence and uniqueness of nonlinear Perron-Frobenius eigenvectors. *Discrete & Continuous Dynamical Systems - A*, 40:207–231, 2020. doi:10.3934/dcds.2020009.

8 M. Akian, S. Gaubert, U. Naepels, and B. Terver. Solving irreducible stochastic mean-payoff games and entropy games by relative Krasnoselskii-Mann iteration. In *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:15, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2023.10.

9 X. Allamigeon, P. Benchimol, and S. Gaubert. The tropical shadow-vertex algorithm solves mean payoff games in polynomial time on average. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 8572 of *Lecture Notes in Comput. Sci.*, pages 89–100. Springer, 2014. doi:10.1007/978-3-662-43948-7_8.

10 X. Allamigeon, P. Benchimol, S. Gaubert, and M. Joswig. Combinatorial simplex algorithms can solve mean payoff games. *SIAM J. Optim.*, 24(4):2096–2117, 2014. doi:10.1137/140953800.

11 X. Allamigeon, P. Benchimol, S. Gaubert, and M. Joswig. Tropicalizing the simplex algorithm. *SIAM J. Discrete Math.*, 29(2):751–795, 2015. doi:10.1137/130936464.

12 X. Allamigeon, P. Benchimol, S. Gaubert, and M. Joswig. Log-barrier interior point methods are not strongly polynomial. *SIAM J. Appl. Algebra Geom.*, 2(1):140–178, 2018. doi:10.1137/17M1142132.

13 X. Allamigeon, S. Gaubert, R. D. Katz, and M. Skomra. Universal complexity bounds based on value iteration and application to entropy games. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 126:1–126:20, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

14 X. Allamigeon, S. Gaubert, and M. Skomra. Solving generic nonarchimedean semidefinite programs using stochastic game algorithms. *J. Symbolic Comput.*, 85:25–54, 2018. doi:10.1016/j.jsc.2017.07.002.

15 R. Beier and B. Vöcking. Typical properties of winners and losers in discrete optimization. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 343–352. ACM, 2004. doi:10.1145/1007352.1007409.

16 Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

17 M. Bezem, R. Nieuwenhuis, and E. Rodríguez-Carbonell. Exponential behaviour of the Butkovič–Zimmermann algorithm for solving two-sided linear systems in max-algebra. *Discrete Appl. Math.*, 156(18):3506–3509, 2008. doi:10.1016/j.dam.2008.03.016.

18 E. Boros, K. Elbassioni, M. Fouz, V. Gurvich, K. Makino, and B. Manthey. Stochastic mean payoff games: smoothed analysis and approximation schemes. In *Proceedings of the 38th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 6755 of *Lecture Notes in Comput. Sci.*, pages 147–158. Springer, 2011. doi:10.1007/978-3-642-22006-7_13.

19 Endre Boros, Khaled Elbassioni, Mahmoud Fouz, Vladimir Gurvich, Kazuhisa Makino, and Bodo Manthey. Approximation schemes for stochastic mean payoff games with perfect information and few random positions. *Algorithmica*, 80:3132–3157, 2018.

20 Peter Bürgisser and Felipe Cucker. *Condition: The geometry of numerical algorithms*, volume 349. Springer Science & Business Media, 2013.

21 Jakub Chaloupka. Parallel algorithms for mean-payoff games: An experimental evaluation. In *European Symposium on Algorithms*, pages 599–610. Springer, 2009.

22 K. Chatterjee, M. Henzinger, S. Krinninger, and D. Nanongkai. Polynomial-time algorithms for energy games with special weight structures. *Algorithmica*, 70(3):457–492, 2014. `doi:10.1007/s00453-013-9843-7`.

23 K. Chatterjee and R. Ibsen-Jensen. The complexity of ergodic mean-payoff games. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 8573 of *Lecture Notes in Comput. Sci.*, pages 122–133. Springer, 2014. `doi:10.1007/978-3-662-43951-7_11`.

24 Miranda Christ and Mihalis Yannakakis. The smoothed complexity of policy iteration for Markov decision processes. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1890–1903, 2023.

25 J. Cochet-Terrasson, S. Gaubert, and J. Gunawardena. A constructive fixed point theorem for min-max functions. *Dyn. Stab. Syst.*, 14(4):407–433, 1999. `doi:10.1080/026811199281967`.

26 Jean Cochet-Terrasson and Stéphane Gaubert. A policy iteration algorithm for zero-sum stochastic games with mean payoff. *Comptes Rendus Mathematique*, 343(5):377–382, 2006.

27 Jean Cochet-Terrasson Guy Cohen, Stephane Gaubert, Michael Mc Gettrick, and Jean-Pierre Quadrat. Numerical computation of spectral elements in max-plus algebra. In *Proceedings of the IFAC Conference on System Structure and Control*, 1998.

28 Daniel Dadush and Sophie Huiberts. A friendly smoothed analysis of the simplex method. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 390–403, 2018.

29 Eric V. Denardo. Contraction mappings in the theory underlying dynamic programming. *Siam Review*, 9(2):165–177, 1967.

30 Eric V. Denardo and Bennett L. Fox. Multichain Markov renewal programs. *SIAM Journal on Applied Mathematics*, 16(3):468–487, 1968.

31 M. Develin and J. Yu. Tropical polytopes and cellular resolutions. *Exp. Math.*, 16(3):277–291, 2007. `doi:10.1080/10586458.2007.10129009`.

32 Vishesh Dhingra and Stéphane Gaubert. How to solve large scale deterministic games with mean payoff by policy iteration. In *Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools (ValueTools)*, pages 12–es, 2006. `doi:10.1145/1190095.1190110`.

33 Y. Disser and N. Mosis. A unified worst case for classical simplex and policy iteration pivot rules. In *Proceedings of the 34th International Symposium on Algorithms and Computation (ISAAC)*, pages 27:1–27:17, 2023.

34 John Fearnley, Paul Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of gradient descent: $\mathsf{CLS} = \mathsf{PPAD} \cap \mathsf{PLS}$. *Journal of the ACM*, 70(1):1–74, 2022.

35 John Fearnley, Spencer Gordon, Ruta Mehta, and Rahul Savani. Unique end of potential line. *Journal of Computer and System Sciences*, 114:1–35, 2020.

36 Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the 34th annual ACM Symposium on Theory of Computing (STOC)*, pages 534–543, 2002.

37 J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, New York, 2007. `doi:10.1007/978-1-4612-4054-9`.

38 Oliver Friedmann. *Exponential lower bounds for solving infinitary payoff games and linear programs*. PhD thesis, Ludwig Maximilian University of Munich, 2011.

39 Alan Frieze and Gregory B. Sorkin. The probabilistic relationship between the assignment and asymmetric traveling salesman problems. *SIAM Journal on Computing*, 36(5):1435–1452, 2007.

40 S. Gaubert and J. Gunawardena. The duality theorem for min-max functions. *C. R. Acad. Sci.*, 326(1):43–48, 1998. `doi:10.1016/S0764-4442(97)82710-3`.

41 Loukas Georgiadis, Andrew V Goldberg, Robert E Tarjan, and Renato F Werneck. An experimental study of minimum mean cycle algorithms. In *2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 1–13. SIAM, 2009.

42 D. Gillette. Stochastic games with zero stop probabilities. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games III*, volume 39 of *Ann. of Math. Stud.*, pages 179–188. Princeton University Press, Princeton, NJ, 1957.

43 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Further collapses in TFNP. In *Proceedings of the 37th Computational Complexity Conference (CCC)*, pages 1–15, 2022.

44 V. A. Gurvich, A. V. Karzanov, and L. G. Khachiyan. Cyclic games and finding minimax mean cycles in digraphs. *Zh. Vychisl. Mat. Mat. Fiz.*, 28(9):1406–1417, 1988. doi:10.1016/0041-5553(88)90012-2.

45 V. A. Gurvich and V. N. Lebedev. A criterion and verification of the ergodicity of cyclic game forms. *Russian Math. Surveys*, 44(1):243–244, 1989. doi:10.1070/RM1989v044n01ABEH002010.

46 N. Halman. Simple stochastic games, parity games, mean payoff games and discounted payoff games are all LP-type problems. *Algorithmica*, 49(1):37–50, 2007. doi:10.1007/s00453-007-0175-3.

47 T. D. Hansen, P. B. Miltersen, and U. Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *J. ACM*, 60(1):1–16, 2013. doi:10.1145/2432622.2432623.

48 T. D. Hansen and U. Zwick. An improved version of the Random-Facet pivoting rule for the simplex algorithm. In *Proceedings of the 47th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 209–218. ACM, 2015. doi:10.1145/2746539.2746557.

49 Thomas Dueholm Hansen and Uri Zwick. Lower bounds for Howard's algorithm for finding minimum mean-cost cycles. In *International Symposium on Algorithms and Computation*, pages 415–426. Springer, 2010.

50 A. J. Hoffman and R. M. Karp. On nonterminating stochastic games. *Manag. Sci.*, 12(5):359–370, 1966. doi:10.1287/mnsc.12.5.359.

51 A. Hordijk and A. A. Yushkevich. Blackwell optimality. In E. A. Feinberg and A. Shwartz, editors, *Handbook of Markov Decision Processes: Methods and Applications*, volume 40 of *Internat. Ser. Oper. Res. Management Sci.*, pages 231–267. Springer, Boston, MA, 2002. doi:10.1007/978-1-4615-0805-2_8.

52 Ronald A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, 1960.

53 Pavel Hubácek and Eylon Yogev. Hardness of continuous local search: Query complexity and cryptographic lower bounds. *SIAM Journal on Computing*, 49(6):1128–1172, 2020.

54 R. Ibsen-Jensen and P. B. Miltersen. Solving simple stochastic games with few coin toss positions. In *Proceedings of the 20th Annual European Symposium on Algorithms (ESA)*, volume 7501 of *Lecture Notes in Comput. Sci.*, pages 636–647. Springer, 2012. doi:10.1007/978-3-642-33090-2_55.

55 David S Johnson, Christos H Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of computer and system sciences*, 37(1):79–100, 1988.

56 M. Jurdziński. Deciding the winner in parity games is in UP $\cap$ co-UP. *Inform. Process. Lett.*, 68(3):119–124, 1998. doi:10.1016/S0020-0190(98)00150-1.

57 L. Kallenberg. Finite state and action MDPs. In E. A. Feinberg and A. Shwartz, editors, *Handbook of Markov Decision Processes: Methods and Applications*, volume 40 of *Internat. Ser. Oper. Res. Management Sci.*, pages 21–87. Springer, Boston, MA, 2002. doi:10.1007/978-1-4615-0805-2_2.

58 Ricardo David Katz. Max-plus (A, B)-invariant spaces and control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 52(2):229–241, 2007.

59 Jan Křetínskỳ and Tobias Meggendorfer. Efficient strategy iteration for mean payoff in Markov decision processes. In *International Symposium on Automated Technology for Verification and Analysis*, pages 380–399. Springer, 2017.

60 T. M. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoff. *SIAM Rev.*, 11(4):604–607, 1969. doi:10.1137/1011093.

61 C. Mathieu and D. B. Wilson. The min mean-weight cycle in a random network. *Combin. Probab. Comput.*, 22(5):763–782, 2013. doi:10.1017/S0963548313000229.

62 Nimrod Megiddo and Christos H Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.

63 J.-F. Mertens and A. Neyman. Stochastic games. *Internat. J. Game Theory*, 10(2):53–66, 1981. doi:10.1007/BF01769259.

64 Rolf H. Möhring, Martin Skutella, and Frederik Stork. Scheduling with AND/OR precedence constraints. *SIAM Journal on Computing*, 33(2):393–415, 2004.

65 Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. In *Proceedings of the 19th annual ACM Symposium on Theory of Computing (STOC)*, pages 345–354, 1987.

66 Christos H Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and system Sciences*, 48(3):498–532, 1994.

67 Anuj Puri. *Theory of hybrid systems and discrete event systems*. University of California at Berkeley, 1995.

68 M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Ser. Probab. Stat. Wiley, Hoboken, NJ, 2005.

69 T.E.S. Raghavan and Zamir Syed. A policy-improvement type algorithm for solving zero-sum two-person stochastic games of perfect information. *Mathematical Programming*, 95(3):513–532, 2003.

70 S. S. Rao, R. Chandrasekaran, and K.P.K. Nair. Algorithms for discounted stochastic games. *Journal of Optimization Theory and Applications*, 11(6):627–637, 1973.

71 H. Röglin and B. Vöcking. Smoothed analysis of integer programming. *Math. Program.*, 110(1):21–56, 2007. doi:10.1007/s10107-006-0055-7.

72 Steven Rudich. Super-bits, demi-bits, and NP/qpoly-natural proofs. In *Proceedings of the International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM/APPROX)*, pages 85–93, 1997.

73 Sven Schewe. From parity and payoff games to linear programming. In *Proceedings of the 34th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 675–686, 2009.

74 Bart Selman, David G Mitchell, and Hector J Levesque. Generating hard satisfiability problems. *Artificial intelligence*, 81(1-2):17–29, 1996.

75 L. S. Shapley. Stochastic games. *Proc. Natl. Acad. Sci. USA*, 39(10):1095–1100, 1953. doi:10.1073/pnas.39.10.1095.

76 Aaron Sidford, Mengdi Wang, Xian Wu, and Yinyu Ye. Variance reduced value iteration and faster algorithms for solving Markov decision processes. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 770–787, 2018.

77 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

78 Y. Ye. The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate. *Math. Oper. Res.*, 36(4):593–603, 2011.

79 U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoret. Comput. Sci.*, 158(1–2):343–359, 1996. doi:10.1016/0304-3975(95)00188-3.