Approximative Methods for Monotone Systems of min-max-Polynomial Equations*

Javier Esparza, Thomas Gawlitza, Stefan Kiefer, and Helmut Seidl

Institut für Informatik
Technische Universität München, Germany
{esparza,gawlitza,kiefer,seidl}@in.tum.de

Abstract. A *monotone system of min-max-polynomial equations* (min-max-MSPE) over the variables $X_1, \ldots, X_n$ has for every i exactly one equation of the form $X_i = f_i(X_1, \ldots, X_n)$ where each $f_i(X_1, \ldots, X_n)$ is an expression built up from polynomials with non-negative coefficients, minimum- and maximum-operators. The question of computing least solutions of min-max-MSPEs arises naturally in the analysis of *recursive stochastic games* [5, 6, 14]. Min-max-MSPEs generalize MSPEs for which convergence speed results of Newton's method are established in [11, 3]. We present the first methods for approximatively computing least solutions of min-max-MSPEs which converge at least linearly. Whereas the first one converges faster, a single step of the second method is cheaper. Furthermore, we compute ϵ-optimal positional strategies for the player who wants to maximize the outcome in a recursive stochastic game.

1 Introduction

In this paper we study *monotone systems of min-max polynomial equations* (min-max-MSPEs). A min-max-MSPE over the variables $X_1, \ldots, X_n$ contains for every $1 \leq i \leq n$ exactly one equation of the form $X_i = f_i(X_1, \ldots, X_n)$ where every $f_i(X_1, \ldots, X_n)$ is an expression built up from polynomials with non-negative coefficients, minimum- and maximum-operators. An example of such an equation is $X_1 = 3X_1X_2 + 5X_1^2 \wedge 4X_2$ (where $\wedge$ is the minimum-operator). The variables range over non-negative reals. Min-max-MSPEs are called monotone because f_i is a monotone function in all arguments.

Min-max-MSPEs naturally appear in the study of two-player stochastic games and competitive Markov decision processes, in which, broadly speaking, the next move is decided by one of the two players or by tossing a coin, depending on the game's position (see e.g. [12, 7]). The min and max operators model the competition between the players. The product operator, which leads to non-linear equations, allows to deal with recursive stochastic games [5, 6], a class of games with an infinite number of positions, and having as special case *extinction games*, games in which players influence with their actions the development of a population whose members reproduce and die, and the player's goals are to extinguish the population or keep it alive (see Section 3).

Min-max-MSPEs generalize several other classes of equation systems. If product is disallowed, we obtain systems of min-max *linear* equations, which appear in classical

* This work was in part supported by the DFG project *Algorithms for Software Model Checking*.

two-person stochastic games with a finite number of game positions. The problem of solving these systems has been thoroughly studied [1, 8, 9]. If both $\min$ and $\max$ are disallowed, we obtain monotone systems of polynomial equations, which are central to the study of recursive Markov chains and probabilistic pushdown systems, and have been recently studied in [4, 11, 3]. If only one of $\min$ or $\max$ is disallowed, we obtain a class of systems corresponding to recursive Markov decision processes [5]. All these models have applications in the analysis of probabilistic programs with procedures [14].

In vector form we denote a min-max-MSPE by $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ where $\boldsymbol{X}$ denotes the vector $(X_1, \ldots, X_n)$ and $\boldsymbol{f}$ denotes the vector $(f_1, \ldots, f_n)$. By Kleene's theorem, if a min-max-MSPE has a solution then it also has a *least* one, denoted by $\mu\boldsymbol{f}$, which is also the relevant solution for the applications mentioned above. Kleene's theorem also ensures that the iterative process $\boldsymbol{\kappa}^{(0)} = \mathbf{0},\ \boldsymbol{\kappa}^{(k+1)} = \boldsymbol{f}(\boldsymbol{\kappa}^{(k)}),\ k \in \mathbb{N}$, the so-called Kleene sequence, converges to $\mu\boldsymbol{f}$. However, this procedure can converge very slowly: in the worst case, the number of accurate bits of the approximation grows with the *logarithm* of the number of iterations (cf. [4]). Thus, the goal is to replace the function $\boldsymbol{f}$ by an operator $G : \mathbb{R}^n \to \mathbb{R}^n$ such that the respective iterative process also converges to $\mu\boldsymbol{f}$ but faster. In [4, 11, 3] this problem was studied for min-max-MSPEs without the $\min$ and $\max$ operator. There, G was chosen as one step of the well-known Newton's method (cf. for instance [13]). This means that, for a given approximate $\boldsymbol{x}^{(k)}$, the next approximate $\boldsymbol{x}^{(k+1)} = G(\boldsymbol{x}^{(k)})$ is determined by the *unique solution* of a linear equation system which is obtained from the first order Taylor approximation of $\boldsymbol{f}$ at $\boldsymbol{x}^{(k)}$. It was shown that this choice guarantees *linear convergence*, i.e., the number of accurate bits grows *linearly* in the number of iterations. Notice that when characterizing the convergence behavior the term linear does not refer to the size of $\boldsymbol{f}$.

However, this technique no longer works for arbitrary min-max-MSPEs. If we approximate $\boldsymbol{f}$ at $\boldsymbol{x}^{(k)}$ through its first order Taylor approximation at $\boldsymbol{x}^{(k)}$ there is no guarantee that the next approximate still lies below the least solution, and the sequence of approximants may even diverge. For this reason, the PReMo tool [14] uses round-robin iteration for min-max-MSPEs, an optimization of Kleene iteration. Unfortunately, this technique also exhibits "logarithmic" convergence behavior in the worst case.

In this paper we overcome the problem of Newton's method. Instead of approximating $\boldsymbol{f}$ (at the current approximate $\boldsymbol{x}^{(k)}$) by a linear function, both of our methods approximate $\boldsymbol{f}$ by a *piecewise* linear function. In contrast to the applications of Newton's method in [4, 11, 3], this approximation may not have a *unique fixpoint*, but it has a *least fixpoint* which we use as the next approximate $\boldsymbol{x}^{(k+1)} = G(\boldsymbol{x}^{(k)})$. Our first method uses an approximation of $\boldsymbol{f}$ at $\boldsymbol{x}^{(k)}$ whose least fixpoint can be determined using the algorithm for *systems of rational equations* from [9]. The approximation of $\boldsymbol{f}$ at $\boldsymbol{x}^{(k)}$ used by our second method allows to use linear programming to compute $\boldsymbol{x}^{(k+1)}$. Our methods are the first algorithms for approximatively computing $\mu\boldsymbol{f}$ which converge at least linearly, provided that $\boldsymbol{f}$ is quadratic, an easily achievable normal form.

The rest of the paper is organized as follows. In Section 2 we introduce basic concepts and state some important facts about min-max-MSPEs. A class of games which can be analyzed using our techniques is presented in Section 3. Our main contribution, the two approximation methods, is presented and analyzed in Sections 4 and 5. In Sec-

tion 6 we study the relation between our two approaches and compare them to previous work. We conclude in Section 7. Missing proofs can be found in a technical report [2].

2 Notations, Basic Concepts and a Fundamental Theorem

As usual, $\mathbb{R}$ and $\mathbb{N}$ denote the set of real and natural numbers. We assume $0 \in \mathbb{N}$. We write $\mathbb{R}_{\geq 0}$ for the set of non-negative real numbers. We use bold letters for vectors, e.g. $\boldsymbol{x} \in \mathbb{R}^n$. In particular $\mathbf{0}$ denotes the vector $(0, \ldots, 0)$. The transpose of a matrix or a vector is indicated by the superscript $^\top$. We assume that the vector $\boldsymbol{x} \in \mathbb{R}^n$ has the components $x_1, \ldots, x_n$. Similarly, the i-th component of a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ is denoted by f_i. As in [3], we say that $\boldsymbol{x} \in \mathbb{R}^n$ has $i \in \mathbb{N}$ valid bits of $\boldsymbol{y} \in \mathbb{R}^n$ iff $|x_j - y_j| \leq 2^{-i}|y_j|$ for $j = 1, \ldots, n$. We identify a linear function from $\mathbb{R}^n$ to $\mathbb{R}^m$ with its representation as a matrix from $\mathbb{R}^{m \times n}$. The identity matrix is denoted by I. The *Jacobian* of a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ at $\boldsymbol{x} \in \mathbb{R}^n$ is the matrix of all first-order partial derivatives of $\boldsymbol{f}$ at $\boldsymbol{x}$, i.e., the $m \times n$-matrix with the entry $\frac{\partial f_i}{\partial X_j}(\boldsymbol{x})$ in the i-th row and the j-th column. We denote it by $\boldsymbol{f}'(\boldsymbol{x})$.

The partial order $\leq$ on $\mathbb{R}^n$ is defined by setting $\boldsymbol{x} \leq \boldsymbol{y}$ iff $x_i \leq y_i$ for all $i = 1, \ldots, n$. We write $\boldsymbol{x} < \boldsymbol{y}$ iff $\boldsymbol{x} \leq \boldsymbol{y}$ and $\boldsymbol{x} \neq \boldsymbol{y}$. The operators $\wedge$ and $\vee$ are defined by $x \wedge y := \min\{x, y\}$ and $x \vee y := \max\{x, y\}$ for $x, y \in \mathbb{R}$. These operators are also extended component-wise to $\mathbb{R}^n$ and point-wise to $\mathbb{R}^n$-valued functions. A function $\boldsymbol{f} : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ it called *monotone* on $M \subseteq D$ iff $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{y})$ for every $\boldsymbol{x}, \boldsymbol{y} \in M$ with $\boldsymbol{x} \leq \boldsymbol{y}$. Let $X \subseteq \mathbb{R}^n$ and $\boldsymbol{f} : X \to X$. A vector $\boldsymbol{x} \in X$ is called *fixpoint* of $\boldsymbol{f}$ iff $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$. It is the *least fixpoint* of $\boldsymbol{f}$ iff $\boldsymbol{y} \geq \boldsymbol{x}$ for every fixpoint $\boldsymbol{y} \in X$ of $\boldsymbol{f}$. If it exists we denote the least fixpoint of $\boldsymbol{f}$ by $\mu\boldsymbol{f}$. We call $\boldsymbol{f}$ *feasible* iff $\boldsymbol{f}$ has some fixpoint $\boldsymbol{x} \in X$.

Let us fix a set $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ of variables. We call a vector $\boldsymbol{f} = (f_1, \ldots, f_m)$ of polynomials $f_1, \ldots, f_m$ in the variables $X_1, \ldots, X_n$ a *system of polynomials*. $\boldsymbol{f}$ is called *linear* (resp. *quadratic*) iff the degree of each f_i is at most 1 (resp. 2), i.e., every monomial contains at most one variable (resp. two variables). As usual, we identify $\boldsymbol{f}$ with its interpretation as a function from $\mathbb{R}^n$ to $\mathbb{R}^m$. As in [11, 3] we call $\boldsymbol{f}$ a *monotone system of polynomials* (MSP for short) iff all coefficients are non-negative.

Min-max-MSPs. Given polynomials $f_1, \ldots, f_k$ we call $f_1 \wedge \cdots \wedge f_k$ a *min-polynomial* and $f_1 \vee \cdots \vee f_k$ a *max-polynomial*. A function that is either a min- or a max-polynomial is also called *min-max-polynomial*. We call $\boldsymbol{f} = (f_1, \ldots, f_n)$ a *system of min-polynomials* iff every component f_i is a min-polynomial. The definition of *systems of max-polynomials* and *systems of min-max-polynomials* is analogous. A system of min-max-polynomials is called *linear* (resp. *quadratic*) iff all occurring polynomials are linear (resp. *quadratic*). By introducing auxiliary variables every system of min-max-polymials can be transformed into a *quadratic* one in time linear in the size of the system (cf. [11]). A system of min-max-polynomials where all coefficients are from $\mathbb{R}^n_{\geq 0}$ is called a *monotone system of min-max-polynomials* (*min-max-MSP*) for short. The terms *min-MSP* and *max-MSP* are defined analogously.

Example 1. $\boldsymbol{f}(x_1, x_2) = (\frac{1}{2}x_2^2 + \frac{1}{2} \wedge 3, x_1 \vee 2)^\top$ is a quadratic min-max-MSP.

A min-max-MSP $\boldsymbol{f} = (f_1, \ldots, f_n)^\top$ can be considered as a mapping from $\mathbb{R}^n_{\geq 0}$ to $\mathbb{R}^n_{\geq 0}$. The Kleene sequence $(\boldsymbol{\kappa}^{(k)}_{\boldsymbol{f}})_{k\in\mathbb{N}}$ is defined by $\boldsymbol{\kappa}^{(k)}_{\boldsymbol{f}} := \boldsymbol{f}^k(\mathbf{0})$, $k \in \mathbb{N}$. We have:

Lemma 1. *Let $\boldsymbol{f} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ be a min-max-MSP. Then: (1) $\boldsymbol{f}$ is monotone and continuous on $\mathbb{R}^n_{\geq 0}$; and (2) If $\boldsymbol{f}$ is feasible (i.e., $\boldsymbol{f}$ has some fixpoint), then $\boldsymbol{f}$ has a least fixpoint $\mu\boldsymbol{f}$ and $\mu\boldsymbol{f} = \lim_{k\to\infty} \boldsymbol{\kappa}^{(k)}_{\boldsymbol{f}}$.*

Strategies. Assume that $\boldsymbol{f}$ denotes a system of min-max-polynomials. A $\vee$-*strategy* σ for $\boldsymbol{f}$ is a function that maps every max-polynomial $f_i = f_{i,1} \vee \cdots \vee f_{i,k_i}$ occurring in $\boldsymbol{f}$ to one of the $f_{i,j}$'s and every min-polynomial f_i to f_i. We also write f_i^σ for $\sigma(f_i)$. Accordingly, a $\wedge$-*strategy* π for $\boldsymbol{f}$ is a function that maps every min-polynomial $f_i = f_{i,1} \wedge \cdots \wedge f_{i,k}$ occurring in $\boldsymbol{f}$ to one of the $f_{i,j}$'s and every max-polynomial f_i to f_i. We denote the set of $\vee$-strategies for $\boldsymbol{f}$ by $\Sigma_{\boldsymbol{f}}$ and the set of $\wedge$-strategies for $\boldsymbol{f}$ by $\Pi_{\boldsymbol{f}}$. For $s \in \Sigma_{\boldsymbol{f}} \cup \Pi_{\boldsymbol{f}}$, we write $\boldsymbol{f}^s$ for $(\boldsymbol{f}^s_1, \ldots, \boldsymbol{f}^s_n)^\top$. We define $\Pi^*_{\boldsymbol{f}} := \{\pi \in \Pi_{\boldsymbol{f}} \mid \boldsymbol{f}^\pi \text{ is feasible}\}$. We drop the subscript whenever it is clear from the context.

Example 2. Consider $\boldsymbol{f}$ from Example 1. Then $\pi : \frac{1}{2}x_2^2 + \frac{1}{2} \wedge 3 \mapsto 3,\ x_1 \vee 2 \mapsto x_1 \vee 2$ is a $\wedge$-strategy. The max-MSP $\boldsymbol{f}^\pi$ is given by $\boldsymbol{f}^\pi(x_1, x_2)^\top = (3, x_1 \vee 2)^\top$. □

We collect some elementary facts concerning strategies.

Lemma 2. *Let $\boldsymbol{f}$ be a feasible min-max-MSP. Then (1) $\mu\boldsymbol{f}^\sigma \leq \mu\boldsymbol{f}$ for every $\sigma \in \Sigma$; (2) $\mu\boldsymbol{f}^\pi \geq \mu\boldsymbol{f}$ for every $\pi \in \Pi^*$; (3) $\mu\boldsymbol{f}^\pi = \mu\boldsymbol{f}$ for some $\pi \in \Pi^*$.*

In [5] the authors consider a subclass of recursive stochastic games for which they prove that a positional optimal strategy exists for the player who wants to maximize the outcome (Theorem 2). The outcome of such a game is the least fixpoint of some min-max-MSP $\boldsymbol{f}$. In our setting, Theorem 2 of [5] implies that there exists a $\vee$-strategy σ such that $\mu\boldsymbol{f}^\sigma = \mu\boldsymbol{f}$ — provided that $\boldsymbol{f}$ is derived from such a recursive stochastic game. Example 3 shows that this property does not hold for arbitrary min-max-MSPs.

Example 3. Consider $\boldsymbol{f}$ from Example 1. Let $\sigma_1, \sigma_2 \in \Sigma$ be defined by $\sigma_1(x_1 \vee 2) = x_1$ and $\sigma_2(x_1 \vee 2) = 2$. Then $\mu\boldsymbol{f}^{\sigma_1} = (1, 1)^\top$, $\mu\boldsymbol{f}^{\sigma_2} = (\frac{5}{2}, 2)^\top$ and $\mu\boldsymbol{f} = (3, 3)^\top$. □

The proof of the following fundamental result is inspired by the proof of Theorem 2 in [5]. Although the result looks very natural it is non-trivial to prove.

Theorem 1. *Let $\boldsymbol{f}$ be a feasible max-MSP. Then $\mu\boldsymbol{f}^\sigma = \mu\boldsymbol{f}$ for some $\sigma \in \Sigma$.*

3 A Class of Applications: Extinction Games

In order to illustrate the interest of min-max-MSPs we consider *extinction games*, which are special stochastic games. Consider a world of n different species $s_1, \ldots, s_n$. Each species s_i is controlled by one of two adversarial players. For each s_i there is a non-empty set A_i of actions. An action $a \in A_i$ replaces a single individual of species s_i by other individuals specified by the action a. The actions can be probabilistic. E.g., an action could transform an adult rabbit to zero individuals with probability 0.2, to an adult rabbit with probability 0.3 and to an adult and a baby rabbit with probability 0.5.

Another action could transform an adult rabbit to a fat rabbit. The max-player (min-player) wants to maximize (minimize) the probability that some initial population is extinguished. During the game each player continuously chooses an individual of a species s_i controlled by her/him and applies an action from A_i to it. Note that actions on different species are never in conflict and the execution order is irrelevant. What is the probability that the population is extinguished if the players follow optimal strategies?

To answer those questions we set up a min-max-MSP $\boldsymbol{f}$ with one min-max-polynomial for each species, thereby following [10, 5]. The variables X_i represent the probability that a population with only a single individual of species s_i is extinguished. In the rabbit example we have $X_{\text{adult}} = 0.2 + 0.3X_{\text{adult}} + 0.5X_{\text{adult}}X_{\text{baby}} \vee X_{\text{fat}}$, assuming that the adult rabbits are controlled by the max-player. The probability that an initial population with p_i individuals of species s_i is extinguished is given by $\prod_{i=1}^{n}((\mu \boldsymbol{f})_i)^{p_i}$. The stochastic termination games of [5, 6, 14] can be considered as extinction games. In the following we present another instance.

The primaries game. Hillary Clinton has to decide her strategy in the primaries. Her team estimates that undecided voters have not yet decided to vote for her for three possible reasons: they consider her (a) cold and calculating, (b) too much part of Washington's establishment, or (c) they listen to Obama's campaign. So the team decides to model those problems as species in an extinction game. The larger the population of a species, the more influenced is an undecided voter by the problem. The goal of Clinton's team is to maximize the extinction probabilities.

Clinton's possible actions for problem (a) are *showing emotions* or *concentrating on her program*. If she shows emotions, her team estimates that the individual of problem (a) is removed with probability 0.3, but with probability 0.7 the action backfires and produces yet another individual of (a). This and the effect of concentrating on her program can be read off from Equation (1) below. For problem (b), Clinton can choose between concentrating on her voting record or her statement "I'll be ready from day 1". Her team estimates the effect as given in Equation (2). Problem (c) is controlled by Obama, who has the choice between his "change" message, or attacking Clinton for her position on Iraq, see Equation (3).

$$X_a = 0.3 + 0.7X_a^2 \quad \vee \quad 0.1 + 0.9X_c \tag{1}$$

$$X_b = 0.1 + 0.9X_c \quad \vee \quad 0.4X_b + 0.3X_c + 0.3 \tag{2}$$

$$X_c = 0.5X_b + 0.3X_b^2 + 0.2 \quad \wedge \quad 0.5X_a + 0.4X_aX_b + 0.1X_b \tag{3}$$

What should Clinton and Obama do? What are the extinction probabilities, assuming perfect strategies? In the next sections we show how to efficiently solve these problems.

4 The τ-Method

Assume that $\boldsymbol{f}$ denotes a feasible min-max-MSP. In this section we present our first method for computing $\mu\boldsymbol{f}$ approximatively. We call it τ-method. This method computes, for each approximate $\boldsymbol{x}^{(i)}$, the next approximate $\boldsymbol{x}^{(i+1)}$ as the least fixpoint of a piecewise linear approximation $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{x}^{(i)}) \vee \boldsymbol{x}^{(i)}$ (see below) of $\boldsymbol{f}$ at $\boldsymbol{x}^{(i)}$. This approximation is a system of *linear* min-max-polynomials where all coefficients of monomials

of degree 1 are non-negative. Here, we call such a system a *monotone linear min-max-system* (*min-max-MLS* for short). Note that a min-max-MLS $\boldsymbol{f}$ is not necessarily a min-max-MSP, since negative coefficients of monomials of degree 0 are allowed, e.g. the min-max-MLS $f(x_1) = x_1 - 1$ is not a min-max-MSP.

In [9] a min-max-MLS $\boldsymbol{f}$ is considered as a system of equations (called *system of rational equations* in [9]) which we denote by $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ in vector form. We identify a min-max-MLS $\boldsymbol{f}$ with its interpretation as a function from $\overline{\mathbb{R}}^n$ to $\overline{\mathbb{R}}^n$ ($\overline{\mathbb{R}}$ denotes the complete lattice $\mathbb{R} \cup \{-\infty, \infty\}$). Since $\boldsymbol{f}$ is monotone on $\overline{\mathbb{R}}^n$, it has a least fixpoint $\mu\boldsymbol{f} \in \overline{\mathbb{R}}^n$ which can be computed using the strategy improvement algorithm from [9].

We now define the min-max-MLS $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y})$, a piecewise linear approximation of $\boldsymbol{f}$ at $\boldsymbol{y}$. As a first step, let us consider a monotone polynomial $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$. Given some approximate $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$, a linear approximation $\mathcal{L}(f, \boldsymbol{y}) : \mathbb{R}^n \to \mathbb{R}$ of f at $\boldsymbol{y}$ is given by the first order Taylor approximation at $\boldsymbol{y}$, i.e.,

$$\mathcal{L}(f, \boldsymbol{y})(\boldsymbol{x}) := f(\boldsymbol{y}) + f'(\boldsymbol{y})(\boldsymbol{x} - \boldsymbol{y}), \qquad \boldsymbol{x} \in \mathbb{R}^n.$$

This is precisely the linear approximation which is used for Newton's method. Now consider a max-polynomial $f = f_1 \vee \cdots \vee f_k : \mathbb{R}^n \to \mathbb{R}$. We define the approximation $\mathcal{L}(f, \boldsymbol{y}) : \mathbb{R}^n \to \mathbb{R}$ of f at $\boldsymbol{y}$ by $\mathcal{L}(f, \boldsymbol{y}) := \mathcal{L}(f_1, \boldsymbol{y}) \vee \cdots \vee \mathcal{L}(f_k, \boldsymbol{y})$. We emphasize that in this case, $\mathcal{L}(f, \boldsymbol{y})$ is in general not a linear function but a linear max-polynomial. Accordingly, for a min-MSP $f = f_1 \wedge \cdots \wedge f_k : \mathbb{R}^n \to \mathbb{R}$, we define $\mathcal{L}(f, \boldsymbol{y}) := \mathcal{L}(f_1, \boldsymbol{y}) \wedge \cdots \wedge \mathcal{L}(f_k, \boldsymbol{y})$. In this case $\mathcal{L}(f, \boldsymbol{y})$ is a linear min-polynomial. Finally, for a min-max-MSP $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, we define the approximation $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y}) : \mathbb{R}^n \to \mathbb{R}^n$ of $\boldsymbol{f}$ at $\boldsymbol{y}$ by $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y}) := (\mathcal{L}(f_1, \boldsymbol{y}), \ldots, \mathcal{L}(f_n, \boldsymbol{y}))^\top$ which is a min-max-MLS.

Example 4. Consider the min-max-MSP $\boldsymbol{f}$ from Example 1. The approximation $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, (\frac{1}{2}, \frac{1}{2}))$ is given by $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, (\frac{1}{2}, \frac{1}{2}))(x_1, x_2) = \big(\frac{1}{2}x_2 + \frac{3}{8} \wedge 3,\ x_1 \vee 2\big)$. □

Using the approximation $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{x}^{(i)})$ we define the operator $\mathcal{N}_{\boldsymbol{f}} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ which gives us, for an approximate $\boldsymbol{x}^{(i)}$, the next approximate $\boldsymbol{x}^{(i+1)}$ by

$$\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) := \mu(\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{x}) \vee \boldsymbol{x}), \qquad \boldsymbol{x} \in \mathbb{R}^n_{\geq 0}.$$

Observe that $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{x}) \vee \boldsymbol{x}$ is still a min-max-MLS (at least after introducing auxiliary variables in order to eliminate components which contain $\vee$- and $\wedge$-operators).

Example 5. In Example 4 we have: $\mathcal{N}_{\boldsymbol{f}}(\frac{1}{2}, \frac{1}{2}) = \mu(\boldsymbol{\mathcal{L}}(\boldsymbol{f}, (\frac{1}{2}, \frac{1}{2})) \vee (\frac{1}{2}, \frac{1}{2})^\top) = (\frac{11}{8}, 2)^\top$.

We collect basic properties of $\mathcal{N}_{\boldsymbol{f}}$ in the following lemma:

Lemma 3. *Let $\boldsymbol{f}$ be a feasible min-max-MSP and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$. Then:*

1. $\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}) \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$;
2. $\boldsymbol{x} = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$ *whenever* $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$;
3. *(Monotonicity of $\mathcal{N}_{\boldsymbol{f}}$)* $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \leq \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$ *whenever* $\boldsymbol{x} \leq \boldsymbol{y}$;
4. $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \leq \boldsymbol{f}(\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}))$ *whenever* $\boldsymbol{x} \leq \boldsymbol{f}(\boldsymbol{x})$;
5. $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \geq \mathcal{N}_{\boldsymbol{f}^\sigma}(\boldsymbol{x})$ *for every $\vee$-strategy $\sigma \in \Sigma$;*
6. $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \leq \mathcal{N}_{\boldsymbol{f}^\pi}(\boldsymbol{x})$ *for every $\wedge$-strategy $\pi \in \Pi$;*
7. $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) = \mathcal{N}_{\boldsymbol{f}^\pi}(\boldsymbol{x})$ *for some $\wedge$-strategy $\pi \in \Pi$.*

In particular Lemma 3 implies that the least fixpoint of $\mathcal{N}_{\boldsymbol{f}}$ is equal to the least fixpoint of $\boldsymbol{f}$. Moreover, iteration based on $\mathcal{N}_{\boldsymbol{f}}$ is at least as fast as Kleene iteration. We therefore use this operator for computing approximates to the least fixpoint. Formally, we define:

Definition 1. *We call the sequence $(\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)})$ of approximates defined by $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)} := \mathcal{N}_{\boldsymbol{f}}^k(\mathbf{0})$ for $k \in \mathbb{N}$ the $\boldsymbol{\tau}$-sequence for $\boldsymbol{f}$. We drop the subscript if it is clear from the context.*

Proposition 1. *Let $\boldsymbol{f}$ be a feasible min-max-MSP. The $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}^{(k)})$ for $\boldsymbol{f}$ (see definition 1) is monotonically increasing, bounded from above by $\mu\boldsymbol{f}$, and converges to $\mu\boldsymbol{f}$. Moreover, $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\tau}^{(k)}$ for all $k \in \mathbb{N}$.*

We now show that the new approximation method converges at least *linearly* to the least fixpoint. Theorem 6.2 of [3] implies the following lemma about the convergence of Newton's method for MSPs, i.e., systems without maxima and minima.

Lemma 4. *Let $\boldsymbol{f}$ be a feasible quadratic MSP. The sequence $(\boldsymbol{\tau}^{(k)})_{k\in\mathbb{N}}$ converges linearly to $\mu\boldsymbol{f}$. More precisely, there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\boldsymbol{\tau}^{(k_{\boldsymbol{f}}+i\cdot(n+1)\cdot 2^n)}$ has at least i valid bits of $\mu\boldsymbol{f}$ for every $i \in \mathbb{N}$.*

We emphasize that linear convergence is the worst case. In many practical examples, in particular if the matrix $I - \boldsymbol{f}'(\mu\boldsymbol{f})$ is invertible, Newton's method converges *exponentially*. We mean by this that the number of accurate bits of the approximation grows *exponentially* in the number of iterations.

As a first step towards our main result for this section, we use Lemma 4 to show that our approximation method converges linearly whenever $\boldsymbol{f}$ is a max-MSPs. In this case we obtain the same convergence speed as for MSPs.

Lemma 5. *Let $\boldsymbol{f}$ be a feasible max-MSP. Let $M := \{\sigma \in \Sigma \mid \mu\boldsymbol{f}^\sigma = \mu\boldsymbol{f}\}$. The set M is non-empty and $\boldsymbol{\tau}_{\boldsymbol{f}}^{(i)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^\sigma}^{(i)}$ for all $\sigma \in M$ and $i \in \mathbb{N}$.*

Proof. Theorem 1 implies that there exists a $\vee$-strategy $\sigma \in \Sigma$ such that $\mu\boldsymbol{f}^\sigma = \mu\boldsymbol{f}$. Thus M is non-empty. Let $\sigma \in M$. By induction on k Lemma 3 implies $\boldsymbol{\tau}_{\boldsymbol{f}}^{(k)} = \mathcal{N}_{\boldsymbol{f}}^k(\mathbf{0}) \geq \mathcal{N}_{\boldsymbol{f}^\sigma}^k(\mathbf{0}) = \boldsymbol{\tau}_{\boldsymbol{f}^\sigma}^{(k)}$ for every $k \in \mathbb{N}$. □

Combining Lemma 4 and Lemma 5 we get linear convergence for max-MSPs:

Theorem 2. *Let $\boldsymbol{f}$ be a feasible quadratic max-MSP. The $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}^{(k)})$ for $\boldsymbol{f}$ (see definition 1) converges linearly to $\mu\boldsymbol{f}$. More precisely, there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\boldsymbol{\tau}^{(k_{\boldsymbol{f}}+i\cdot(n+1)\cdot 2^n)}$ has at least i valid bits of $\mu\boldsymbol{f}$ for every $i \in \mathbb{N}$.*

A direct consequence of Lemma 5 is that the $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}_{\boldsymbol{f}}^{(i)})$ converges exponentially if $(\boldsymbol{\tau}_{\boldsymbol{f}^\sigma}^{(i)})$ converges exponentially for some $\sigma \in \Sigma$ with $\mu\boldsymbol{f}^\sigma = \mu\boldsymbol{f}$. This is in particular the case if the matrix $I - (\boldsymbol{f}^\sigma)'(\mu\boldsymbol{f})$ is invertible. In order to extend this result to min-max-MSPs we state the following lemma which enables us to relate the sequence $(\boldsymbol{\tau}_{\boldsymbol{f}}^{(i)})$ to the sequences $(\boldsymbol{\tau}_{\boldsymbol{f}^\pi}^{(i)})$ where $\mu\boldsymbol{f}^\pi = \mu\boldsymbol{f}$.

Lemma 6. *Let $\boldsymbol{f}$ be a feasible min-max-MSP and m denote the number of strategies $\pi \in \Pi$ with $\mu\boldsymbol{f} = \mu\boldsymbol{f}^\pi$. There is a constant $k \in \mathbb{N}$ such that for all $i \in \mathbb{N}$ there exists some strategy $\pi \in \Pi$ with $\mu\boldsymbol{f} = \mu\boldsymbol{f}^\pi$ and $\boldsymbol{\tau}_{\boldsymbol{f}^\pi}^{(i)} \leq \boldsymbol{\tau}_{\boldsymbol{f}}^{(k+m\cdot i)}$.*

We now present the main result of this section which states that our approximation method converges at least linearly also in the general case, i.e., for min-max-MSPs.

Theorem 3. *Let $\boldsymbol{f}$ be a feasible quadratic min-max-MSP and m denote the number of strategies $\pi \in \Pi$ with $\mu\boldsymbol{f} = \mu\boldsymbol{f}^{\pi}$. The $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}^{(k)})$ for $\boldsymbol{f}$ (see definition 1) converges linearly to $\mu\boldsymbol{f}$. More precisely, there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\boldsymbol{\tau}^{(k_{\boldsymbol{f}}+i\cdot m\cdot(n+1)\cdot 2^n)}$ has at least i valid bits of $\mu\boldsymbol{f}$ for every $i \in \mathbb{N}$.*

The upper bound on the convergence rate provided by Theorem 2 is by the factor m worse than the upper bound obtained for MSPs. Since m is the number of strategies $\pi \in \Pi$ with $\mu\boldsymbol{f}^{\pi} = \mu\boldsymbol{f}$, m is trivially bounded by $|\Pi|$ but is usually much smaller. The $\boldsymbol{\tau}$-sequence $(\boldsymbol{\tau}_{\boldsymbol{f}}^{(i)})$ converges exponentially whenever $(\boldsymbol{\tau}_{\boldsymbol{f}^{\pi}}^{(i)})$ converges exponentially for every π with $\mu\boldsymbol{f}^{\pi} = \mu\boldsymbol{f}$ (see [2]). The latter condition is typically satisfied (see the discussion after Theorem 2).

In order to determine the approximate $\boldsymbol{\tau}^{(i+1)} = \mathcal{N}_{\boldsymbol{f}}(\boldsymbol{\tau}^{(i)})$ from $\boldsymbol{\tau}^{(i)}$ we must compute the least fixpoint of the min-max-MLS $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{\tau}^{(i)})\vee\boldsymbol{\tau}^{(i)}$. This can be done by using the strategy improvement algorithm from [9]. The algorithm iterates over $\vee$-strategies. For each strategy it solves a linear program or alternatively iterates over $\wedge$-strategies. The number of $\vee$-strategies used by this algorithm is trivially bounded by the number of $\vee$-strategies for $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{\tau}^{(i)})\vee\boldsymbol{\tau}^{(i)}$ which is exponentially in the number of $\vee$-expressions occurring in $\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{\tau}^{(i)}) \vee \boldsymbol{\tau}^{(i)}$. However, we do not know an example for which the algorithm considers more than linearly many strategies.

5 The $\boldsymbol{\nu}$-Method

The $\boldsymbol{\tau}$-method, presented in the previous section, uses strategy iteration over $\vee$-strategies to compute $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$. This could be expensive, as there may be exponentially many $\vee$-strategies. Therefore, we derive an alternative generalization of Newton's method that in each step picks the currently most promising $\vee$-strategy directly, without strategy iteration.

Consider again a fixed feasible min-max-MSP $\boldsymbol{f}$ whose least fixpoint we want to approximate. Assume that $\boldsymbol{y}$ is some approximation of $\mu\boldsymbol{f}$. Instead of applying $\mathcal{N}_{\boldsymbol{f}}$ to $\boldsymbol{y}$, as the $\boldsymbol{\tau}$-method, we now choose a strategy $\sigma \in \Sigma$ such that $\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{f}^{\sigma}(\boldsymbol{y})$, and compute $\mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{y})$, where $\mathcal{N}_{\boldsymbol{f}^{\sigma}}$ was defined in Section 4 as $\mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{y}) := \mu(\boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y})\vee\boldsymbol{y})$. In the following we write $\mathcal{N}_{\sigma}$ instead of $\mathcal{N}_{\boldsymbol{f}^{\sigma}}$ if $\boldsymbol{f}$ is understood.

Assume for a moment that $\boldsymbol{f}$ is a max-MSP and that there is a unique $\sigma \in \Sigma$ such that $\boldsymbol{f}(\boldsymbol{y}) = \boldsymbol{f}^{\sigma}(\boldsymbol{y})$. The approximant $\mathcal{N}_{\sigma}(\boldsymbol{y})$ is the result of applying one iteration of Newton's method, because $\boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y})$ is not only a linearization of $\boldsymbol{f}^{\sigma}$, but the first order Taylor approximation of $\boldsymbol{f}$ at $\boldsymbol{y}$. More precisely, $\boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y})(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{y}) + \boldsymbol{f}'(\boldsymbol{y})\cdot(\boldsymbol{x}-\boldsymbol{y})$, and $\mathcal{N}_{\sigma}(\boldsymbol{y})$ is obtained by solving $\boldsymbol{x} = \boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y})(\boldsymbol{x})$. In this sense, the $\boldsymbol{\nu}$-method is a more direct generalization of Newton's method than the $\boldsymbol{\tau}$-method. Formally, we define the $\boldsymbol{\nu}$-method by a sequence of approximates, the *$\boldsymbol{\nu}$-sequence*.

Definition 2 ($\boldsymbol{\nu}$-sequence). *A sequence $(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})_{k\in\mathbb{N}}$ is called* $\boldsymbol{\nu}$-sequence *of a min-max-MSP $\boldsymbol{f}$ if $\boldsymbol{\nu}_{\boldsymbol{f}}^{(0)} = \mathbf{0}$ and for each k there is a strategy $\sigma_{\boldsymbol{f}}^{(k)} \in \Sigma$ with $\boldsymbol{f}(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)}) = \boldsymbol{f}^{\sigma_{\boldsymbol{f}}^{(k)}}(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})$ and $\boldsymbol{\nu}_{\boldsymbol{f}}^{(k+1)} = \mathcal{N}_{\sigma_{\boldsymbol{f}}^{(k)}}(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})$. We may drop the subscript if $\boldsymbol{f}$ is understood.*

Notice the nondeterminism here if there is more than one $\vee$-strategy that attains $\boldsymbol{f}(\boldsymbol{\nu}^{(k)})$. The following proposition is analogous to Proposition 1 and states some basic properties of $\boldsymbol{\nu}$-sequences.

Proposition 2. *Let $\boldsymbol{f}$ be a feasible min-max-MSP. The sequence $(\boldsymbol{\nu}^{(k)})$ is monotonically increasing, bounded from above by $\mu\boldsymbol{f}$, and converges to $\mu\boldsymbol{f}$. More precisely, we have $\boldsymbol{\kappa}^{(k)} \leq \boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \leq \boldsymbol{\nu}^{(k+1)} \leq \mu\boldsymbol{f}$ for all $k \in \mathbb{N}$.*

The goal of this section is again to strengthen Proposition 2 towards quantitative convergence results for $\boldsymbol{\nu}$-sequences. To achieve this goal we again relate the convergence of $\boldsymbol{\nu}$-sequences to the convergence of Newton's method for MSPs. If $\boldsymbol{f}$ is an MSP, Lemma 4 allows to argue about the Newton operator $\mathcal{N}_{\boldsymbol{f}}$ when applied to approximates $\boldsymbol{x} \leq \mu\boldsymbol{f}$. To transfer this result to min-max-MSPs $\boldsymbol{f}$ we need an invariant like $\boldsymbol{\nu}^{(k)} \leq \mu\boldsymbol{f}^{\sigma^{(k)}}$ for $\boldsymbol{\nu}$-sequences. As a first step to such an invariant we further restrict the selection of the $\sigma^{(k)}$. Roughly speaking, the strategy in a component i is only changed when it is immediate that component i has not yet reached its fixpoint.

Definition 3 (lazy strategy update). *Let $\boldsymbol{x} \leq \boldsymbol{f}^{\sigma}(\boldsymbol{x})$ for a $\sigma \in \Sigma$. We say that $\sigma' \in \Sigma$ is obtained from $\boldsymbol{x}$ and σ by a* lazy strategy update *if $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}^{\sigma'}(\boldsymbol{x})$ and $\sigma'(f_i) = \sigma(f_i)$ holds for all components i with $f_i(\boldsymbol{x}) = x_i$. We call a $\boldsymbol{\nu}$-sequence $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$* lazy *if for all k, the strategy $\sigma^{(k)}$ is obtained from $\boldsymbol{\nu}^{(k)}$ and $\sigma^{(k-1)}$ by a lazy strategy update.*

The key property of lazy $\boldsymbol{\nu}$-sequences is the following non-trivial invariant.

Lemma 7. *Let $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ be a lazy $\boldsymbol{\nu}$-sequence. Then $\boldsymbol{\nu}^{(k)} \leq \mu\boldsymbol{f}^{\sigma^{(k)}\pi}$ holds for all $k \in \mathbb{N}$ and all $\pi \in \Pi^*$.*

The following example shows that lazy strategy updates are essential to Lemma 7 even for max-MSPs.

Example 6. Consider the MSP $\boldsymbol{f}(x,y) = (\frac{1}{2} \vee x, xy + \frac{1}{2})$. Let $\sigma^{(0)}(\frac{1}{2} \vee x) = \frac{1}{2}$ and $\sigma^{(1)}(\frac{1}{2} \vee x) = x$. Then there is a $\boldsymbol{\nu}$-sequence $(\boldsymbol{\nu}^{(k)})$ with $\boldsymbol{\nu}^{(0)} = \mathbf{0}$, $\boldsymbol{\nu}^{(1)} = \mathcal{N}_{\sigma^{(0)}}(\mathbf{0}) = (\frac{1}{2}, 0)$, $\boldsymbol{\nu}^{(2)} = \mathcal{N}_{\sigma^{(1)}}(\boldsymbol{\nu}^{(1)})$. However, the conclusion of Lemma 7 does not hold, because $(\frac{1}{2}, 0) = \boldsymbol{\nu}^{(1)} \not\leq \mu\boldsymbol{f}^{\sigma^{(1)}} = (0, \frac{1}{2})$. Notice that $\sigma^{(1)}$ is not obtained by a lazy strategy update, as $f_1(\boldsymbol{\nu}^{(1)}) = \nu_1^{(1)}$. □

Lemma 7 falls short of our subgoal to establish $\boldsymbol{\nu}^{(k)} \leq \mu\boldsymbol{f}^{\sigma^{(k)}}$, because $\Pi \setminus \Pi^*$ might be non-empty. In fact, we provide an example in [2] showing that $\boldsymbol{\nu}^{(k)} \leq \mu\boldsymbol{f}^{\sigma^{(k)}\pi}$ does not always hold for all $\pi \in \Pi$, even when $\boldsymbol{f}^{\sigma^{(k)}\pi}$ is feasible. Luckily, Lemma 7 will suffice for our convergence speed result.

The left procedure of Algorithm 1 summarizes the *lazy $\boldsymbol{\nu}$-method* which works by computing lazy $\boldsymbol{\nu}$-sequences. The following lemma relates the $\boldsymbol{\nu}$-method for min-max-MSPs to Newton's method for MSPs.

Lemma 8. *Let $\boldsymbol{f}$ be a feasible min-max-MSP and $(\boldsymbol{\nu}^{(k)})$ a lazy $\boldsymbol{\nu}$-sequence. Let m be the number of strategy pairs $(\sigma,\pi) \in \Sigma \times \Pi$ with $\mu\boldsymbol{f} = \mu\boldsymbol{f}^{\sigma\pi}$. Then $m \geq 1$ and there is a constant $k_{as} \in \mathbb{N}$ such that, for all $k \in \mathbb{N}$, there exist strategies $\sigma \in \Sigma$, $\pi \in \Pi$ with $\mu\boldsymbol{f} = \mu\boldsymbol{f}^{\sigma\pi}$ and $\boldsymbol{\nu}_{\boldsymbol{f}}^{(k_{as}+m\cdot k)} \geq \boldsymbol{\tau}_{\boldsymbol{f}^{\sigma\pi}}^{(k)}$.*

Algorithm 1 lazy $\boldsymbol{\nu}$-method

procedure lazy-$\boldsymbol{\nu}(\boldsymbol{f}, k)$
assumes: $\boldsymbol{f}$ is a min-max-MSP
returns: $\boldsymbol{\nu}^{(k)}, \sigma^{(k)}$ obtained by k iterations of the lazy $\boldsymbol{\nu}$-method

$\boldsymbol{\nu} \leftarrow \mathbf{0}$
$\sigma \leftarrow$ any $\sigma \in \Sigma$ such that $\boldsymbol{f}(\mathbf{0}) = \boldsymbol{f}^{\sigma}(\mathbf{0})$
for i **from** 1 **to** k **do**
  $\boldsymbol{\nu} \leftarrow \mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{\nu})$
  $\sigma \leftarrow$ lazy strategy update from $\boldsymbol{\nu}$ and σ
od
return $\boldsymbol{\nu}, \sigma$

procedure $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{y})$
assumes: $\boldsymbol{f}$ is a min-MSP, $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$
returns: $\mu(\boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y}) \vee \boldsymbol{y})$

$\boldsymbol{g} \leftarrow$ linear min-MSP with $\boldsymbol{g}(\boldsymbol{d}) = \boldsymbol{\mathcal{L}}(\boldsymbol{f}, \boldsymbol{y})(\boldsymbol{y} + \boldsymbol{d}) - \boldsymbol{y}$
$\boldsymbol{u} \leftarrow \boldsymbol{\kappa}^{(n)}_{\boldsymbol{g}}$
$\widetilde{\boldsymbol{g}} \leftarrow (\widetilde{g}_1, \ldots, \widetilde{g}_n)^{\top}$ where $\widetilde{g}_i = \begin{cases} 0 & \text{if } u_i = 0 \\ g_i & \text{if } u_i > 0 \end{cases}$
$\boldsymbol{d}^* \leftarrow$ maximize $x_1 + \cdots + x_n$ subject to $\mathbf{0} \leq \boldsymbol{x} \leq \widetilde{\boldsymbol{g}}(\boldsymbol{x})$ by 1 LP
return $\boldsymbol{y} + \boldsymbol{d}^*$

In typical cases, i.e., if $I - (\boldsymbol{f}^{\sigma\pi})'(\mu\boldsymbol{f})$ is invertible for all $\sigma \in \Sigma$ and $\pi \in \Pi$ with $\mu\boldsymbol{f}^{\sigma\pi} = \mu\boldsymbol{f}$, Newton's method converges exponentially. The following theorem captures the worst-case, in which the lazy $\boldsymbol{\nu}$-method still converges linearly.

Theorem 4. *Let $\boldsymbol{f}$ be a quadratic feasible min-max-MSP. The lazy $\boldsymbol{\nu}$-sequence $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ converges linearly to $\mu\boldsymbol{f}$. More precisely, let m be the number of strategy pairs $(\sigma, \pi) \in \Sigma \times \Pi$ with $\mu\boldsymbol{f} = \mu\boldsymbol{f}^{\sigma\pi}$. Then there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\boldsymbol{\nu}^{(k_{\boldsymbol{f}} + i \cdot m \cdot (n+1) \cdot 2^n)}$ has at least i valid bits of $\mu\boldsymbol{f}$ for every $i \in \mathbb{N}$.*

Next we show that $\mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{y})$ can be computed exactly by solving a single LP. The right procedure of Algorithm 1 accomplishes this by taking advantage of the following proposition which states that $\mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{y})$ can be determined by computing the least fixpoint of some linear min-MSP $\boldsymbol{g}$.

Proposition 3. *Let $\boldsymbol{y} \leq \boldsymbol{f}^{\sigma}(\boldsymbol{y}) \leq \mu\boldsymbol{f}$. Then $\mathcal{N}_{\boldsymbol{f}^{\sigma}}(\boldsymbol{y}) = \boldsymbol{y} + \mu\boldsymbol{g}$ for the linear min-MSP $\boldsymbol{g}$ with $\boldsymbol{g}(\boldsymbol{d}) = \boldsymbol{\mathcal{L}}(\boldsymbol{f}^{\sigma}, \boldsymbol{y})(\boldsymbol{y} + \boldsymbol{d}) - \boldsymbol{y}$.*

After having computed the linear min-MSP $\boldsymbol{g}$, Algorithm 1 determines the 0-components of $\mu\boldsymbol{g}$. This can be done by performing n Kleene steps, since $(\mu\boldsymbol{g})_i = 0$ whenever $(\boldsymbol{\kappa}^{(n)}_{\boldsymbol{g}})_i = 0$. Let $\widetilde{\boldsymbol{g}}$ be the linear min-MSP obtained from $\boldsymbol{g}$ by substituting the constant 0 for all components g_i with $(\mu\boldsymbol{g})_i = 0$. The least fixpoint of $\widetilde{\boldsymbol{g}}$ can be computed by solving a single LP, as implied by the following lemma. The correctness of Algorithm 1 follows.

Lemma 9. *Let $\boldsymbol{g}$ be a linear min-MSP such that $g_i = 0$ whenever $(\mu\boldsymbol{g})_i = 0$ for all components i. Then $\mu\boldsymbol{g}$ is the greatest vector $\boldsymbol{x}$ with $\boldsymbol{x} \leq \boldsymbol{g}(\boldsymbol{x})$.*

The following theorem is a direct consequence of Lemma 7 for the case where $\Pi = \Pi^*$. It shows the second major advantage of the lazy $\boldsymbol{\nu}$-method, namely, that that the strategies $\sigma^{(k)}$ are meaningful in terms of games.

Theorem 5. *Let $\Pi = \Pi^*$. Let $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ be a lazy $\boldsymbol{\nu}$-sequence. Then $\boldsymbol{\nu}^{(k)} \leq \mu\boldsymbol{f}^{\sigma^{(k)}}$ holds for all $k \in \mathbb{N}$.*

As $(\boldsymbol{\nu}^{(k)})$ converges to $\mu\boldsymbol{f}$, the max-strategy $\sigma^{(k)}$ can be considered ϵ-*optimal*. In terms of games, Theorem 5 states that the strategy $\sigma^{(k)}$ guarantees the max-player an outcome

of at least $\boldsymbol{\nu}^{(k)}$. It is open whether an analogous theorem holds for the $\boldsymbol{\tau}$-method.

Application to the primaries example. We solved the equation system of Section 3 approximatively by performing 5 iterations of the lazy $\boldsymbol{\nu}$-method. Using Theorem 5 we found that Clinton can extinguish a problem (a) individual with a probability of at least $X_a = 0.492$ by concentrating on her program and her "ready from day 1" message. (More than 70 Kleene iterations would be needed to infer that X_a is at least 0.49.) As $\boldsymbol{\nu}^{(5)}$ seems to solve above equation system quite well in the sense that $\left\|\boldsymbol{f}(\boldsymbol{\nu}^{(5)}) - \boldsymbol{\nu}^{(5)}\right\|$ is small, we are pretty sure about Obama's optimal strategy: he should talk about Iraq. As $\nu^{(2)}_{X_1} > 0.38$ and $\sigma^{(2)}$ maps f_1 to $0.3 + 0.7X_1^2$, Clinton's team can use Theorem 5 to infer that $X_a \geq 0.38$ by showing emotions and using her "ready from day 1" message.

6 Discussion

In order to compare our two methods in terms of convergence speed, assume that $\boldsymbol{f}$ denotes a feasible min-max-MSP. Since $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x}) \geq \mathcal{N}_{\boldsymbol{f}^\sigma}$ (Lemma 3.5), it follows that $\boldsymbol{\tau}^{(i)}_{\boldsymbol{f}} \geq \boldsymbol{\nu}^{(i)}_{\boldsymbol{f}}$ holds for all $i \in \mathbb{N}$. This means that the $\boldsymbol{\tau}$-method is as least as fast as the $\boldsymbol{\nu}$-method if one counts the number of approximation steps. Next, we construct an example which shows that the number of approximation steps needed by the lazy $\boldsymbol{\nu}$-method can be much larger than the respective number needed by the $\boldsymbol{\tau}$-method. It is parameterized with an arbitrary $k \in \mathbb{N}$ and given by $\boldsymbol{f}(x_1, x_2) = \left(x_2 \wedge 2,\ \ x_1^2 + 0.25 \vee x_1 + 2^{-2(k+1)}\right)^\top$. Since the constant $2^{-2(k+1)}$ is represented using $\mathcal{O}(k)$ bits, it is of size linear in k. It can be shown (see [2]) that the lazy $\boldsymbol{\nu}$-method needs at least k steps. More precisely, $\nu\boldsymbol{f} - \boldsymbol{\tau}^{(k)} \geq (1.5, 1.95)$. The $\boldsymbol{\tau}$-method needs exactly 2 steps.

We now compare our approaches with the tool PReMo [14]. PReMo employs 4 different techniques to approximate $\mu\boldsymbol{f}$ for min-max-MSPs $\boldsymbol{f}$: It uses Newton's method only for MSPs without min or max. In this case both of our methods coincide with Newton's method. For min-max-MSPs, PReMo uses Kleene iteration, round-robin iteration (called Gauss-Seidel in [14]), and an "optimistic" variant of Kleene which is not guaranteed to converge. In the following we compare our algorithms only with Kleene iteration, as our algorithms are guaranteed to converge and a round-robin step is not faster than n Kleene steps.

Our methods improve on Kleene iteration in the sense that $\boldsymbol{\kappa}^{(i)} \leq \boldsymbol{\tau}^{(i)}, \boldsymbol{\nu}^{(i)}$ holds for all $i \in \mathbb{N}$, and our methods converge linearly, whereas Kleene iteration does not converge linearly in general. For example, consider the MSP $g(x) = \frac{1}{2}x^2 + \frac{1}{2}$ with $\mu g = 1$. Kleene iteration needs exponentially many iterations for j bits [4], whereas Newton's method gives exactly 1 bit per iteration. For the slightly modified MSP $\tilde{g}(x) = g(x) \wedge 1$ which has the same fixpoint, PReMo no longer uses Newton's method, as $\tilde{g}$ contains a minimum. Our algorithms still produce exactly 1 bit per iteration.

In the case of linear min-max systems our methods compute the precise solution and not only an approximation. This applies, for example, to the max-linear system of [14] describing the expected time of termination of a nondeterministic variant of Quicksort. Notice that Kleene iteration does not compute the precise solution (except for trivial instances), even for linear MSPs without min or max.

We implemented our algorithms prototypically in Maple and ran them on the quadratic nonlinear min-max-MSP describing the termination probabilities of a recursive simple stochastic game. This game stems from the example suite of PReMo (`rssg2.c`) and we used PReMo to produce the equations. Both of our algorithms reached the least fixpoint after 2 iterations. So we could compute the precise $\mu \boldsymbol{f}$ and optimal strategies for both players, whereas PReMo computes only approximations of $\mu \boldsymbol{f}$.

7 Conclusion

We have presented the first methods for approximatively computing the least fixpoint of min-max-MSPs, which are guaranteed to converge at least linearly. Both of them are generalizations of Newton's method. Whereas the $\boldsymbol{\tau}$-method converges faster in terms of number of approximation steps, one approximation step of the $\boldsymbol{\nu}$-method is cheaper. Furthermore, we have shown that the $\boldsymbol{\nu}$-method computes ϵ-optimal strategies for games. Whether such a result can also be established for the $\boldsymbol{\tau}$-method is still open. A direction for future research is to evaluate our methods in practice. In particular, the influence of imprecise computation through floating point arithmetic should be studied. It would also be desirable to find a bound on the "threshold" $k_{\boldsymbol{f}}$.

Acknowledgement. We thank the anonymous referees for valuable comments.

References

1. A. Condon. The complexity of stochastic games. *Inf. and Comp.*, 96(2):203–224, 1992.
2. J. Esparza, T. Gawlitza, S. Kiefer, and H. Seidl. Approximative methods for monotone systems of min-max-polynomial equations. Technical report, Technische Universität München, Institut für Informatik, February 2008.
3. J. Esparza, S. Kiefer, and M. Luttenberger. Convergence thresholds of Newton's method for monotone polynomial equations. In *Proceedings of STACS*, pages 289–300, 2008.
4. K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. In *STACS*, pages 340–352, 2005.
5. K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *Proc. ICALP 2005*, volume 3580 of *LNCS*. Springer, 2005.
6. K. Etessami and M. Yannakakis. Efficient qualitative analysis of classes of recursive Markov decision processes and simple stochastic games. In *STACS*, pages 634–645, 2006.
7. J. Filar and K. Vrieze. *Competitive Markov Decision processes*. Springer, 1997.
8. T. Gawlitza and H. Seidl. Precise fixpoint computation through strategy iteration. In *European Symposium on Programming (ESOP)*, LNCS 4421, pages 300–315. Springer, 2007.
9. T. Gawlitza and H. Seidl. Precise relational invariants through strategy iteration. In J. Duparc and T. A. Henzinger, editors, *CSL*, volume 4646 of *LNCS*, pages 23–40. Springer, 2007.
10. T. Harris. *The Theory of Branching Processes*. Springer, 1963.
11. S. Kiefer, M. Luttenberger, and J. Esparza. On the convergence of Newton's method for monotone systems of polynomial equations. In *STOC*, pages 217–226. ACM, 2007.
12. A. Neyman and S. Sorin. *Stochastic Games and Applications*. Kluwer Academic Press, 2003.
13. J. Ortega and W. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, 1970.
14. D. Wojtczak and K. Etessami. PReMo: An analyzer for probabilistic recursive models. In *Proc. of TACAS*, volume 4424 of *LNCS*, pages 66–71. Springer, 2007.