Computing Fixpoints of Learned Functions: Chaotic Iteration and Simple Stochastic Games

Paolo Baldan[1], Sebastian Gurke[2], Barbara König[2], and Florian Wittbold[2]

[1] Department of Mathematics, University of Padua, Italy
baldan@math.unipd.it
[2] Universität Duisburg-Essen
{sebastian.gurke,barbara_koenig,florian.wittbold}@uni-due.de

Abstract. The problem of determining the (least) fixpoint of (higher-dimensional) functions over the non-negative reals frequently occurs when dealing with systems endowed with a quantitative semantics. We focus on the situation in which the functions of interest are not known precisely but can only be approximated. As a first contribution we generalize an iteration scheme called dampened Mann iteration, recently introduced in the literature. The improved scheme relaxes previous constraints on parameter sequences, allowing learning rates to converge to zero or not converge at all. While seemingly minor, this flexibility is essential to enable the implementation of chaotic iterations, where only a subset of components is updated in each step, allowing to tackle higher-dimensional problems. Additionally, by allowing learning rates to converge to zero, we can relax conditions on the convergence speed of function approximations, making the method more adaptable to various scenarios. We also show that dampened Mann iteration applies immediately to compute the expected payoff in various probabilistic models, including simple stochastic games, not covered by previous work.

1 Introduction

This paper focuses on the problem of identifying fixpoint iteration schemes for determining the (least) fixpoint of a d-dimensional function f on the reals in scenarios where this function is not known precisely, but can only be approximated. Concretely, we assume to be able to construct a sequence of approximating functions $f_1, f_2, f_3, \ldots$ that converges to f. The problem is non-trivial for a number of reasons, starting from the fact that the least fixpoint operator is not necessarily continuous and hence the sequence of least fixpoints of the functions f_n does not always converge to the least fixpoint of f.

This task can be seen as an abstract formulation of the problem that occurs in reinforcement learning, where a Markov Decision Process (MDP) – a transition system based on probabilistic and non-deterministic branching as well as rewards – is given and the aim is to determine the expected return and the optimal strategy to achieve it. In a real-world scenario the probabilities (and possibly

S. Junges and G. Katz (Eds.): TACAS 2026, LNCS 16505, pp. 547–565, 2026.
https://doi.org/10.1007/978-3-032-22752-2_28

the rewards) of such MDPs are often not known, but can only be determined by exploring the MDP and sampling. Reinforcement learning algorithms such as Q-learning [22], SARSA [17] or Dyna-Q [18,19,12] typically proceed by interleaving exploration of the MDP and strategy computation. These algorithms can be classified as model-based (e.g., Dyna-Q) – where the (approximated) MDP is explicitly computed – and model-free (e.g., SARSA, Q-learning) – where updates are not dependent on the estimated probabilities, but on the current sample.

Indeed, the expected return of a (finite state) MDP can be characterized as the least fixpoint of a function based on the Bellman equation. This function is monotone with respect to the pointwise order on tuples and non-expansive (1-Lipschitz) with respect to the supremum metric, i.e., function application does not increase the distance of two points in the space. The fact that the same happens for a number of other fixpoint equations of interest (e.g., for computing payoffs for simple stochastic games [8] and for behavioural metrics [1]) leads us to develop our theory for monotone and non-expansive functions.

In [3], we proposed a solution to this problem: instead of computing fixpoints via Kleene iteration – as common for such models – the idea is to perform a so-called dampened Mann iteration that adds a dampening factor to the well-known Mann iteration [6]. Concretely the following iteration schema is proposed:

$$x_{n+1} = (1 - \beta_n) \cdot (x_n + \alpha_n \cdot (f_n(x_n) - x_n))$$

The parameter α_n, often referred to as the learning rate, is used to take a weighted sum between the previous value and the value obtained by applying the n-th approximated function, making the iteration more robust with respect to perturbations. A central ingredient is the dampening factor $1 - \beta_n$, inspired by [14]. Its purpose is to decrease a value that for some reasons might overestimate the least fixpoint. This is essential when the function of interest admits more than one fixpoint and the iteration in early phases can get stuck at a fixpoint that is not the least. According to [3], the parameters α_n, β_n have to be chosen such that α_n converges to 1 (the iteration scheme converges to a Kleene iteration) and β_n converges to 0 (the dampening is vanishing over time). For most results the first condition can be relaxed to α_n converging to some $\alpha > 0$.

Then, the sequence x_n generated by the above iteration scheme is guaranteed to converge to the least fixpoint of f in a number of situations: when the function f is a (power) contraction, when the the sequence (f_n) converges monotonically to f and when (f_n) converges normally. The latter means that $\sum_n \|f - f_n\| < \infty$ (where $\|\cdot\|$ is the supremum norm), a condition which can always be enforced by "speeding" up the iteration and determining a function g such that $f_{g(i)}$ converges fast enough and thus normally. For instance, consider the sequence of functions $f_n(x) = 1/n + (1 - 1/n) \cdot x$ converging to the identity $f(x) = x$. It can be seen that (f_n) does not converge normally as $\|f_n - f\| = 1/n$ and, indeed, the dampened Mann iteration scheme from [3] fails. However, normal convergence can be enforced by considering a subsequence $f_{g(n)}$ with $g(n) = n^2$. Whenever approximations are realized by means of samplings, this speedup can be obtained by increasing the number of samplings performed between two iterations.

Interestingly, the mentioned paper also shows that, in the case of MDPs, approximations obtained by sampling always guarantee convergence of the dampened Mann iteration scheme, independently of the speed of convergence.

The solution in [3] has some drawbacks. A relevant issue resides in the fact that it implicitly assumes that all components are updated at each iteration. Clearly, in applications such as reinforcement learning where large models can produce functions working in dimensions of several thousands, this becomes impractical if not impossible. Chaotic iterations [11] instead offer a scalable alternative where only a subset (possibly only one) of the components are updated at each iteration. The components to be updated might be chosen, for example, to be the ones that seem statistically more relevant to the goal (such as states that are visited more often in sampled runs of a Markov decision process) or that are dependent on components that were previously updated. Using chaotic strategies can often result in less computation and while not necessarily so, its faster update cycle can give better insight into the convergence behaviour at runtime or even produce approximations where full updates are infeasible.

While this might look like a minor implementation aspect, one can see that generalizing to chaotic iteration requires quite a radical change to the theory. The parameters α_n and β_n have to become dependent on the dimension (the state), i.e., they become themselves vectors. If we write $z(i)$ for the i-th component of a vector z, the iteration schemes becomes of the following form, where setting $\alpha_n(i) = \beta_n(i) = 0$ we ensure that component i is not updated at step n:

$$x_{n+1}(i) = (1 - \beta_n(i)) \cdot (x_n(i) + \alpha_n(i) \cdot (f_n(x_n) - x_n)(i))$$

Since in a proper chaotic iteration, components may not be updated infinitely often, parameters $\alpha_n(i)$ must be allowed to be 0 infinitely often, which means that they either converge to 0 or do not converge at all. This deeply deviates from the theory in [3], which heavily relies on the learning rate sequence α_n converging to 1 or at least to a value strictly larger than 0.

In this paper, we will indeed show that the iteration scheme can work with more general parameter sequences, where the learning rate can converge to 0 or is even allowed not to converge. This relaxation of the condition on the parameters requires a substantial change in the proof techniques.

Besides enabling chaotic iteration, the additional flexibility for the parameter sequences allows one to treat different components of the system in a different way. If the value of some components is known to be close to the solution, the corresponding $\beta_n(i)$ and $\alpha_n(i)$ should be chosen close to zero. For components where little information is available and and further correction of a potential over-approximation is needed, the parameters should be chosen away from 0.

The more general schema allows us to match it with the practice of a number of model-based reinforcement learning techniques. For instance, Dyna-Q, a model-based extension of Q-learning, uses chaotic iteration and considers a learning rate α_n converging to 0. More generally, the present work might be a useful step also towards the development of a theory encompassing model-free reinforcement learning techniques. In fact, in a model-free approach, such as

Q-learning [22], the update is based on the current sample and not on the estimated probabilities and rewards. Hence, in order to converge to the solution, the learning rate α_n must converge to 0, giving more and more weight to the previous estimate rather than to the result of applying the function, otherwise the iteration would oscillate (due to the varying samples) rather than converge. Existing convergence results (e.g., [7,22]) for these algorithms typically assume functions with unique fixpoints (as is the case for discounted problems or systems with some termination guarantees) or convergence to some fixpoint, while the case of convergence to the least fixpoint is not treated.

Finally, the possibility of setting a learning rate α_n converging to 0 also offers the potential to relax the condition on the speed of convergence of the sequence f_n. While we required in [3] that f_n converges normally to f, i.e., $\sum_n \|f - f_n\| < \infty$, here we can relax the condition to $\sum_n \alpha_n \|f - f_n\| < \infty$, which is easier to satisfy when α_n tends to 0 and thus the iteration relies less and less on the approximations.

Another relevant question – unanswered in [3] – is whether dampened Mann iteration works – out of the box – for simple stochastic games (SSGs). The mentioned paper only offers a solution, sketched above, involving a speedup of the sampling process for ensuring normal convergence. However, this is costly since it requires to obtain many samples before being able to do the next iteration and obtain the next estimate of the solution. Even worse, the periods until the next estimate are getting longer and longer since the number of needed samples might increase non-linearly. The question of whether one can instead directly apply dampened Mann iteration to SSGs was thus relevant and still open.

The results from [3] cannot be easily adapted to handle SSGs. The proof strategy was to first show that dampened Mann iteration works in the exact case and for the case of power contractions. From there one is able to derive other results, for instance for MDPs (where the fact is exploited that the Bellman equations of MDPs without end components result in fixpoint functions that are power contractions). Including SSGs in the picture and accomodating for the generalized parameter sequences, requires completely new proofs. We prove convergence of dampened Mann iteration for sequences of functions converging monotonically and generalize the result for power contractions to the case of Picard operators. Together with the fact that the least fixpoint operator is continuous for Markov Chains and for SSGs, these results allow us to conclude that dampened Mann iteration works for both MDPs and SSGs. For MDPs the novelty lies in the new types of parameters, while for SSGs this result was completely open in [3].

Being able to treat SSGs also allows us to solve (least) fixpoint equations for behavioural metrics [20,21], another important application area. In [13,10], it was observed that a small perturbation of the transition probabilities in a model can drastically change the distance and the authors suggest an alternative, more robust, notion of distance. With our results it is now also possible to keep the original notion of pseudometric and use dampened Mann iteration with increasingly better approximations to converge to the true distance.

In summary, our main contribution is a generalisation of the dampened Mann iteration scheme proposed in previous work for approximating fixpoints of functions arising from quantitative models. More precisely

- We allow the learning rate to converge to 0 or not converge at all, enabling better adaptation to different convergence rates of function approximations.
- We introduce a generalized dampened Mann iteration where the learning rate and dampening parameters can vary across dimensions, allowing for chaotic iteration strategies.
- We establish convergence of the introduced schemes for simple stochastic games and their sampled approximations.

Full proofs for all results can be found in [5].

2 Preliminaries and Notation

We denote by $\mathbb{R} = (-\infty, \infty)$ the set of real numbers, by $\mathbb{R}_+ = [0, \infty)$ the set of non-negative reals, by $\overline{\mathbb{R}} = [-\infty, \infty]$ and $\overline{\mathbb{R}}_+ = [0, \infty]$ the sets of (non-negative) real numbers including infinity, and by $\mathbb{N}_0$ and $\mathbb{N}$ the set of natural numbers with and without 0, respectively.

For sets X, Y, we will denote by Y^X the set of functions $f\colon X \to Y$. If X is finite, we will identify Y^X with the set of vectors $Y^{|X|}$. A sequence in X is denoted by $(x_n)_{n\in\mathbb{N}_0}$ or simply by (x_n) when the index is clear from the context. For $x \in \mathbb{R}^d$ and $i \in \{1, \dots, d\}$, we will write $x(i) \in \mathbb{R}$ to denote its i-th component and, if clear from the context, we will sometimes identify $x \in \mathbb{R}$ with the vector $(x, \dots, x) \in \mathbb{R}^d$. For $x, y \in \mathbb{R}^d$, $x \cdot y = (x(1)y(1), \dots, x(d)y(d)) \in \mathbb{R}^d$ will denote their component-wise product.

For $x, y \in \overline{\mathbb{R}}^d$, we write $x \leq y$ for the pointwise (partial) order, i.e., $x(i) \leq y(i)$ for all $i \in \{1, \dots, d\}$. A function $f \in Y^X$, where $X, Y \subseteq \overline{\mathbb{R}}^d$, is monotone if, for all $x, y \in X$, we have that $x \leq y$ implies $f(x) \leq f(y)$. It is called ω-continuous if, for all $(x_n) \subseteq X$ with $x_1 \leq x_2 \leq \dots$ and $x = \sup_{n\in\mathbb{N}} x_n \in X$, we have $f(x) = \sup_{n\in\mathbb{N}} f(x_n)$. Note that any ω-continuous function is monotone.

We equip $\mathbb{R}^d$ with the *supremum norm* defined as $\|x\| = \sup\{x(i) \mid i \in \{1, \dots, d\}\}$ for $x \in \mathbb{R}^d$ and extended to functions $f \in Y^X$, where $X, Y \subseteq \mathbb{R}^d$, by $\|f\| = \sup_{x\in X} \|f(x)\| \in [0, \infty]$. A function $f \in Y^X$ is L-Lipschitz for a constant $L \in \mathbb{R}_+$ if, for all $x, y \in X$, $\|f(x) - f(y)\| \leq L\|x - y\|$. It is non-expansive if it is 1-Lipschitz. It is a contraction if it is q-Lipschitz for contraction factor $q < 1$.

A fixpoint of $f\colon X \to X, X \subseteq \mathbb{R}^d_+$, is $x \in X$ such that $f(x) = x$. We denote the set of fixpoints of f by $\mathrm{Fix}(f)$ and, in case it exists, the *least fixpoint* of f by μf. We will be interested in approximating least fixpoints of non-expansive and monotone functions $f\colon X \to X$ where X is a 0-*box*, i.e. a set of the form

$$X = \{x \in \mathbb{R}^d_+ \mid x \leq x^*\} \qquad \text{for some } x^* \in \overline{\mathbb{R}}^d_+. \tag{1}$$

Each such function f extends to an ω-continuous function $\overline{f}\colon \overline{\mathbb{R}}^d_+ \to \overline{\mathbb{R}}^d_+$ given by $\overline{f}(x) = \sup\{f(y) \mid y \leq x\}$. Using Kleene's fixpoint theorem, $\overline{f}$ has

a least fixpoint and $\mathrm{Fix}(f) \neq \emptyset$ iff $\mu\overline{f} \in X$ in which case $\mu f = \mu\overline{f}$. Abusing the notation, we will write μf for $\mu\overline{f} \in \overline{\mathbb{R}}_+^d$. A function $f\colon X \to X$ is a power contraction if the k-fold iteration f^k is a contraction for some $k \in \mathbb{N}$. The map f is a Picard operator if it has a unique fixpoint $x^* \in X$ and the sequence $(f^n(x))$ converges to x^* for all $x \in X$. If $f\colon X \to X$ is a contraction on a closed (hence complete) set $X \subseteq \mathbb{R}_+^d$, then, by Banach's theorem, it is a Picard operator.

Dampened Mann iteration. For the rest of the paper, we will be interested in approximating least fixpoints of non-expansive and monotone functions $f\colon X \to X$, where X is a 0-box, given only approximations $f_n\colon X \to X$ of f. In [3], we suggested the use of a dampened Mann iteration, which is a variation of Mann iteration scheme with the addition of dampening factor.

Definition 2.1 (Mann scheme). *A* (dampened) Mann scheme $\mathcal{S}$ *is a pair* $((\alpha_n),(\beta_n))$ *of parameter sequences in* $[0,1)$*, referred to as* learning rates *and* dampening factors*, respectively, such that*

$$\lim_{n\to\infty} \beta_n = 0, \tag{2}$$

$$\sum_{n\in\mathbb{N}_0} \beta_n = \infty \quad \textit{(or equivalently,} \quad \prod_{n\in\mathbb{N}_0} (1-\beta_n) = 0\textit{).} \tag{3}$$

Given a 0*-box* $X \subseteq \mathbb{R}_+^d$ *and a sequence* (f_n) *of functions* $f_n\colon X \to X$*, a Mann scheme* $\mathcal{S}$ *generates a class of sequences* (x_n) *with* $x_0 \in X$ *chosen aritrarily and*

$$x_{n+1} = (1-\beta_n)\cdot(x_n + \alpha_n \cdot (f_n(x_n) - x_n)) \tag{4}$$

A Mann scheme $\mathcal{S}$ *is* exact *if for all monotone and non-expansive* $f\colon X \to X$*, setting* $f_n = f$ *for all* n*, the sequences* (x_n) *generated by* $\mathcal{S}$ *converge to* μf*.*

Intuitively, the learning rates determine to what extent the update $f(x_n) - x_n$ is applied to the current guess x_n, ranging from a full Kleene update ($\alpha_n = 1$, implying $x_{n+1} = (1-\beta_n)f_n(x_n)$) to no update at all ($\alpha_n = 0$, implying $x_{n+1} = (1-\beta_n)x_n$). The dampening factors ensure that over-approximations of μf introduced by starting from x_0 instead of 0 and using f_n instead of f, and which would lead to possible divergence of a Kleene iteration, can be dampened. Condition (2) ensures that dampening eventually reduces – meaning that, in the long run, when we are close to the least fixpoint of f, we stay close to it. On the other hand, Condition (3) guarantees that at any stage there is still enough "dampening power" left to correct possible over-approximations.

In [3], we showed several convergence results for *Mann-Kleene schemes*, i.e., Mann schemes such that $\lim_{n\to\infty} \alpha_n = 1$ and *relaxed Mann-Kleene schemes* where $\lim_{n\to\infty} \alpha_n > 0$.[3] More precisely, it shows that (relaxed) Mann-Kleene schemes are exact and it proves convergence in the "approximated case", i.e., when we use at each iteration approximations from a sequence (f_n) converging to f, with conditions on the target functions f (power contraction) or on the mode of convergence of the sequence (f_n) (monotone or normal convergence).

[3] Note that in [3] α_n is replaced with $1-\alpha_n$. Here, we chose this notation instead as it better reflects the widespread interpretation of α_n as a learning rate.

3 Generalizing Dampened Mann Iteration

In this section we generalize the convergence results for Mann schemes from [3]. First, in §3.1 we overcome the restriction to Mann-Kleene schemes, and show convergence results for learning rates which are allowed to converge to 0 or to not converge at all. Relying on this, in §3.2 we further generalize the scheme by allowing the parameter values α_n, β_n to depend on the dimension, thus, in particular, enabling a chaotic iteration. While the results for chaotic iteration in §3.2 are for the exact case, i.e., the function f is known and used at each iteration, the extension to the case in which f can be only approximated via a sequence (f_n) is described in §3.3.

3.1 Schemes With Non-Converging Learning Rates

We start by lifting the restriction to *(relaxed) Mann-Kleene* schemes, showing that convergence can be guaranteed also when the learning rate does not converge or converges to zero. In addition, while [3] restricts to functions $f\colon X \to X$ with $\mu f < \infty$, here we allow the fixpoint to be infinite in some (or all) dimensions.

Definition 3.1 (Progressing scheme). *A Mann scheme $\mathcal{S} = ((\alpha_n), (\beta_n))$ is called* progressing *if, assuming $0/0 = 0$ and $x/0 = \infty$ for $x > 0$, it holds*

$$\lim_{n\to\infty} {}^{\beta_n}\!/_{\alpha_n} = 0. \tag{5}$$

Intuitively, Condition (5) ensures that the scheme eventually prioritizes updates over dampening. Also, Condition (2) of Definition 2.1, i.e., $\lim_{n\to\infty} \beta_n = 0$, is implied by Condition (5) above, since ${}^{\beta_n}\!/_{\alpha_n} \geq \beta_n$. Note also that there is no convergence requirement on α_n and thus, in particular, α_n is allowed to tend to 0.

Canonical choices of the parameters are $\beta_n = 1/n$ and either $\alpha_n = 1$ (Kleene updates at every step) or $\alpha_n = 1/n^\varepsilon$ with $0 < \varepsilon < 1$ (decreasing learning rates).

We first show that, when the exact function f is known and can be used at each iteration, progressing schemes ensure convergence to the least fixpoint.

Theorem 3.2 (Progressing is exact). *Progressing Mann schemes are exact.*

When f can only be approximated, the following generalisation of [3, Theorem 4.4.2] shows the benefits of using decreasing learning rates.

Corollary 3.3 (Approximated function). *Let $\mathcal{S}$ be a progressing Mann scheme, $f\colon X \to X$ a monotone and non-expansive map over a 0-box $X \subseteq \mathbb{R}^d_+$, and (f_n) a sequence of maps $f_n\colon X \to X$ such that $\sum \alpha_n \|f - f_n\| < \infty$. Then the sequences (x_n) generated by $\mathcal{S}$ on (f_n) satisfy $x_n \to \mu f$.*

This significantly improves [3, Theorem 4.6] which requires $\sum \|f - f_n\| < \infty$, since when $\alpha_n \to 0$, then $\sum \alpha_n \|f - f_n\|$ is, of course, more likely to be bounded. Indeed, for any sequence (f_n) converging to f it can be shown that there is a progressing Mann scheme working for any starting point.

Example 3.4. Consider the identity $f\colon [0,1] \to [0,1]$ and a sequence of approximating functions $f_n\colon [0,1] \to [0,1]$ defined by $f_n(x) = (1 - 1/n)x + 1/n$. Then one can see that any sequence (x_n) generated, starting from $x_0 = 1$, by a relaxed Mann-Kleene scheme with dampening factors $\beta_n = 1/n$ does not converge to the least fixpoint $\mu f = 0$, for any choice of the learning rates α_n . Instead, by Corollary 3.3, for any $0 < \epsilon < 1$ the sequence obtained with learning rates $\alpha_n := 1/n^{\varepsilon}$ converges to μf.

3.2 Chaotic Dampened-Mann Iteration

We now generalize Mann schemes by allowing parameters α_n, β_n to depend not only on the step, but also on the dimensions. This establishes the basis for performing chaotic iteration to approximate the least fixpoint.

The formal switch is, again, apparently minor: dampening and learning parameters become vectors instead of scalars.

Definition 3.5 (Generalized Mann scheme). *A* generalized (dampened) Mann scheme $\mathcal{S} = ((\alpha_n), (\beta_n))$ *is a pair of parameter sequences where the* learning rates (α_n) *and* dampening factors (β_n) *are sequences of* vectors *in* $[0,1)^d$*, such that conditions (2)* $\lim_{n\to\infty} \beta_n = 0$ *and (3)* $\sum_{n\in\mathbb{N}_0} \beta_n = \infty$ *of Definition 2.1 hold pointwise.*

Notions introduced for Mann schemes naturally extend to generalized ones. The sequence generated by a generalized scheme $\mathcal{S}$ is still defined by (4), but interpreted pointwise. Similarly, the notion of *exactness* naturally extends to generalized Mann schemes.

One might think that in order to ensure convergence one could just extend the notion of progressing schemes to the generalized case by interpreting Condition (5) pointwise. The next example shows that this does not work in general.

Example 3.6. Consider $f\colon [0,1]^2 \to [0,1]^2$ with $f(x,y) = (\max(x,y), \max(x,y))$ and the (vector-)sequences $(\alpha_n), (\beta_n)$ with $\alpha_n = (1,1)$ and

$$\beta_n(1) = \begin{cases} 1/(n+2) & \text{if } n \text{ is even} \\ 1/(n+2)^2 & \text{if } n \text{ is odd} \end{cases} \qquad \beta_n(2) = \begin{cases} 1/(n+2)^2 & \text{if } n \text{ is even} \\ 1/(n+2) & \text{if } n \text{ is odd} \end{cases}$$

Clearly, $\mu f = (0,0)$, but it is easy to see that the sequence (x_n) generated by dampened Kleene Iteration with these parameters from the starting point $(1,1)$ will not converge to $(0,0)$, because the positive number $\prod_{n=2}^{\infty}(1 - 1/n^2)$ is a lower bound for at least one component of all x_n.

The crux is that we need sufficiently strong dampening, uniformly across all components. Otherwise, thinking of applications to state-based systems, it might happen that two states *convince each other* of a too large value by passing it on between each other and, at each step, one of them preserves the over-approximation while the other one gets dampened.

For this reasons, we extend the notion of progressing schemes as follows:

Definition 3.7 (Progressing generalized Mann scheme). *A generalized Mann scheme* $\mathcal{S} = ((\alpha_n), (\beta_n))$ *is* progressing *if condition below holds pointwise*

$$\lim_{n\to\infty} {}^{\beta_n}\!/_{\alpha_n} = 0 \tag{6}$$

and there exists a strictly increasing sequence (m_k) *such that, for all* $k \in \mathbb{N}_0$ *and* $i \in \{1, \dots, d\}$, $U_k(i) \coloneqq \{\ell \in \{m_k, \dots, m_{k+1} - 1\} \mid \alpha_\ell(i) > 0\} \neq \emptyset$ *and, letting* $\ell_k(i) \coloneqq \max U_k(i)$, *we have*

$$\sum_k \min\{\beta_{\ell_k(i)}(i) \mid i = 1, \dots, d\} = \infty \tag{7}$$

Intuitively, $U_k(i)$ is the set of steps in $\{m_k, \dots, m_{k+1} - 1\}$ in which component i has been updated and $\ell_k(i)$ is the last update in the interval. Hence Condition (7) requires that the dampening factors in each component decrease in some sense fairly: There is a sequence of time steps such that in between any two subsequent time steps, each component is updated at least once and, considering only the last updates in each component, the worst dampening factors still sum up to infinity. This is a natural extension of the non-generalized case: a (non-generalized) progressing scheme using the same parameter sequences in each component fulfills (7) for the sequence $m_k = k$.

Our first main result is the convergence of generalized dampened Mann iteration to the least fixpoint of f when the exact function f is known.

Main Theorem 3.8 (Exact Case). *Progressing generalized Mann schemes are exact.*

Proof (sketch). The statement is shown in two parts:

- First we show that $\limsup_{n\to\infty} x_n \leq \mu f$. This is done by comparing (x_n) to a sequence (y_n) that is generated with the same parameters but starts from the bottom element $y_n = 0$. It is clear that (y_n) always stays below the least fixpoint of f and one can show that the distance $\|x_n - y_n\|$ between the two sequences converges to zero. Here we need our additional assumption that the dampening is uniform over all components.
- In the more interesting part of the proof we show that $\limsup_{n\to\infty} x_n \geq \mu f$. To this end we construct, by transfinite induction, a sequence $(\boldsymbol{t}_\alpha)$ of *stepping stones* below μf with the property that (x_n) is eventually above $\boldsymbol{t}_\alpha$. One might hope that $(\boldsymbol{t}_\alpha)$ is monotonically increasing and converges to μf. We do not achieve monotonicity in the proof but we show that the sequence can be constructed in a way that the sequence of sums $\boldsymbol{t}_\alpha(1) + \cdots + \boldsymbol{t}_\alpha(d)$ is strictly increasing. That means, even if we loose progress towards μf in one dimension, the overall progress in the other components still makes up for it. For this we need the property that the dampening factors decrease more rapidly than the learning-rates. The fact that there is always an overall gain in the sum of the components of $\boldsymbol{t}_\alpha$ (together with the first part of the proof) allows us to conclude that the sequence (x_n) converges to μf. □

Dampened Mann iteration admits chaotic iteration as a special case.

Definition 3.9 (Chaotic iteration). *Let $\mathcal{S} = ((\alpha_n), (\beta_n))$ be a generalized Mann scheme and let (f_n) be a sequence of monotone and non-expansive maps over a 0-box $X \subseteq \mathbb{R}^d_+$. A* chaotic iteration *on $\mathcal{S}$ based on a sequence of index sets (I_n), $I_n \subseteq \{1, \dots, d\}$, is defined as below, with $x_0 \in X$ arbitrary, and*

$$\begin{cases} x_{n+1}(i) = (1 - \beta_n(i))(x_n(i) + \alpha_n(i)(f_n(x_n) - x_n)(i)) & \text{for } i \in I_n \\ x_{n+1}(j) = x_n(j) & \text{for } j \notin I_n \end{cases} \tag{8}$$

The results for generalized dampened Mann iterations immediately imply analogous results for chaotic dampened Mann iterations. Here, assuming (β_n) pointwise monotonically decreasing, the somewhat technical condition in Theorem 3.8 translates to the intuitive requirement that the time interval between two complete sweeps should not become arbitrary large.

Corollary 3.10 (Chaotic iteration). *Let $\mathcal{S} = ((\alpha_n), (\beta_n))$ be a generalized Mann scheme with (β_n) pointwise monotonically decreasing. Let (f_n) be a sequence of monotone and non-expansive maps on a 0-box $X \subseteq \mathbb{R}^d_+$, converging to f. Let (I_n) be a sequence of set of indices and inductively define (m_k) by $m_0 = 0$ and m_{k+1} the least $m \in \mathbb{N}$ such that $\bigcup_{n=m_k}^{m-1} I_n = \{1, \dots, d\}$. If*

$$\sum_{k \in \mathbb{N}} \min_i \beta_{m_k}(i) = \infty \tag{9}$$

then the sequences (x_n) generated by chaotic iteration converge to μf.

Given a (non-generalized) Mann scheme, a simple way of applying the scheme in a chaotic fashion consists in associating to each component a local copy of the parameters and then updating at each step a single (independently) randomly chosen state, letting the parameters progress to the next value only for this state. It can be proved that, in this way, starting from a progressing scheme, the generated sequence converges almost surely to the fixpoint.This will be used in the numerical experiments in §5.

3.3 Working with Approximations

Theorem 3.8 assumes that the monotone and non-expansive function f of interest is known exactly. The benefits of the (generalized) dampened Mann iteration, however, lie in the case where f can only be approximated by a sequence (f_n) of monotone and non-expansive approximations.

The main result of this section is the following result that gives an essential sufficient criterion for convergence.

Main Theorem 3.11 (Approximated Case). *Let $\mathcal{S} = ((\alpha_n), (\beta_n))$ be a progressing Mann scheme, $X \subseteq \mathbb{R}^d_+$ a 0-box, and let (f_n) be a sequence of monotone and non-expansive functions $f_n \colon X \to X$, pointwise converging to $f \colon X \to X$. Then, the sequence (x_n) generated by $\mathcal{S}$ fulfills*

1. $\liminf_{n\to\infty} x_n \geq \mu f$
2. *if* $\lim_{n\to\infty} \mu(\sup_{k\geq n} f_k) = \mu f$*, then* $\lim_{n\to\infty} x_n = \mu f$

This result, in particular, implies convergence to the least fixpoint if (f_n) converges monotonically and $\mu f_n \to \mu f$, thus generalizing [3, Theorem 4.6.1]. Instead, in general, the fact $\mu f_n \to \mu f$ is not sufficient to have convergence, as shown by the example below, inspired by [3, Example 4.4].

Example 3.12. Consider the function $f(x, y) = (y, x)$ and the sequence of approximations (f_n) given by

$$f_n(x,y) = \begin{cases} (y,x) & \text{if } n \text{ even} \\ (y \ominus 2/n, x \oplus 2/n) & \text{if } n \text{ uneven} \end{cases}$$

Despite μf_n converging to $\mu f = (0, 0)$, it can be seen that dampened Mann iteration does not converge in general. Indeed, the hypotheses of Theorem 3.11(2) are violated, since the supremum $F_n = \sup_{k\geq n} f_k$ is always of the form $F_n(x, y) = (y, x + \varepsilon)$ for some $\varepsilon > 0$, thus having fixpoint $\mu F_n = (\infty, \infty) \nrightarrow (0, 0)$.

By relying on Theorem 3.11 one can also prove convergence in the approximated case when the limit function f is sufficiently well-behaved (without assumptions on the mode of convergence).

Theorem 3.13 (Convergence for Picard limit functions). *Let (f_n) be a sequence of monotone and non-expansive functions $f_n \colon X \to X$ with X a 0-box, converging pointwise to a Picard operator $f \colon X \to X$. Then, we have*

$$\lim_{n\to\infty} \mu(\sup_{k\geq n} f_k) = \mu f.$$

As a consequence, for any progressing Mann scheme $\mathcal{S}$, the sequence (x_n) generated by $\mathcal{S}$ on (f_n) converges to μf.

Note that this result implies convergence, in particular, for cases where the limit function is contractive or power contractive, thus generalizing both [4, Theorem 4.3 and Theorem B.3].

4 Applications to State-based Systems

We treat the convergence of dampened Mann iterations for piecewise linear functions such as those arising when dealing with optimal returns and optimal policies in zero-sum two-player (turn-based) stochastic games. These results generalize those in [3] which are limited to MDPs, with proofs relying on techniques specifically designed for this setting, like collapsing end-components, making it difficult to generalize them to other function classes. Furthermore, we lift the requirement that the least fixpoint, i.e., the expected return is finite for all states. Finally, in [3], we showed convergence only for Mann-Kleene schemes, while here we consider generalized Mann schemes which allow also for asynchronous iteration.

4.1 Simple Stochastic Games

We will be working with the following definition of simple stochastic games.

Definition 4.1 (Simple stochastic game). *A zero-sum two-player (turn-based) stochastic game $\mathcal{G}$ (short: simple stochastic game or SSG) is a tuple $(S, A, T, R, S_{\max}, S_{\min})$ where*

- *S is a (finite) set of states*
- *$(A(s))_{s\in S}$ is a family of (finite) sets of actions $A(s)$ enabled in state s, and, by abuse of notation, we write $A \coloneqq \bigcup_{s\in S} A(s)$;*
- *$T\colon S \times A \times S \to [0,1], (s,a,s') \mapsto T_a(s,s')$ is a transition function fulfilling;*

$$T_a(s) \coloneqq \sum_{s'\in S} T_a(s,s') \in [0,1] \quad \text{for all } s \in S, a \in A$$

- *$R\colon S \times A \to \mathbb{R}_+, (s,a) \mapsto R_a(s)$ is a reward function;*
- *$S_{\max} \cup S_{\min} = S$ are a partition of S where $S_{\max}$ and $S_{\min}$ are states controlled by the* max- *and* min*-player, respectively.*

We write $F = F(\mathcal{G}) \coloneqq \{s \in S \mid A(s) = \emptyset\}$ for the set of final states.

Intuitively, an SSG is a probabilistic state-based system where, in a given non-final state $s \in S \setminus F$, one of two external agents (the max-player if $s \in S_{\max}$ or min-player if $s \in S_{\min}$) chooses one of the actions $a \in A(s)$ enabled in said state after which the system emits a reward $R_a(s)$ and transitions to a new state $s' \in S$ with probability $T_a(s,s')$ (or terminates with probability $1 - T_a(s)$). The goal of the two players is to choose actions in order to maximize (max-player) or minimize (min-player) the total expected reward.

As we allow $T_a(s) < 1$, this definition also encompasses systems with a discount factor. Furthermore, it comprises as special cases also Markov chains (where $|A(s)| \leq 1$ for all $s \in S$), (maximizing) Markov decision processes (MDPs; where $|A(s)| \leq 1$ for all $s \in S_{\min}$), and minimizing Markov decision processes (Min-MDPs; where $|A(s)| \leq 1$ for all $s \in S_{\max}$).

Policies, as defined below, resolve the non-determinism present in SSGs by fixing the agents' behaviour.

Definition 4.2 (Policy). *Let $(S, A, T, R, S_{\max}, S_{\min})$ be a SSG. A (memoryless and deterministic) policy on a set $S' \subseteq S \setminus F$ is a map $\pi\colon S' \to A$ with $\pi(s) \in A(s)$ for $s \in S'$. Given an SSG $\mathcal{G} = (S, A, T, R, S_{\max}, S_{\min})$ and a policy $\pi\colon S' \to A$, we write $\mathcal{G}^\pi = (S, A^\pi, T, R, S_{\max}, S_{\min})$ for the SSG where, for $s \in S$*

$$A^\pi(s) = \begin{cases} \{\pi(s)\} & \text{if } s \in S' \\ A(s) & \text{otherwise} \end{cases}$$

Note that, when $\pi_{\min}\colon S_{\min} \setminus F \to A$ and $\pi_{\max}\colon S_{\max} \setminus F \to A$ are policies of the min- and max-player, respectively, we have that $\mathcal{G}^{\pi_{\min}}$ is a (Max-)MDP, $\mathcal{G}^{\pi_{\max}}$ is a Min-MDP, and $\mathcal{G}^{(\pi_{\min},\pi_{\max})}$ is a Markov chain (where $(\pi_{\min}, \pi_{\max})$ denotes the combined policy for both players).

The expected total reward for each starting state s in $\mathcal{G}$, assuming that both players act optimally, can be characterized as the least fixpoint (in $\overline{\mathbb{R}}_+^S$) of the so-called Bellman operator $f_{\mathcal{G}} \colon \mathbb{R}_+^S \to \mathbb{R}_+^S$ given by

$$f_{\mathcal{G}}(v)(s) = \begin{cases} \max_{a \in A(s)} (R_a(s) + \sum_{s' \in S} T_a(s,s') v(s')) & \text{if } s \in S_{\max} \\ \min_{a \in A(s)} (R_a(s) + \sum_{s' \in S} T_a(s,s') v(s')) & \text{if } s \in S_{\min} \end{cases}$$

Note that the expected total reward might be infinite.

For finite-state SSGs, it is known that memoryless policies such as the ones described above are sufficient to act optimally (see, e.g., [8]) whence we have

$$\mu f_{\mathcal{G}} = \mu f_{\mathcal{G}^{\pi^*_{\min}}} = \mu f_{\mathcal{G}^{\pi^*_{\max}}} = \mu f_{\mathcal{G}^{\pi^*_{\max}, \pi^*_{\min}}} \tag{10}$$

for some (optimal) policies $\pi_{\min} \colon S_{\min} \setminus F \to A$ and $\pi_{\max} \colon S_{\max} \setminus F \to A$.

Before we give convergence proofs, we end this subsection by giving a definition of a sampled SSG:

Definition 4.3 (Sampling). *Given an SSG $\mathcal{G} = (S, A, T, R, S_{\max}, S_{\min})$, a sequence $(\mathcal{G}_n)$ of SSGs $\mathcal{G}_n = (S, A, T_n, R_n, S_{\max}, S_{\min})$ is a* sampling *of $\mathcal{G}$ if*

- *$T_n \to T$ and $R_n \to R$ for $n \to \infty$.*
- *For $s, s' \in S, a \in A$, if $T_a(s,s') = 0$, then $(T_n)_a(s,s') = 0$ for all $n \in \mathbb{N}$, i.e., a non-existing transition can never be sampled.*
- *For $s \in S, a \in A$, if $R_a(s) = 0$, then $(R_n)_a(s) = 0$ for all $n \in \mathbb{N}$, i.e., a non-existing reward can never be sampled.*
- *For $s \in S, a \in A$, if $T_a(s) = 1$, then $(T_n)_a(s) = 1$ for all $n \in \mathbb{N}$, i.e., a non-terminating action can never be sampled as terminating.*

4.2 Convergence for Simple Stochastic Games

The proof of convergence of the dampened Mann iteration for SSGs will mainly be based on the technical convergence criterion given by Theorem 3.11.

To be precise, we will show that the sequences of Bellman operators (f_n) of sampled SSGs fulfill the condition

$$\lim_{n \to \infty} \mu(\sup_{k \geq n} f_k) = \mu(\lim_{n \to \infty} f_n). \tag{11}$$

This will be done by first reducing the problem to MDPs which, in turn, depends on results for Markov chains, using (10). In fact, we use the following result for the optimal values of Markov chains.

Lemma 4.4. *Given a sampling $\mathcal{M} = (M_n)$ of a Markov chain M, we have $\mu f_M = \lim_{n \to \infty} \mu f_{M_n}$.*

We proved such a result already in [3], yet some extra work has to be done to adapt to the new setting, allowing for approximated rewards and infinite values. The idea is that states with zero or infinite value are determined by the

underlying structure of the Markov chain (in terms of Definition 4.3), whereas on the 'non-trivial' states, the Bellman operator enjoys a power contraction property ensuring convergence to the (unique) fixpoint.

Using this result for Markov chains, it is easy to generalize to SSGs using (10).

Corollary 4.5 (Sampling-continuity of least-fixpoint operator). *Given a sampling $\mathcal{G} = (G_n)$ of an SSG G, we have $\mu f_G = \lim_{n\to\infty} \mu f_{G_n}$.*

We continue by showing Condition (11) for MDPs and derive the following.

Theorem 4.6 (Convergence for MDPs). *Given a sampling $\mathcal{M} = (M_n)$ of an MDP M, we have $\lim_{n\to\infty} \mu(\sup_{k\geq n} f_{M_k}) = \mu f_M$. Therefore for any progressing Mann scheme $\mathcal{S}$, the sequence (x_n) generated by $\mathcal{S}$ on (f_{M_n}) converges to μf_M.*

Proof (sketch). The main idea used to prove this statement is that the supremum of Bellman operators of arbitrary many MDPs of the same underlying structure (in the sense of Definition 4.3) is itself a Bellman operator of an MDP with the 'same' structure (although with potentially many more actions) as long as all transition probabilities and rewards are close enough. Using this, Condition (11) can be shown to break down to continuity of the least fixpoint operator for a 'sampling' sequence. □

Using the validity of Condition (11) for MDPs, we can derive the same result for SSGs using (10) and an optimal min-player strategy.

Corollary 4.7. *Given a sampling $\mathcal{G} = (G_n)$ of an SSG G, we have that $\lim_{n\to\infty} \mu(\sup_{k\geq n} f_{G_k}) = \mu f_G$. As a consequence, for any progressing Mann scheme $\mathcal{S}$, the sequence (x_n) generated by $\mathcal{S}$ on (f_{G_n}) converges to μf_G.*

Remark 4.8. This result can be used to show convergence for a wider range of 'sampled' piecewise linear functions by making use of the fact that convergence of progressing generalized schemes for a sampled SSG (f_{M_n}) directly implies convergence for the sequences $(f^k_{M_n})_n$ for any fixed $k > 1$ (k-step Bellman operators). The result can also be used to obtain a direct proof of convergence for state-action-value Bellman operators, where rewards are given to transitions.

5 Numerical Experiments

In this section we illustrate a number of experiments comparing the convergence behaviour of (relaxed) Mann-Kleene schemes as presented in [3] with schemes working with vanishing or divergent learning rates as well as chaotic schemes. The aim of these experiments is to provide some insights on how the generalisations to dampened Mann iteration in the paper affect the convergence speed when using a randomized strategy for chaotic updates. In particular, we cannot expect an improvement in efficiency that, instead, could potentially be obtained using more informed heuristics for choosing which components to update.

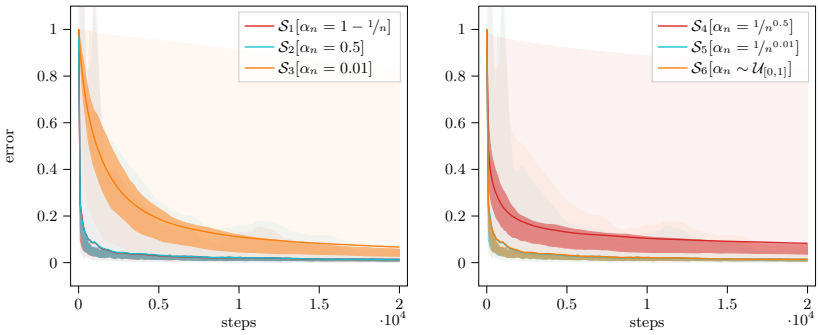

Fig. 1: Results for the experiments of non-chaotic iterations on 50 randomly generated SSGs. Lines denote the mean error, the half-opaque area the 25th to 75th percentile, the weakly visible area the minimum to maximum area. Results for $\mathcal{S}_1$ and $\mathcal{S}_4$ are only barely visible as they are almost identical to the ones from $\mathcal{S}_2$ and $\mathcal{S}_6$, respectively.

We tested all iteration schemes on the same set of 50 randomly generated SSGs G with 15 min- and 15 max-player states, at most 5 actions per state, each of them with a normalized value $\|\mu f_G\| = 1$ and for which the Kleene iteration on f_G 'converges' in at most 10000 steps (we assume convergence when changes are below 10^{-8}).

Vanishing and diverging learning rates. To test the influence of vanishing or diverging learning rates on the convergence rate, we compared the performance of the six iteration schemes shown in Table 1. The scheme $\mathcal{S}_1$ is a Mann-Kleene scheme, while $\mathcal{S}_2$ and $\mathcal{S}_3$ are relaxed Mann-Kleene schemes. Instead $\mathcal{S}_4$ and $\mathcal{S}_5$ have vanishing learning rate and $\mathcal{S}_6$ chooses a random learning rate at each step independently w.r.t. a uniform distribution, and thus the sequence of learning rates is almost surely non-converging. Note that convergence for the first three schemes (at least for MDPs) is covered by the results in [3] while the last three require the results in the present paper.

In the experiments, before each Mann iteration step, the SSGs $\mathcal{G}$ are sampled by performing 30 model steps for randomly chosen state-action pairs.

$\mathcal{S}$	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$	$\mathcal{S}_6$
α_n	$1 - 1/n$	0.5	0.01	$1/n^{0.5}$	$1/n^{0.01}$	$\sim \mathcal{U}_{[0,1]}$
β_n	$1/n$	$1/n$	$1/n$	$1/n$	$1/n$	$1/n$

Table 1: Dampened Mann Schemes used in the experiments.

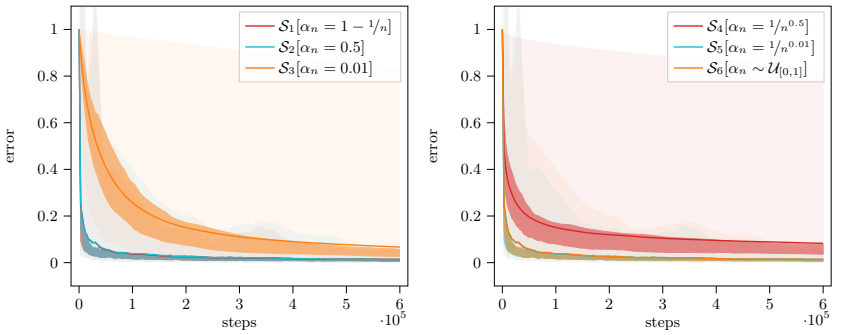

Fig. 2: Results for the experiments of chaotic iterations on 50 randomly generated SSGs. Again, the results for $\mathcal{S}_1$ and $\mathcal{S}_4$ are only barely visible as they are almost identical to the ones from $\mathcal{S}_2$ and $\mathcal{S}_6$, respectively.

The results of these experiments are in Figure 1. Observe that both vanishing and random (diverging) learning rates yield similar results as (relaxed) Mann-Kleene parameters.

Also, the schemes that keep having a higher learning rate (for longer) perform better (higher constant learning rates or lower exponent α for learning rates of the form $1/n^{\alpha}$). This is expected since we are dealing with faultless sampling of SSGs and the advantage of lower learning rates does not lie in faster convergence but in higher robustness to 'irregular' approximations (see Example 3.4).

Chaotic iteration. To test the influence of chaotic updates on the convergence rate, we repeated the experiments with chaotic variants of the above iteration schemes. As outline at the end of §3.2, at each step, the chaotic iteration performs an update for one single (independently) randomly chosen state, updating the learning rate (and dampening factor) only for this state. Thus, to have a 'full sweep' over the states, the chaotic variants need (at least) 30 steps (while the non-chaotic variants above perform 30 updates in one step). To have a fair comparison to the non-chaotic variants, the sampling $\mathcal{G}$ only simulates a single model step for a randomly chosen state-action pair before each iteration step. In total, after 30 steps, the chaotic schemes thus have the same information and the same number of updates (thus, the same amount of computation) as a single step of the non-chaotic variants, while allowing for intermediate results.

The results of these experiments are in Figure 2. It can be seen directly that the schemes all have almost the same convergence speed as the non-chaotic variants (keeping in mind that the non-chaotic variants perform 30 updates per step), even if we use a random non-informed choice of components to update in each step. As a consequence, the experiments suggest that performing a chaotic iteration (randomly) comes at no additional cost while allowing for intermediate results even when the computation with non-chaotic variants would be infeasible. It should further be noted that, in most practical use cases, the components to

update might be chosen in a more informed manner, e.g., by taking into account which components had been updated, which might result in faster convergence.

6 Conclusion

Mann iteration is a classical technique for generating a sequence which, under suitable hypotheses, converges to a fixpoint of a continuous function starting from any initial state [6]. Dampened versions are considered in [14,23] where convergence to a fixpoint is shown in a uniformly smooth Banach space and Hilbert spaces, under suitable assumptions on the parameter sequences. In [3], we proposed the use of dampened Mann iteration for dealing with functions which can only be approximated. This work sensibly deviates from [14,23] as the intended applications, for approximating fixpoints of functions arising from quantitative models like MDPs and SSGs, lead to focus on the case of monotone non-expansive functions in (finite-dimensional) Banach spaces with the supremum norm, which are neither uniformly smooth nor Hilbert spaces.

Our paper takes the same setting as [3] and generalizes its results. The generalisations include the possibility of having learning rates converging to 0 or not converge at all, the possibility of varying parameters across dimensions, thus enabling chaotic iteration and the proof of convergence of these schemes for simple stochastic games approximated via sampling.

We provided some numerical results, showing that the chaotic iteration (with random choice of updated component) yields almost identical results in terms of convergence of standard dampened Mann iteration, while giving more flexibility, e.g., opening the way to parallel and distributed implementations. We plan a further exploration of effective chaotic iteration strategies, including data-driven approaches to select the components to update at each step based on statistical properties of the model, possibly tuning the learning rate and dampening parameters based on observed convergence behaviour during execution.

We furthermore plan to investigate applications of our results to the computation of behavioural metrics, in particular probabilistic bisimilarity distances. In fact, [2] shows how to reduce this problem to the solution of a simple stochastic game. Computing behavioural metrics in the approximated setting is also discussed [13,10] where the authors observe that a small perturbation of the transition probabilities can drastically change the distance apparently hindering the possibility of properly approximating the distance when such probabilities can be only estimated. Our results suggest that, instead, one can use dampened Mann iteration for obtaining increasingly better approximations of the distance.

We are also interested in considering the problem of approximating the least fixpoint in scenarios, where the functions do not necessarily converge, but converge in the limit-average, providing a close match with reinforcement learning algorithms such as Q-learning [22]. For this one might be able to draw inspiration from theory of stochastic approximation [7,9,15], going back to [16] that deals with the case where a function contains an error term with expected value zero.

References

1. Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. On-the-fly exact computation of bisimilarity distances. *Logical Methods in Computer Science*, 13(2:13):1–25, 2017.
2. Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, Radu Mardare, Qiyi Tang, and Franck van Breugel. Computing probabilistic bisimilarity distances for probabilistic automata. *Logical Methods in Computer Science*, 17(1), 2021.
3. Paolo Baldan, Sebastian Gurke, Barbara König, Tommaso Padoan, and Florian Wittbold. Approximating fixpoints of approximated functions. In Ruzica Piskac and Zvonimir Rakamaric, editors, *Proc. of CAV 2025 – Part II*, pages 193–215. Springer, 2025. LNCS 15932.
4. Paolo Baldan, Sebastian Gurke, Barbara König, Tommaso Padoan, and Florian Wittbold. Approximating fixpoints of approximated functions, 2025. arXiv:2501.08950.
5. Paolo Baldan, Sebastian Gurke, Barbara König, and Florian Wittbold. Computing fixpoints of learned functions: Chaotic iteration and simple stochastic games, 2026. arXiv:2601.16142.
6. Vasile Berinde. *Iterative Approximation of Fixed Points*, volume 1912 of *Lecture Notes in Mathematics*. Springer, 2007.
7. Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, Massachusetts, 1996.
8. Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.
9. Marie Duflo. *Random Iterative Models*. Number 34 in Applications of Mathematics – Stochastic Modelling and Applied Probability. Springer, 1991. Translated by Stephen S. Wilson.
10. Syyeda Zainab Fatmi, Stefan Kiefer, David Parker, and Franck van Breugel. Robust probabilistic bisimilarity for labelled markov chains. In Ruzica Piskac and Zvonimir Rakamarić, editors, *Proceedings of CAV'2025*, pages 254–275. Springer, 2025.
11. Andreas Frommer and Daniel B. Szyld. On asynchronous iterations. *Journal of Computational and Applied Mathematics*, 123(1):201–216, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.
12. Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
13. Stefan Kiefer and Qiyi Tang. Approximate bisimulation minimisation. In *Proc. of FSTTCS '21*, volume 213 of *LIPIcs*, pages 48:1–48:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.
14. Tae-Hwa Kim and Hong-Kun Xu. Strong convergence of modified Mann iterations. *Nonlinear Analysis: Theory, Methods & Applications*, 61(1):51–60, 2005.
15. Harold J. Kushner and Dean S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Number 26 in Applied Mathematical Sciences. Springer, 1978.
16. H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.
17. Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.
18. Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bulletin*, 2(4):160–163, 1991.

19. Richard S. Sutton. Planning by incremental dynamic programming. In *Proc. of ML '91 (Conference on Machine Learning)*, pages 353–357. Morgan Kaufmann, 1991.
20. Franck van Breugel. Probabilistic bisimilarity distances. *ACM SIGLOG News*, 4(4):33–51, 2017.
21. Franck van Breugel and James Worrell. The complexity of computing a bisimilarity pseudometric on probabilistic automata. In *Horizons of the Mind. A Tribute to Prakash Panangaden – Essays Dedicated to Prakash Panangaden on the Occasion of His 60th Birthday*, pages 191–213. Springer, 2014. LNCS 8464.
22. Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.
23. Yonghong Yao, Haiyun Zhou, and Yeong-Cheng Liou. Strong convergence of a modified Krasnoselski-Mann iterative algorithm for non-expansive mappings. *Journal of Applied Mathematics and Computing*, 29:383–389, 2008.