Solving Discounted Stochastic Two-Player Games with Near-Optimal Time and Sample Complexity

Aaron Sidford  
Stanford University  
sidford@stanford.edu

Mengdi Wang  
Princeton University  
mengdiw@princeton.edu

Lin F. Yang  
UCLA  
linyang@ee.ucla.edu

Yinyu Ye  
Stanford University  
yyyeye@stanford.edu

Abstract

In this paper we settle the sampling complexity of solving discounted two-player turn-based zero-sum stochastic games up to polylogarithmic factors. Given a stochastic game with discount factor $\gamma \in (0, 1)$ we provide an algorithm that computes an $\epsilon$-optimal strategy with high-probability given $\tilde{O}((1 - \gamma)^{-3}\epsilon^{-2})$ samples from the transition function for each state-action-pair. Our algorithm runs in time nearly linear in the number of samples and uses space nearly linear in the number of state-action pairs. As stochastic games generalize Markov decision processes (MDPs) our runtime and sample complexities are optimal due to Azar et al. (2013). We achieve our results by showing how to generalize a near-optimal Q-learning based algorithms for MDP, in particular Sidford et al. (2018a), to two-player strategy computation algorithms. This overcomes limitations of standard Q-learning and strategy iteration or alternating minimization based approaches and we hope will pave the way for future reinforcement learning results by facilitating the extension of MDP results to multi-agent settings with little loss.

1 Introduction

In this paper we study the sample complexity of learning a near-optimal strategy in discounted two-player turn-based zero-sum stochastic games Shapley (1953); Hansen et al. (2013), which we refer to more concisely as stochastic games. Stochastic games model dynamic strategic settings in which two players take turns and the state of game evolves stochastically according to some transition law. This model encapsulates a major challenge in multi-agent learning: other agents may be learning and adapting as well. Further, stochastic games are a generalization of the Markov decision process (MDP), a fundamental model for reinforcement learning, to the two-player setting Littman (1994). MDPs can be viewed as degenerate stochastic games in which one of the players has no influence. Consequently, understanding stochastic games is a natural step towards resolving challenges in reinforcement learning of extending single-agent learning to multi-agent settings.

There is a long line of research in both MDPs and stochastic games (for a more thorough introduction, see Filar and Vrieze (2012); Hansen et al. (2013) and references therein). Strikingly, Hansen et al. (2013) showed that there exists a pure-strategy Nash equilibrium which can be computed in strongly polynomial time for stochastic games, if the game matrix is fully accessible and the discount factor is fixed. In reinforcement learning settings, however, the transition function of the game is unknown and a common goal is to obtain an approximately optimal strategy (a function that maps states to actions) that is able to obtain an expected cumulative reward of at least (or at most) the Nash equilibrium value no matter what the other player does. Unfortunately, despite interest in generalizing MDP results to stochastic games, currently the best known running times/sample complexity for solving stochastic games in a variety of settings are worse than for solving MDPs. This may not be surprising since in general stochastic games are harder to solve than MDPs, e.g., whereas MDPS can be solved in (weakly) polynomial time it remains open whether or not the same can be done for stochastic games.

There are two natural approaches towards achieving sample complexity bounds for solving stochastic games. The first is to note that the popular stochastic value iteration, dynamic programming, and Q-learning methods all apply to stochastic games Littman (1994); Hu and Wellman (2003); Littman (2001a); Perolat et al. (2018a).
Consequently, recent advances in these methods developed for MDPs can be directly generalized to solving stochastic games (though the sample complexity of these generalized methods has not been analyzed previously). It is tempting to generalize the analysis of sample optimal methods for estimating values of MDPs to stochastic games. However, this is challenging as these methods rely on monotonicities in MDPs induced by the linear program nature of the problem .

The second approach would be to apply strategy iteration or alternating minimization / maximization to reduce solving stochastic games to approximately solving a sequence of MDPs. Unfortunately, the best analysis of such a method requires solving $\Omega(1/(1 - \gamma))$ MDPs. Consequently, even if this approach could be carried out with approximate MDP solvers, the resulting sample complexity for solving stochastic games would be larger than that needed for solving MDPs. More discussion of related literatures is given in Section 1.4.

Given the importance of solving stochastic games in reinforcement learning (e.g. Hu et al. (1998); Bowling and Veloso (2000, 2001); Hu and Wellman (2003); Arslan and Yüksel (2017)), this suggests the following fundamental open problem:

**Can we design stochastic game learning algorithms that provably match the performance of MDP algorithms and achieve near-optimal sample complexities?**

In this paper, we answer this question in the affirmative in the particular case of solving discounted stochastic games with a generative model, i.e. an oracle for sampling from the transition function for state-action pairs. We provide an algorithm with the same near-optimal sample complexity that is known for solving discounted MDPs. Further, we achieve this result by showing how to transform particular MDP algorithms to solving stochastic games that satisfy particular two-sided monotonicity constraints. Therefore, while there is a major gap between MDPs and stochastic games in terms of computation time for obtaining the exact solutions, this gap disappears when considering the sampling complexity between the two. We hope this work opens the door to more generally extend results for MDP to stochastic games and thereby enable the application of the rich research on reinforcement learning to a broader multi-player settings with little overhead.

### 1.1 The Model

Formally, throughout this paper, we consider discounted turn-based two-player zero-sum stochastic games described as the tuple $\mathcal{G} = (\mathcal{S}_{\min}, \mathcal{S}_{\max}, \mathcal{A}, P, r, \gamma)$. In these games there are two players, a min or minimization player which seeks to minimize the expected cumulative reward in the game and a max or maximization player which seeks to maximize the cumulative reward. Here, $\mathcal{S}_{\min}$ and $\mathcal{S}_{\max}$ are disjoint finite sets of states controlled by the min-player and the max-player respectively and their union $\mathcal{S} := \mathcal{S}_{\min} \cup \mathcal{S}_{\max}$ is the set of all possible states of the game. Further, $\mathcal{A}$ is a finite set of actions available at each state, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is a transition probability function, $r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is the payoff or reward function and $\gamma \in (0, 1)$ is a discount factor.

Stochastic games $\mathcal{G} = (\mathcal{S}_{\min}, \mathcal{S}_{\max}, \mathcal{A}, P, r, \gamma)$ are played dynamically in a sequence of turns, $\{t\}_{t=0}^{\infty}$, starting from some initial state $s^0 \in \mathcal{S}$ at turn $t = 0$. In each turn $t \geq 0$, the game is in one of the states $s^t \in \mathcal{S}$ and the player who controls the state $s^t$ chooses or plays an action $a^t$ from the action space $\mathcal{A}$. This action yields reward $r^t := r(s^t, a^t)$ for the turn and causes the next state $s^{t+1}$ to be chosen at random from $\mathcal{S}$ where the transition probability $P(s^{t+1} = s' | s^t, a^t)$ is defined to be the probability of transitioning to state $s'$ at turn $t + 1$. The goal of the min-player (resp. max-player) is to choose actions to minimize (resp. maximize) the expected infinite-horizon discounted-reward or value of the game $\sum_{t=0}^{\infty} \gamma^t r^t$.

In this paper we focus on the case where the players play pure (deterministic) stationary strategies (policies), i.e. strategies which depend only on the current state. That is we wish to compute a min-player strategy or policy $\pi_{\min} : \mathcal{S}_{\min} \rightarrow \mathcal{A}$ which defines the action the min player chooses at a state in $\mathcal{S}_{\min}$ and max-player strategy $\pi_{\max} : \mathcal{S}_{\max} \rightarrow \mathcal{A}$ which defines the action the max player chooses at a state in $\mathcal{S}_{\max}$. We call a pair of min-player and max-player strategies $\sigma = (\pi_{\min}, \pi_{\max})$ simply a strategy. Further, we let $\sigma(s) := \pi_{\min}(s)$ for $s \in \mathcal{S}_{\min}$ and $\sigma(s) := \pi_{\max}(s)$ for $s \in \mathcal{S}_{\max}$ and define the value function or expected discounted cumulative reward by $v^\sigma$ where

$$v^\sigma(s) = v[\sigma](s) := E \left[ \sum_{t=0}^{\infty} \gamma^t r(s^t, \sigma(s^t)) \middle| s^0 = s \right]$$

for all $s \in \mathcal{S}$ and the expectation is over the random sequence of states, $s^0, s^1, s^2, \ldots$ generated according to $P$ under the strategy $\sigma$, i.e. $P(s^{t+1} = s' | s^t, s^{t-1}, \ldots, s^0) = P(s' | s^t, \sigma(s^t))$ for all $t > 0$.

Standard reductions allow this result to be applied for rewards of a broader range (Sidford et al. 2018a). Further, while we assume there are the same number of actions per state, our results easily extend to the case where this is non-uniform; in this case our dependencies on $|\mathcal{S}|, |\mathcal{A}|$ can be replaced with the number of state-action pairs.
Our goal in solving a game is to compute an approximate Nash equilibrium restricted to stationary strategies [Nash 1951; Maskin and Tirole 2001]. We call a strategy \( \sigma = (\pi_{\min}, \pi_{\max}) \) an equilibrium strategy or optimal if
\[
\max_{\pi_{\min}, \pi_{\max}} \mathbf{v}(\pi_{\min}, \pi_{\max}) \leq \mathbf{v}^{\sigma} \leq \min_{\pi_{\min}, \pi_{\max}} \mathbf{v}(\pi_{\min}, \pi_{\max})
\]
and we call it \( \epsilon \)-optimal if these same inequalities hold up to an additive \( \epsilon \) entrywise. It is worth noting that the best response strategy to a stationary policy is also stationary [Fudenberg and Tirole 1991] and there always exists a pure stationary strategy attaining the Nash equilibrium [Shapley 1953]. Consequently, it is sufficient to focus on deterministic strategies.

Throughout this paper we focus on solving stochastic games in the learning setting where the game is not fully specified. We assume that a generative model is available which given any state-action pair, i.e. \( s \in S \) and \( a \in A \), can sample a random \( s' \) independently at random from the transition probability function, i.e. \( \Pr[s' | s, a] = \mathbf{P}(s' | s, a) \). Accessibility to a generative model is a standard and natural assumption [Kakade 2003; Azar et al. 2013; Sidford et al. 2018a; Agarwal et al. 2019] and corresponds to PAC learning. The special case of solving a MDP given a generative model has been studied extensively [Kakade 2003; Azar et al. 2013; Sidford et al. 2018a; Agarwal et al. 2019] and is a natural proving ground towards designing theoretically motivated reinforcement learning algorithms.

1.2 Our Results

In this paper we provide an algorithm that computes an \( \epsilon \)-optimal strategy using a sample size that matches the best known sample complexity for solving discounted MDPs. Further, our algorithm runs in time proportional to the number of samples and space proportional to \( |S||A| \). Interestingly, we achieve this result by showing how to run two-player variant of Q-learning such that the value-strategy sequences induced enjoy certain monotonicity properties. Essentially, we show that providing a value improving algorithm is sufficiently stable, then it can be extended to the two-player setting with limited loss. This allows us to leverage recent advances in solving single player games to solve stochastic games with limited overhead. Our main result is given below.

**Theorem 1.1 (Main Theorem).** There is an algorithm which given a stochastic game, \( \mathcal{G} = (S_{\min}, S_{\max}, P, r, \gamma) \) with a generative model, outputs, with probability at least \( 1 - \delta \), an \( \epsilon \)-optimal strategy \( \sigma \) by querying \( Z = \tilde{O}(|S||A|(1 - \gamma)^{-3}\epsilon^{-2}) \) samples, where \( \epsilon \in (0, 1) \) and \( \tilde{O}(\cdot) \) hides polylogarithmic factors. The algorithm runs in time \( \tilde{O}(Z) \) and uses space \( \tilde{O}(|S||A|) \).

Our sample and time complexities are optimal due to a known lower bound in the single player case by [Azar et al. 2013]. It was shown in [Azar et al. 2013] that solving any one-player MDP to \( \epsilon \)-optimality with high probability needs at least \( \Omega(|S||A|(1 - \gamma)^{-3}\epsilon^{-2}) \) samples. Our sample complexity upper bound generalizes the recent sharp sample complexity results for solving the discounted MDP [Sidford et al. 2018a; Agarwal et al. 2019], and tightly matches the information-theoretic sample complexity up to polylogarithmic factors. This result provides the first and near-optimal sample complexity for solving the two-person stochastic game.

1.3 Notations and Preliminaries

**Notation:** We use \( \mathbf{1} \) to denote the all-ones vector whose dimension is adapted to the context. We use the operators \( | \cdot |, (\cdot)^{\top}, \sqrt{\cdot}, \leq, \geq, \geq_{\epsilon}, \geq_{\epsilon_{\omega}} \) as entrywise operators on vectors. We identify the transition probability function \( P \) as a matrix in \( \mathbb{R}^{(S \times A) \times S} \) and each row \( P(\cdot | s, a) \in \mathbb{R}^{S} \) as a vector. We denote \( \mathbf{v} \) as a vector in \( \mathbb{R}^{S} \) and \( Q \) as a vector in \( \mathbb{R}^{S \times A} \). Therefore \( P\mathbf{v} \) is a vector in \( \mathbb{R}^{S \times A} \). We use \( \sigma \) to denote strategy pairs and \( \pi \) for the min-player or max-player strategy. For any strategy \( \sigma \), we define \( Q_{\sigma} \in \mathbb{R}^{S} \) as \( Q_{\sigma}(s) := Q(s, \sigma(s)) \) for all \( s \in S \). We denote \( P^{\sigma} \) as a linear operator defined as
\[
\forall s \in S: \quad [P^{\sigma}\mathbf{v}](s) = P(\cdot | s, \sigma(s))^{\top}\mathbf{v},
\]
\[
\forall s, a \in S \times A: \quad [P^{\sigma}Q](s, a) = P(\cdot | s, a)^{\top}Q_{\sigma}.
\]

**Min-value and max-value:** For a min-player strategy \( \pi_{\min} \), we define its value as
\[
\mathbf{v}^{\pi_{\min}} := \max_{\pi_{\max} \to A} \mathbf{v}(\pi_{\min}, \pi_{\max}),
\]
(1)

We let \( \sigma_{\max}(\pi_{\min}) \) denote a maximizing argument of the above and call it an optimal counter strategy of \( \pi_{\min} \). Thus a value of a min-player strategy gives his expected reward in the worst case. We say a min-player strategy \( \pi_{\min} \) is \( \epsilon \)-optimal if
\[
\mathbf{v}^{\pi_{\min}} \leq \min_{\pi_{\min}, \pi_{\max} \to A} \mathbf{v}^{\pi_{\min}} + \epsilon \cdot \mathbf{1}, \quad \text{entrywisely.}
\]

The value and \( \epsilon \)-optimality for the max player is defined similarly. We denote by \( \sigma^{\pi} \) the optimal strategy and by \( \mathbf{v}^{*} \) the value function of the optimal strategy.

**Q-function:** For a strategy \( \sigma \), we denote its \( Q \)-function (or action value) as \( Q^{\sigma} \in \mathbb{R}^{S \times A} \) by \( Q^{\sigma} := r + \gamma P^{\sigma}Q^{\sigma} \). For a vector \( \mathbf{v} \in \mathbb{R}^{S} \) we denote \( Q(\mathbf{v}) := r + \gamma P^{\sigma} \mathbf{v} \). Given a \( Q \in \mathbb{R}^{S \times A} \), we denote the greedy value of \( Q \) as
\[
V[Q](s) := \min_{a \in A} Q(s, a) \quad \text{if} \quad s \in S_{\min}
\]
and
\[
V[Q](s) := \max_{a \in A} Q(s, a) \quad \text{if} \quad s \in S_{\max}.
\]
Bellman Operator: We denote the Bellman operator, $T$, as follows: $T[v](s) := V[r + \gamma P_v]$. We also denote the greedy strategy, $\sigma(v)$ or $\sigma(Q)$, as the maximization/minimization argument of the $T$ operator. Moreover, for a given strategy $\sigma$, we denote $T_\sigma[v] = Q[v]_\sigma$. For a given min-player strategy $\pi_{\min}$, we define the half Bellman operator $\mathcal{H}_{\pi_{\min}}$:

$$\mathcal{H}_{\pi_{\min}}[v](s) = r(s, \pi_{\min}(s)) + \gamma P(\cdot | s, \pi_{\min}(s))^\top v$$

if $s \in S_{\min}$;

$$\mathcal{H}_{\pi_{\min}}[v](s) \quad \text{if} \quad s \in S_{\max}.$$ We define $\mathcal{H}_{\pi_{\max}}$ similarly. Note that $v^*$ is the unique fixed point of the Bellman operator, i.e., $T[v^*] = v^*$ (known as the Bellman equation [Bellman, 1957]). Similarly, $v^{\pi_{\min}}$ (resp. $v^{\pi_{\max}}$) is the unique fixed point for $\mathcal{H}_{\pi_{\min}}$ (resp. $\mathcal{H}_{\pi_{\max}}$). The (half) Bellman-operators satisfy the following properties (see. e.g. [Hansen et al., 2013; Puterman, 2014])

1. contraction: $\|T[v_1] - T[v_2]\| \leq \gamma \|v_1 - v_2\|$;
2. monotonicity: $v_1 \leq v_2 \Rightarrow T[v_1] \leq T[v_2]$.

High Probability: we say an algorithm has a property “with high probability” if for any $\delta$ by increasing the time and sample complexity by $O(\log(1/\delta))$ it has the property with probability $1 - \delta$.

1.4 Previous Work

Here we provide a more detailed survey of previous works related to stochastic games and MDPs. Two-person stochastic games generalize MDPs [Shapley, 1953]. When one of the players has only one action to choose from, the problem reduces to a MDP. A related game is the stochastic game where both players choose their respective actions simultaneously at each state and the process transitions to the next state under the control of both players [Shapley, 1953]. The turn-based stochastic game can be reduced to the game with simultaneous moves [Pérolat et al., 2015].

Computing an optimal strategy for a two-player turn-based zero-sum stochastic game is known to be in NP $\cap$ co-NP [Condon, 1992]. Later [Hansen et al., 2013] showed that the strategy iteration, a generalization of Howard’s policy iteration algorithm [Howard, 1960], solves the discounted problem in strongly polynomial time when the discount factor is fixed. Their work uses ideas from [Ye, 2011] which proved that the policy iteration algorithm solves the discounted MDP (DMDP) in strongly polynomial time when the discount factor is fixed. In general (e.g., if the discount factor is part of the input size), it is open if stochastic games can be solved in polynomial time [Littman, 1996]. This is in contrast to MDPs which can be solved in (weakly) polynomial time as they are a special case of linear programming.

The algorithms and complexity theory for solving two-player stochastic games is closely related to that of solving MDPs. Their is vast literature on solving MDPs which dates back to Bellman who developed value iteration in 1957 [Bellman, 1957]. The policy iteration was introduced shortly after by Howard [Howard, 1960], and its complexity has been extensively studied in [Mansour and Singh, 1999; Ye, 2011; Scherrer, 2013]. Then [d’Epenoux, 1963] and [De Ghellinck, 1960] discovered that MDPs are special cases of a linear program, which leads to the insight that the simplex method, when applied to solving DMDPs, is a simple policy iteration method. [Ye, 2011] showed that policy iteration (which is a variant of the general simplex method for linear programming) and the simplex method are strongly polynomial for DMDP and terminate in $O(|S|^2|A|(1-\gamma)^{-1}1\log(|S|/(1-\gamma)^{-1})$ iterations. [Hansen et al., 2013] and [Scherrer, 2013] improved the iteration bound to $O(|S|^2|A|(1-\gamma)^{-1}1\log(|S|/(1-\gamma)^{-1})$ for Howard’s policy iteration method. The best known convergence result for policy and strategy iteration are given by [Ye, 2005] and [Hansen et al., 2013]. The best known iteration complexities for both problems are of the order $(1-\gamma)^{-1}$, which becomes unbounded as $\gamma \to 1$. It is worth mentioning that [Ye, 2005] designed a combinatorial interior-point algorithm (CIPA) that solves the DMDP in strongly polynomial time.

Sample-based algorithms for learning value and policy functions for MDP have been studied in [Kearns and Singh, 1999; Kakade, 2003; Singh and Yee, 1994; Azar et al., 2011b; 2013; Sidford et al., 2018a; 2018b; Agarwal et al., 2019] and many others. Among these papers, Azar et al. [2013] obtains the first tight sample bound for finding an $\epsilon$-optimal value function and for finding $\epsilon$-optimal policies in a restricted $\epsilon$ regime and [Sidford et al., 2018a] obtains the first tight sample bound for finding an $\epsilon$-optimal policy for any $\epsilon$. Both sample complexities are of the form $\tilde{O}(|S||A|/(1-\gamma)^{-3})$. Lower bounds have been shown in [Azar et al., 2011a; Even-Dar et al., 2006; 2017] which achieves sample complexity $\tilde{O}(|S||A|^{2\min/\max}/r^2_{\min})$, where $t_{\min}$ is the worst-case mixing time and $\epsilon_{\max}/\epsilon_{min}$ is the ergodicity ratio. Sampling-based method for two-player stochastic game has been considered in [Wei et al., 2017] in an online learning setting. However, their algorithm leads to a sub-optimal sample-complexity when generalized to the generative model setting.
As for general stochastic games, the minimax Q-learning algorithm and the friend-and-foe Q-learning algorithm were introduced in Littman (1994) and Littman (2001a), respectively. The Nash Q-learning algorithm was proposed for zero-sum games in Hu and Wellman (2003) and for general-sum games in Littman (2001b); Hu and Wellman (1999).

2 Technique Overview

Since stochastic games are a generalization of MDPs, many techniques for solving MDPs can be immediately generalized to stochastic games. However, as we have discussed, some of the techniques used to achieve optimal sample complexities for solving MDPs in a generative model do not have a clear generalization to stochastic games. Nevertheless, we show how to design an algorithm that carefully extends particular Q-learning based methods, i.e. methods that always maintain an estimator for the optimal value function (or $Q^*$), to achieve our goals.

Q-Learning: To motivate our approach we first briefly review previous Q-learning based methods and the core technique that achieves near-optimal sample complexity. To motivate Q-learning, we first recall the value iteration algorithm solving an MDP. Given a full model for the MDP value iteration updates the iterates as follows

$$v^{(i)} \leftarrow T[v^{(i-1)}] := V(Q[v^{(i-1)}])$$

where $v^{(0)}$ can be an arbitrary vector. Since the Bellman operator is contractive and $v^*$ is a fix point of $T$, this method gives an $\epsilon$-optimal value in $O((1 - \gamma)^{-1} \log(1/\epsilon))$ iterations. In the learning setting, $T$ cannot be exactly computed. The Q-learning approach estimates $T$ by its approximate version, i.e., to compute $P[\cdot | s, a]^\top v^{(i-1)}$, we obtain samples from $P[\cdot | s, a]$, and then compute the empirical average. Then we compute the approximate Q-value at the $i$-th iteration as

$$Q^{(i)} = \hat{Q}[v^{(i-1)}] := r + \hat{P}v^{(i-1)}$$

and $\hat{T}v^{(i-1)} := V(\hat{Q}[v^{(i-1)}])$,

where

$$\hat{P}[\cdot | s, a]^\top v = \frac{1}{m} \sum_{s_i \sim P[\cdot | s, a], i \in [m]} v(s_i)$$

for some $m > 0$. Then the estimation error per step is defined as

$$\epsilon^{(i)} = Q[v^{(i-1)}] - \hat{Q}[v^{(i-1)}].$$

Since the exact value iteration takes at least $\Omega((1 - \gamma)^{-1})$ iterations to converge, the Q-learning (or approximated value iteration) takes at least $\Omega((1 - \gamma)^{-1})$ iterations. The total number of samples used over all the iterations is the sample complexity of the algorithm.

Variance Control and Monotonicity Techniques:

To obtain the optimal sample complexity for one-player MDP, one approach is to carefully bound each entry of $\epsilon^{(i)}$. By Bernstein inequality (Azar et al. (2013); Sidford et al. (2018a); Agarwal et al. (2019)), we have, with high probability,

$$|\epsilon^{(i)}| \leq \sqrt{\text{var}(v^{(i-1)})}/m \leq \sqrt{\text{var}(v^*)}/m + \text{lower-order terms.}$$

where $\text{var}(v) = P v^2 - (Pv)^2$ is the variance-of-value vector and "\leq" means “approximately less than.” Let $\pi^{(i)}$ be a policy maintained in the $i$-th iteration (e.g. the greedy policy of the current $Q$-value). Due to the estimation error $\epsilon^{(i)}$, the per step error bound reads,

$$Q^* - Q^{(i)} \leq \gamma P\pi^{(i)} Q^* - \gamma P\pi^{(i-1)} Q^{(i-1)} + \epsilon^{(i)}.$$  

To derive the overall error accumulation, Sidford et al. (2018a) use the crucial monotonicity property, i.e., since $\pi^{(i-1)}(s) = \arg \max_a Q^{(i-1)}(s, a)$, we have

$$Q^{(i-1)}(s, \pi^*(s)) \leq Q^{(i)}(s, \pi^{(i-1)}(s)).$$  \hspace{1cm} (2)

We thus have

$$Q^* - Q^{(i)} \leq \gamma P\pi^* Q^* - \gamma P\pi^* Q^{(i-1)} + \epsilon^{(i)}.$$

By induction, we have

$$Q^* - Q^{(0)} \leq (1 - \gamma P\pi^*)^{-1} \sqrt{\text{var}(v^*)}/m + \text{lower-order terms.} \hspace{1cm} (3)$$

The leading-order error accumulation term $(1 - \gamma P\pi^*)^{-1} \sqrt{\text{var}(v^*)}/m$ satisfies the so-called total variance property, and can be upper bounded uniformly by $(1 - \gamma)^{-3} m^{-1}$, resulting the correct dependence on $(1 - \gamma)$. Therefore the monotonicity property allows us to use $\pi^*$ as a proxy policy, which carefully bounds the error accumulation. For the additional subtlety of how to obtain an optimal policy, please refer to Sidford et al. (2018a) for the variance reduction technique and the monotone-policy technique.

Similar observations regarding MDPs was used in Agarwal et al. (2019) as well. This powerful technique, however, does not generalize to the game case due to the lack of monotonicity. Indeed, (2) does not hold for stochastic games due to the existence of both minimization and maximization operations in the Bellman operator. This is the critical issue which this paper seeks to overcome.

Finding Monotone Value-Strategy Sequences for Stochastic Games: Analogously to the MDP case, one approach is to bound error accumulation for stochastic games is to bound each entry of the error...
vector $\epsilon^{(i)}$ carefully. In fact, our method for solving stochastic games is very much like the MDP method used in [Sidford et al. 2018a]. However, the analysis is much different in order to resolve the difficulty introduced by the lack of monotonicity.

Since a stochastic game has two players, we modify the variance reduced Q-value iteration (vQVI) method in [Sidford et al. 2018a] to obtain a min-player strategy and a max-player strategy respectively. Since the two players are symmetric, let us focus on introducing and analyzing the algorithm for the min-player. By a slight modification of the vQVI method, we can guarantee that a sequence of strategies and values, $\{v(i), Q(i), \sigma(i), \epsilon(i)\}_{i=0}^{\infty}$, that satisfy, with high probability,
1. $v(0) \geq v(1) \geq \ldots \geq v(R) \geq v^*$;
2. $T_\pi(i)[v(i)] \leq v(i), T[v(i)] \leq v(i), H_\pi(i) [v(i)] \leq v(i)$;
3. $Q(i) \leq Q[v(i-1)] + \epsilon(i)$;
4. $v(i) \leq V(Q(i))$.

where $\sigma(i) = (\pi_{\max}(i), \pi_{\min}(i))$. The first property guarantees that the value sequences are monotonically decreasing, the second property guarantees $v(i)$ is always an upper bound of the value $v_{\min}^{(i)}$, and the third and fourth inequality guarantees that $v(i)$ is well approximated by $V(Q(i))$ and the estimation error satisfy $|\epsilon(i)| \leq \sqrt{\text{var}(v(i))/m}$, where $m$ is the total number of samples used per state-action pair. Note that, as long as we can guarantee that $v(R) - v^* \leq \epsilon$, we can guarantee the min-strategy $\pi_{\min}^{(R)}$ is also good: $v^* \leq v_{\min}^{(R)} \leq v(R)$.

Controlling Error Accumulation using Auxiliary Markovian Strategy: Due to the lack of monotonicity [2], we cannot use the optimal strategy $\sigma^*$ as a proxy strategy to carefully account for the error accumulation. To resolve this issue, we construct a new proxy strategy $\sigma^\infty$. This strategy is a Markovian strategy, which is time-dependent but not history dependent, i.e., at time $t$, the strategy played is a deterministic map $\sigma^\infty : S \rightarrow A$. The proxy strategy satisfies the following:

**Underestimation.** its value, $v[\sigma^\infty]$, (expected discounted cumulative reward starting from any time) is upper bounded by $v^*$;

**Contraction.**

\[ v^{(i)}(s) - v[\sigma^\infty](s) \leq \gamma P(s, s^\infty(s))^{\top}(v^{(i-1)} - v[\sigma^\infty_{i-1}]) + \epsilon^{(i)}(s, \sigma^\infty(s)), \]

Similarly, we can bound the error $\epsilon^{(i)}(s, \sigma^\infty(s))$ by the variance-of-value of the proxy strategy

\[ \epsilon^{(i)}(s, \sigma^\infty(s)) \leq \sqrt{\text{var}(v[\sigma^\infty](s, \sigma^\infty(s))/m + \text{lower-order terms}. \]

Based on the first property, we can upper bound

\[ v^{(i)} - v^* \leq v^{(i)} - v[\sigma^\infty]. \]

Based on the second property, and induction on $i$, we can now write a new form of error accumulation,

\[ v^{(R)} - v^* \leq \sum_{i=1}^{R} \sigma^{\infty \top} P \sigma^{\infty \top} - \ldots - P \sigma^{\infty \top} \sqrt{\text{var}(v[\sigma^\infty_{i-1}]}/m + \text{lower-order terms}, \]

where $\text{var}(v[\sigma^\infty_{i-1}])\sigma^\infty(s) := \text{var}(v[\sigma^\infty](s, \sigma^\infty(s)))$ for all $s \in S$. We derive a new law of total variance bound for the first term and ultimately prove an error accumulation upper bound:

\[ v^{(R)} - v^* \leq \sqrt{(1 - \gamma)^{\beta^m} + \text{lower-order terms}} \]

giving the optimal sample bound.

### 3 Sample Complexity of Stochastic Games

In this section, we provide and analyze our sampling-based algorithm for solving stochastic games. Recall that we have a generative model for the game such that we can obtain samples from state-action pairs. Each sample is obtained in time $O(1)$. As such we care about the total number of samples used or the total amount of time consumed by the algorithm. We will provide an efficient algorithm that takes input a generative model and obtains a good strategy for the underlying stochastic game.

We now describe the algorithm. Since the min-player and max-player are symmetric, let us focus on the min-player strategy. For the max player strategy, we can either consider the game $G' = (S_{\min}, S_{\max}, P, 1 - r, \gamma)$, in which the roles of the max and min players switched, or use the corresponding algorithm for the max-player defined in Section A.4, an algorithm that is a direct generalization from the min-player algorithm.

**The Full Algorithm.** For simplicity, let us denote $\beta = 1/(1 - \gamma)$. Our full algorithm will use the QVI-MDVSS algorithm (Algorithm 1) as a subroutine. As we will show shortly, this subroutine maintains a monotonic value strategy sequence with high probability. Suppose the algorithm is specified by an accuracy parameter $\epsilon \in (0, 1]$. We initialize a value vector $v^{(0)} = \beta \mathbf{1}$, and an arbitrary strategy $\sigma^{(0)} = (\pi_{\min}^{(0)}, \pi_{\max}^{(0)})$. Let $u^{(0)} = \beta$. Then our initial value and strategy satisfy the requirement of the input specified by Algorithm 1:

\[ v^{(0)} \leq u^{(0)} + u^{(0)} \mathbf{1}, \quad u^{(0)} \geq T[v^{(0)}], \]

and $\pi^{(0)} \geq T_{\pi^{(0)}}[v^{(0)}]$.
Algorithm 1 QVI-MDVS: algorithm for computing monotone decreasing value-strategy sequences.

1: **Input:** A generative model for stochastic game, $\mathcal{M} = (S, A, r, P, \gamma)$;
2: **Input:** Precision parameter $\epsilon \in [0, (1 - \gamma)^{-1}]$, and error probability $\delta \in (0, 1)$;
3: **Input:** Initial values $v^{(0)}, \sigma^{(0)}$ that satisfies monotonicity:
   \begin{align*}
   v^* &\leq v^{(0)} \leq v^* + u^1, \quad v^{(0)} \geq T[v^{(0)}], \quad \text{and} \quad v^{(0)} \geq T_{a+0}[v^{(0)}];
   \end{align*}
4: **Output:** $\{v^{(i)}, Q^{(i)}, \sigma^{(i)}, \xi^{(i)}\}_{i=0}^{\infty}$ which is an MDVSS with probability at least $1 - \delta$;
5: **INITIALIZATION:**
6: Let $c_1, c_2, c_3, c$ be some tunable absolute constants;
7: \begin{algorithmic}[1]
8: \textbf{Initialize constants:}
9: $\beta \leftarrow (1 - \gamma)^{-1}$, and $R \leftarrow [c_1 \beta \ln(\frac{1}{\delta})]$, $m_1 \leftarrow c_2 \beta^3 \cdot \min(1, u^{-2}) \cdot \log(\frac{\gamma}{\delta})$;
10: $m_2 \leftarrow c_3 \beta^2 \log[2R|S|/\delta^{1-\epsilon}]$;
11: $\alpha_1 \leftarrow L/m_1$, where $L = c \log(|S|/\delta^{1-\epsilon})$;
12: \textbf{Obtain an initial batch of samples:}
13: For each $(s, a) \in S \times A$: obtain independent samples $s_{(1)}, s_{(2)}, ..., s_{(m)}$ from $P(\cdot|s, a)$;
14: **Initialize:** $\mathbf{w}^i = \bar{\mathbf{w}}^i = \bar{\sigma}^i = Q^{(0)} = Q^{(1)} \leftarrow \beta \cdot 1_{S \times A}$ and $i \leftarrow 0$;
15: \textbf{for each} $(s, a) \in S \times A$ \textbf{do}
16: \textbf{Compute empirical estimates of} $P_{s,a} v^{(i)}$ \textbf{and var}($v^{(i)}$)($s,a$):
17: $\bar{\mathbf{w}}^i(s,a) \leftarrow \frac{1}{m_1} \sum_{j=1}^{m_1} v^{(i)}(s_j) s_{(j)}$;
18: $\bar{\sigma}^i(s,a) \leftarrow \frac{1}{m_1} \sum_{j=1}^{m_1} (v^{(i)}(s_{(j)}))^2(s_{(j)}) - (\bar{\mathbf{w}}^i)^2(s,a)$;
19: \textbf{Shift} the empirical estimate to have one-sided error and guarantee monotonicity:
20: $\mathbf{w}^i(s,a) \leftarrow \bar{\mathbf{w}}^i(s,a) + \sqrt{\alpha_1 \bar{\sigma}^i(s,a) + \alpha_1^{3/4} \beta}$
21: \textbf{Compute coarse estimate of the Q-function and make sure its value is in $[0, \beta]$:
22: $Q^{(i)}(s,a) \leftarrow \min[\mathbf{r}(s,a) + \gamma \mathbf{w}^i(s,a), \beta]$
23: \textbf{end for}
24: \textbf{for} $i = 1$ to $R$ \textbf{do}
25: \textbf{Compute the one-step dynamic programming:}
26: Let $v^{(i)} \leftarrow \bar{v}^{(i)} \leftarrow T(Q^{(i-1)})$, $\sigma^{(i)} \leftarrow \bar{\sigma}^{(i)} \leftarrow \sigma(Q^{(i-1)})$;
27: \textbf{Compute strategy and value and maintain monotonicity:}
28: For each $s \in S$ if $v^{(i)}(s) \geq v^{(i-1)}(s)$, then $v^{(i)}(s) \leftarrow v^{(i-1)}(s)$ and $\sigma^{(i)}(s) \leftarrow \sigma^{(i-1)}(s)$;
29: \textbf{Obtain a small batch of samples:}
30: For each $(s, a) \in S \times A$: draw independent samples $s_{(1)}, s_{(2)}, ..., s_{(m)}$ from $P(\cdot|s,a)$;
31: \textbf{Compute the expected value,} $g^{(i)}$, the estimate of $P[v^{(i)} - v^{(0)}]$ with one-sided error:
32: \textbf{Estimate the approximation error:}
33: $\xi^{(i)} \leftarrow 2 \sqrt{\sigma_1 v^{(i)} + 2 [\sigma_1^{3/4} \beta + C(1 - \gamma)] u} \cdot 1$
34: $Q^{(i+1)} \leftarrow \min \left[ \mathbf{r} + \gamma \cdot [\mathbf{w}^i + g^{(i)}], \beta \right]$;
35: \textbf{end for}
36: return $\{v^{(i)}, Q^{(i)}, \sigma^{(i)}, \xi^{(i)}\}_{i=0}^{\infty}$
\end{algorithmic}

Let $\nu^{(j)} \leftarrow \beta/2$ and $\delta \leftarrow 1/\text{poly}(\log(\beta/\epsilon))$.
We run Algorithm 1 repeatedly:

\begin{align*}
(v^{(j+1)}, \sigma^{(j+1)}) \leftarrow \text{QVI-MDVS}
\end{align*}

where $\sigma^{(j)} = (x^{(j)}_{\min}, x^{(j)}_{\max})$ and we take the terminal value and strategy of the output sequence of Algorithm 1 as the input for the next iteration. In total we run $R' = \Theta(\log(\beta/\epsilon))$ iterations. In the end, we output $\pi_{\min}$ from $\sigma^{(R')} = (\pi_{\min}', \pi_{\max}')$ as our min-player strategy.

The formal guarantee of the algorithm is presented in the following theorem.

**Theorem 3.1** (Restatement of Theorem 1.1). Given a stochastic game $\mathcal{G} = (S, A, r, P, \gamma)$ with a generative model, there exists (constructively) an algorithm that outputs, with probability at least $1 - \delta$, an $\epsilon$-optimal strategy $\sigma$ by querying $Z := O(|S| |A| (1 - \gamma)^{-3} \epsilon^{-2})$ samples in time $O(Z)$ using space $O(|S| |A|)$ where $\epsilon \in (0, 1)$ and $O(\cdot)$ hides $\text{poly} \log(|S| |A| / (1 - \gamma) / \epsilon / \delta)$ factors.

The formal proof of Theorem 3.1 is given in the next section. Here we give a sketch of the proof.

**Proof Sketch of Theorem 3.1** We first show the
high-level Structure of Algorithm 1. We would like to approximate \( P^V \). We also compute the empirical variances of the updates. These computations are presented in Line 15-20. To approximate value-iteration while preserving all monotonic properties required by an MDVSS. The basic step of the algorithm is to do approximate estimation error upper bounds to make our estimators one-sided. We additionally maintain carefully-designed strategies in Line 26-28 to preserve monotonicity. Hence the algorithm can be viewed as a value-strategy iteration algorithm.

Correctness of Algorithm 1. We now sketch the proof of correctness for Algorithm 1. Firstly, Proposition A.3 shows that the if an MDVSS, e.g., \( \{ v^{+(i)}, Q^{+(i)}, \sigma^{+(i)}, \epsilon^{+(i)} \}_{R=0}^{R} \), satisfies \( \| v^{+(R)} - v^* \|_{\infty} \leq \epsilon \) for some \( \epsilon > 0 \) and then their terminal strategies and values satisfy

\[
v^{\pi_m(R)} \leq v^{\pi_m(R)} \leq v^* + \epsilon 1.
\]

This indicates that as long as we can show \( \epsilon \leq \eta/2 \), then the halving-error-property (7) holds.

Proposition A.4 shows the halving-error-property can be achieved by setting

\[
\epsilon^{+(i)} \leq \frac{\sqrt{\text{var}(v^{+(0)})}}{m + \text{lower-order terms}},
\]

where \( \text{var}(v^{+(0)}) \) is the variance-of-value vector of \( v^{+(0)} \) and \( m \geq \sqrt{\beta^2 u^{-2}} \). This proof is based on constructing an auxiliary Markovian strategy for analyzing the error accumulation throughout the value-strategy iterations. The Markovian strategy is a time-dependent strategy used as a proxy for analyzing the entrywise error recursion (Lemmas A.4 A.11).

Proposition A.12 shows, with high probability, Algorithm 1 produces value-strategy sequences \( \{ v^{+(i)}, Q^{+(i)}, \sigma^{+(i)}, \epsilon^{+(i)} \}_{R=0}^{R} \), which is indeed an MDVSS and \( \xi^{+(i)} \) satisfies Proposition A.4. The proof involves analyzing the probability of “good events” on which monotonicity is preserved at every iteration by using confidence estimates computed during the iterations and concentration arguments. See Lemmas A.13 A.18 for the full proof of Proposition A.12.

Putting Everything Together. Finally by putting together the strategies, we conclude that the terminal strategy of the iteration is always an approximately optimal min-player strategy to the game, with high probability. For implementation, our algorithm only computes the inner product based on samples, the total computation time is proportional to the number of samples. Moreover, since we can update as samples are drawn and output the monotone sequences as they are generated, we do not need to store samples or the value-strategy sequences, thus the overall space is \( O(|S||A|) \).

\[\square\]
References


A Proof of Main Results

The remainder of this section is devoted to proving Theorem 1.1. We prove this by formally providing a notion of monotone value-strategy sequences. With this, we show if an algorithm outputs some monotone value-strategy sequence, then the terminal strategy of the sequence is always an approximately optimal strategy to the game. We then show that Algorithm 1 produces monotone value-strategy sequences with high probability.

A.1 Additional Notation

First we provide additional notation critical to our proofs.

Markovian Strategies: We denote a Markovian strategy \( \sigma^\infty \) as an infinitely long sequence of pre-defined strategies
\[
\sigma^\infty := (\sigma_1, \sigma_2, \ldots),
\]
where each \( \sigma_i \) is a normal deterministic strategy. We denote
\[
\sigma_t^\infty = (\sigma_t, \sigma_{t+1}, \ldots)
\]
as another Markovian strategy. We denote \( \sigma^\infty_{\min} \) and \( \sigma^\infty_{\max} \) as the min-player strategy and the max-player strategy respectively. When using the strategy, players uses \( \sigma_t \) at time \( t \). The strategy is Markovian because it does not depend on the historical moves. Note that a stationary strategy \( \sigma \) is a special case of the Markovian strategy: \( \sigma = (\sigma, \sigma, \sigma, \ldots) \). The value of a Markovian strategy is defined as before, but the states are generated by playing the action \( \sigma_t(s^t) \) at time \( t \). Since the strategy has a time dependence, we denote
\[
v_t^\infty := v[\sigma_t^\infty] \quad \text{and} \quad Q_t^\infty = r + \gamma P v_{t+1}^\infty.
\]
The (half) Bellman operators are defined similarly to that of stationary policies.

A.2 Monotone Value-Strategy Sequence

In this section we formally define monotone strategy value sequences. Such a sequence, although not explicitly stated in Sidford et al. (2018[b], are crucial for these algorithms to obtain good policy while obtaining a good value for an MDP. In the following sections, we denote \( m \geq 1, L \geq 1 \) and \( \epsilon \in [0, (1 - \gamma)^{-1}] \) as parameters. Monotone value-strategy sequences are formally defined as follows.

Definition A.1 (Monotone Decreasing Value-Strategy Sequence). A monotone decreasing value-strategy sequence (MDVSS) is a sequence of \( \{v^{(i)}, Q^{(i)}, \sigma^{(i)}, \epsilon^{(i)}\}_{i=0}^R \) where \( v^{(i)}, \epsilon^{(i)} \in \mathbb{R}^S \), \( Q^{(i)} \in \mathbb{R}^{S \times A} \) and \( \sigma^{(i)} = (\sigma^{(i)}_{\min}, \sigma^{(i)}_{\max}) \in \mathcal{A}^S \) satisfy

1. \( v^{(0)} \geq v^{(1)} \geq \ldots \geq v^{(R)} \geq v^* \);
2. \( \forall i \in [0, R], T_{\sigma^{(i)}}[v^{(i)}] \leq v^{(i)}, T[v^{(i)}] \leq v^{(i)}, H^{\sigma^{(i)}_{\min}}[v^{(i)}] \leq v^{(i)} ;
3. \( \forall i \in [R], Q^{(i)} \leq r + \gamma P v^{(i-1)} + \epsilon^{(i)} ;
4. \( \forall i \in [R], v^{(i)} \leq V[Q^{(i)}].
\)

Note that \( Q^{(0)}, \epsilon^{(0)} \) can be arbitrary.

Here, we explain the intuition of the sequence. The first property guarantees that the value-estimator \( v^{(i)} \)'s always upper bound the optimal value. The second property guarantees that \( v^{\sigma^{\infty}_{\min}} \leq v^{(i)} \). Indeed
\[
v^{\sigma^{\infty}_{\min}} = \lim_{t \to \infty} \mathcal{H}^t_{\sigma^{\infty}_{\min}}[v^{(i)}] \leq v^{(i)},
\]
where \( \mathcal{H}^t_{\sigma^{\infty}_{\min}} \) denotes applying \( \mathcal{H}_{\sigma^{\infty}_{\min}} \) for \( t \) times. Therefore, as long as \( v^{(R)} - v^* \leq \epsilon 1 \), we have
\[
v^* \leq v^{\sigma^{\infty}_{\min}} \leq v^* + \epsilon 1.
\]
The third and the fourth property guarantees \( v^{(R)} \) is good by requiring that \( v^{(i)} \) and \( Q^{(i)} \) satisfy the approximate value iteration with one-sided error. However the overall error \( v^{(R)} - v^* \) is controlled by the per-step error term \( \epsilon^{(i)} \).

Similarly, we define monotone increasing value-strategy sequence (MIVSS) analogously with every inequality reversed.

**Definition A.2** (Monotone Increasing Value-Strategy Sequence). A monotone increasing value-strategy sequence (MIVSS) is a sequence of \( \{v^{(i)}, Q^{(i)}, \sigma^{(i)}, \epsilon^{(i)}\}_{i=0}^R \) where \( v^{(i)} \in \mathbb{R}^S, Q^{(i)} \in \mathbb{R}^{S \times A} \) and \( \sigma^{(i)} \in \mathcal{A}^S \) that satisfies,

1. \( v^{(0)} \leq v^{(1)} \leq \ldots \leq v^{(R)} \leq v^* \);
2. \( \forall i \in [0, R], \mathcal{T}_{\sigma^{(i)}}[v^{(i)}] \geq v^{(i)}, \mathcal{T}v^{(i)} \geq [v^{(i)}], \mathcal{H}_{\pi_{\max}}[v^{(i)}] \geq v^{(i)} ; \)
3. \( \forall i \in [R], Q^{(i)} \geq r + \gamma P v^{(i-1)} - \epsilon^{(i)} ; \)
4. \( \forall i \in [R], v^{(i)} \geq V[Q^{(i)}] . \)

Note that \( Q^{(0)}, \epsilon^{(0)} \) can be arbitrary.

### A.3 Monotone Value-Strategy Sequence Implies Good Strategy

Next, we show that MDVSS or MIVSS implies a good terminal value/strategy. First we show that if the terminal value \( v^{(R)} \) is close to the optimal value, then we are guaranteed to have good strategies as well.

**Proposition A.3.** Suppose we have an MDVSS, \( \{v^{(i)}, Q^{(i)}, \sigma^{(i)}, \epsilon^{(i)}\}_{i=0}^R \), with \( ||v^{(R)} - v^*||_{\infty} \leq \epsilon \) for some \( \epsilon \geq 0 \). Then we have

\[
v^{(R)}_{\min} \leq v^* + \epsilon \cdot 1.
\]

Similarly, suppose \( \{v^{(i)}, Q^{(i)}, \sigma^{(i)}, \epsilon^{(i)}\}_{i=0}^R \) is an MIVSS, then

\[
v^{(R)}_{\max} \geq v^* - \epsilon \cdot 1.
\]

**Proof.** By the property of an MDVSS, we have

\[
v^{(R)}_{\min} \leq v^{(R)}.
\]

Since \( v^{(R)} \leq v^* + \epsilon \cdot 1 \), we prove the first inequality. The second inequality follows similarly.

Next we consider when it is the case we achieve a good terminal value. The following proposition shows that an MDVSS (MIVSS) with an appropriate error parameters has a better terminal value than its initial value.

**Proposition A.4.** Let \( u \in (0, \beta), \beta = (1 - \gamma)^{-1}, R = \Theta[\beta \log(\beta/u)] \). Suppose an MDVSS (or MIVSS) \( \{v^{(i)}, Q^{(i)}, \sigma^{(i)}, \epsilon^{(i)}\}_{i=0}^R \) satisfies

\[
||v^{(0)} - v^*||_{\infty} \leq u \quad \text{and} \quad \epsilon^{(i)} = \sqrt{L \cdot \text{var}(v^{(0)})/m + \beta \cdot (L/m)^{3/4} + u/(CR)},
\]

for some large constant \( C > 1 \) and \( m \geq 1 \). Then we have

\[
||v^{(R)} - v^*||_{\infty} \leq u/2 \quad \text{for} \quad m = \Omega\left(\frac{1}{\min(1, u^2) \cdot (1 - \gamma)^3}\right).
\]

Note that Proposition A.4 shows that in an MDVSS/MIVSS, the distance to the optimal value of the terminal value reduces by at least half of its initial value. Starting from some \( v^{(0)} \) with distance at most \( \beta \) to \( v^* \), by concatenating \( O(\log(\beta/\epsilon)) \) many MDVSS/MIVSS's, with the initial value of one sequence set as the terminal value of the last sequence, an \( \epsilon \)-optimal value can be obtained. The remainder of this subsection devotes to proving the above proposition. Since MIVSS and MDVSS are symmetric, in the following analysis, we focus on MDVSS and the analysis follows similarly for MIVSS.
A.3.1 Auxiliary Markovian Strategy

Due to the lack of monotonicity we do not know how to use the optimal strategy \( \sigma^* \) to carefully account for the error accumulation of the MDVSS. To resolve this issue, we instead use the following auxiliary Markovian strategy as such a proxy.

**Definition A.5 (Auxiliary Strategy).** Given a MDVSS, \( \{v^{(i)}, Q^{(i)}, \sigma^{(i)}, \epsilon^{(i)}\}_{i=0}^{\infty} \), we denote the Markovian auxiliary strategy for the max-player as

\[
\pi^{\infty(i)}_{aux \max} = (\pi^{(i)}_{aux \max}, \pi^{(i-1)}_{aux \max}, \ldots, \pi^{(1)}_{aux \max}, \pi^*_{max}, \pi^*_{max}, \pi^*_{max}, \ldots),
\]

where \( \pi^{(i)}_{aux \max}(s) = \arg \max_a Q^{(i)}(s,a) \) for \( s \in S_{max} \). We denote the auxiliary strategy for the min-player as

\[
\pi^{\infty(i)}_{aux \min} = \sigma_{min}[\pi^{\infty(i)}_{aux \max}] = (\pi^{(i)}_{aux \min}, \pi^{(i-1)}_{aux \min}, \ldots, \pi^{(1)}_{aux \min}, \pi^*_{min}, \pi^*_{min}, \pi^*_{min}, \ldots),
\]

which is the optimal counter Markovian policy of \( \pi^{\infty(i)}_{aux \max} \), i.e.,

\[
\forall s \in S_{min} : \pi^{\infty(i)}_{aux \min}(s) = \arg \min_a [r(s,a) + \gamma P(\cdot|s,a)^T v[\pi^{\infty(i-1)}_{aux}]].
\]

We also denote

\[
\sigma^{\infty(i)}_{aux} = \left(\pi^{*}_{min}, \pi^{*}_{aux \max}, \pi^{(i-1)}_{aux \max}, \pi^{(i-1)}_{min}, \pi^{(1)}_{aux \max}, \pi^{(1)}_{min}, \pi^*_{max}, \pi^*_{max}, \pi^*_{min}, \ldots\right).
\]

Furthermore, we denote \( \sigma^{(i)}_{aux} = (\pi^{(i)}_{aux \min}, \pi^{(i)}_{aux \max}) \) for \( i \geq 1 \) and \( \sigma^{(i)}_{aux} = \pi^* \) for \( i \leq 0 \).

For a Markovian strategy, we first show that the strategy has a value always smaller than the optimal value.

**Lemma A.6.** For all \( i \in [R] \), we have

\[
v[\sigma^{\infty(i)}_{aux}] \leq v^*.
\]

**Proof.** Denote

\[
\tilde{\sigma}^{\infty(i)}_{aux} = \left(\pi^*_{min}, \pi^{*}_{aux \max}, \pi^{(i-1)}_{aux \max}, \pi^{(i-1)}_{min}, \pi^{(1)}_{aux \max}, \pi^{(1)}_{min}, \pi^*_{max}, \pi^*_{max}, \pi^*_{min}, \ldots\right).
\]

Denote \( \sigma^{\infty(i)}_{aux} = \tilde{\sigma}^{\infty(i)}_{aux} = (\pi^*_{min}, \ldots) \). We first show that for all \( i \in [R] \), \( v[\sigma^{\infty(i)}_{aux}] \leq v[\tilde{\sigma}^{\infty(i)}_{aux}] \). Indeed it holds trivially for \( i = 0 \). Suppose it holds for some \( i \geq 0 \). Then, for each \( s \in S_{min} \), we have,

\[
v[\sigma^{\infty(i)}_{aux}](s) = \min_a [r(s,a) + \gamma P(\cdot|s,a)^T v[\sigma^{\infty(i-1)}_{aux}]]
\]

\[
\leq [r(s,\sigma^*(s)) + \gamma P(\cdot|s,\sigma^*(s))^T v[\sigma^{\infty(i-1)}_{aux}]]
\]

\[
\leq [r(s,\sigma^*(s)) + \gamma P(\cdot|s,\sigma^*(s))^T v[\tilde{\sigma}^{\infty(i-1)}_{aux}]] \quad \text{(due to } v[\sigma^{\infty(i-1)}_{aux}] \leq v[\tilde{\sigma}^{\infty(i-1)}_{aux}] \text{)}
\]

\[
= v[\tilde{\sigma}^{\infty(i)}_{aux}](s).
\]

For each \( s \in S_{max} \), we have,

\[
v[\sigma^{\infty(i)}_{aux}](s) = [r(s,\sigma^{(i)}_{aux}(s)) + \gamma P(\cdot|s,\sigma^{(i)}_{aux}(s))^T v[\sigma^{\infty(i-1)}_{aux}]]
\]

\[
\leq [r(s,\sigma^{(i)}_{aux}(s)) + \gamma P(\cdot|s,\sigma^{(i)}_{aux}(s))^T v[\tilde{\sigma}^{\infty(i-1)}_{aux}]] \quad \text{(due to } v[\sigma^{\infty(i-1)}_{aux}] \leq v[\tilde{\sigma}^{\infty(i-1)}_{aux}] \text{)}
\]

\[
= v[\tilde{\sigma}^{\infty(i)}_{aux}](s).
\]

Now, since \((\pi^*_{max}, \ldots)\) is the optimal counter strategy of \((\pi^*_{min}, \ldots)\), we have

\[
v[\tilde{\sigma}^{\infty(i)}_{aux}] \leq v^*
\]

holds similarly. This concludes the proof. \( \square \)

Consider the error vector \( \epsilon^{(i)} \). Recall that \( \epsilon^{(i)}_{\sigma^{(i)}_{aux}} \) denotes a vector in \( \mathbb{R}^S \) whose \( s \)-th entry is given by \( \epsilon^{(i)}(s,\sigma^{(i)}_{aux}(s)) \). The next lemma shows a recursive relation between a Markovian strategy and the corresponding MDVSS values.
Lemma A.7. For all $i \in [R]$, we have

$$v^{(i)} - v^{[\infty(i)]} \leq \gamma P^\infty(s_{aux}^i (v^{(i-1)} - v^{[\infty(i-1)]})) + e^{(i)}_{\infty}$$

Proof. Note that $v^{(i)} \geq v^* \geq v^{[\infty]}. For each $s \in S_{min}$, we have

$$v^{(i)}(s) \leq \min_a Q^{(i)}(s, a) \leq Q^{(i)}(s, s_{aux}^i(s))$$

$$\leq r(s, s_{aux}^i(s)) + P\{-|s, s_{aux}^i(s)}^\top v^{(i-1)} + e^{(i)}(s, s_{aux}^i(s)),$$

and

$$v^{[\infty]}(s) = r(s, s_{aux}^i(s)) + P\{-|s, s_{aux}^i(s)}^\top v^{[\infty(i-1)]}.$$ 

Thus

$$v^{(i)}(s) - v^{[\infty]}(s) \leq P\{-|s, s_{aux}^i(s)}^\top (v^{(i-1)} - v^{[\infty(i-1)]}) + e^{(i)}(s, s_{aux}^i(s)).$$

Similarly, for each $s \in S_{max}$, we have, $v^{(i)}(s) \leq \max_a Q^{(i)}(s, a) := Q^{(i)}(s, s_{aux}^i(s))$, thus

$$v^{(i)}(s) - v^{[\infty]}(s) \leq Q^{(i)}(s, s_{aux}^i(s)) - v^{[\infty]}(s)$$

$$\leq P\{-|s, s_{aux}^i(s)}^\top (v^{(i-1)} - v^{[\infty(i-1)]}) + e^{(i)}(s, s_{aux}^i(s))$$

as desired. □

With an inductive application of the above lemma, we obtain the following corollary, which states an upper bound between the difference of $v^{(R)}$ and $v^{[\infty]}$. It connects the upper bound with a recursive propagation of the error.

Corollary A.8.

$$v^{(R)} - v^{[\infty]} \leq \sum_{i=1}^R \gamma^{R-i} P^\infty(s_{aux}^R \cdot P^\infty(s_{aux}^{R-1}) \cdot \ldots \cdot P^\infty(s_{aux}) (v^{(0)} - v^*)$$

$$+ \sum_{i=1}^R \gamma^{R-i} P^\infty(s_{aux}^R \cdot P^\infty(s_{aux}^{R-1}) \cdot \ldots \cdot P^\infty(s_{aux}) \cdot e^{(i)}_{\infty}.$$ 

By this corollary, we know that the major error accumulation term is the second term.

A.3.2 Error Accumulation

We now consider the error accumulation in the sequence. As will show shortly, we relate $e^{(i)}$ to the variance vector $\sqrt{\text{var}(v^{[\infty(i-1)]})} \in \mathbb{R}^{S \times A}$, where $\text{var}(v)[s, a] := \text{var}_{s' \sim P(s, a)}[v(s')]$, $\forall (s, a), v$. Therefore, it suffices to consider the following bound.

Lemma A.9.

$$\sum_{i=1}^R \gamma^{R-i} P^\infty(s_{aux}^R \cdot P^\infty(s_{aux}^{R-1}) \cdot \ldots \cdot P^\infty(s_{aux}) \sqrt{\text{var}(v^{[\infty(i-1)]})_{s_{aux}^i}}$$

$$\leq \sqrt{\sum_{i=1}^R \gamma^{2(R-i)} P^\infty(s_{aux}^R \cdot P^\infty(s_{aux}^{R-1}) \cdot \ldots \cdot P^\infty(s_{aux}) \text{var}(v^{[\infty(i-1)]})_{s_{aux}^i}}$$

Proof. Follows from Cauchy-Schwarz and that the $P$ matrices are non-negative with each row summing to 1. □

The following lemma establishes a Bellman-like equation for the variance vector of a Markovian strategy.

Lemma A.10. For any Markovian strategy $\pi^{\infty} = (\pi^{(0)}$, $\pi^{(1)}$, $\ldots )$, we have, for all $s \in S$

$$\text{var}\left[\sum_{t=0}^\infty \gamma^t r(s^t, \pi^{(t)}(s))\right]_{s^0 = s} = \sum_{i=0}^\infty \gamma^{2(t+1)} P^{\infty} \pi^{(0)} \pi^{(1)} \pi^{(2)} \ldots \pi^{(t-1)} \text{var}(v^{[\infty(t+1)]})_{s_{U}} \pi^{(t)} (s)$$

$(8)$
Proof.

\[
\text{var}\left[\sum_{t=0}^{\infty} \gamma^t r(s^t, \pi(t)(s^t)) \bigg| s^0 = s\right] = E\left[\left(\sum_{t=0}^{\infty} \gamma^t r(s^t, \pi(t)(s^t))\right)^2 \bigg| s^0 = s\right] - E\left[\sum_{t=0}^{\infty} \gamma^t r(s^t, \pi(t)(s^t)) \bigg| s^0 = s\right]^2.
\]

For the second term, we have

\[
E\left[\sum_{t=0}^{\infty} \gamma^t r(s^t, \pi(t)(s^t)) \bigg| s^0 = s\right]^2 = v(\pi^{\infty(0)})^2(s) = r(s, \pi^{(0)}(s))^2 + \gamma^2(P^{\pi(0)}v[\pi^{\infty(1)}])^2(s) + 2\gamma r(s, \pi^{(0)}(s))(P^{\pi(0)}v[\pi^{\infty(1)}])(s).
\]

For the first term, we have

\[
\left(\sum_{t=0}^{\infty} \gamma^t r(s^t, \pi(t)(s^t))\right)^2 = r(s, \pi^{(0)}(s))^2 + 2r(s, \pi^{(0)}(s))\left(\sum_{t=1}^{\infty} \gamma^t r(s^t, \pi(t)(s^t))\right) + \left(\sum_{t=1}^{\infty} \gamma^t r(s^t, \pi(t)(s^t))\right)^2.
\]

Note that

\[
E\left[\left(\sum_{t=1}^{\infty} \gamma^t r(s^t, \pi(t)(s^t))\right)^2 \bigg| s^0 = s\right] = E\left[\left(\sum_{t=1}^{\infty} \gamma^t r(s^t, \pi(t)(s^t))\right)^2 \bigg| s^0 = s\right] - \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)v[\pi^{\infty(1)}](s') + \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)v[\pi^{\infty(1)}](s')
\]

\[
= \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)\text{var}\left[\sum_{t=0}^{\infty} \gamma^t r(s^{t+1}, \pi(t+1)(s)) \bigg| s^1 = s'\right] + \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)v[\pi^{\infty(1)}](s')
\]

Combining the above two equations, we have,

\[
E\left[\left(\sum_{t=0}^{\infty} \gamma^t r(s^t, \pi(t)(s^t))\right)^2 \bigg| s^0 = s\right] = r(s, \pi^{(0)}(s))^2 + 2\gamma r(s, \pi^{(0)}(s))\sum_{s'} P^{\pi(0)}(s'|s)v[\pi^{\infty(1)}](s')
\]

\[
+ \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)\text{var}\left[\sum_{t=0}^{\infty} \gamma^t r(s^{t+1}, \pi(t+1)(s)) \bigg| s^1 = s'\right] + \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)v[\pi^{\infty(1)}](s')
\]

We thus obtain

\[
\text{var}\left[\sum_{t=0}^{\infty} \gamma^t r(s^t, \pi(t)(s^t)) \bigg| s^0 = s\right] = \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)\text{var}\left[\sum_{t=0}^{\infty} \gamma^t r(s^{t+1}, \pi(t+1)(s)) \bigg| s^1 = s'\right]
\]

\[
+ \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)v[\pi^{\infty(1)}](s') - 2\gamma^2 (P^{\pi(0)}v[\pi^{\infty(1)}])^2(s)
\]

\[
= \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)\text{var}\left[\sum_{t=0}^{\infty} \gamma^t r(s^{t+1}, \pi(t+1)(s)) \bigg| s^1 = s'\right] + \gamma^2 \text{var}(v[\pi^{\infty(1)}])_{\pi^{(0)}}
\]

Let LHS and RHS be the left hand side and right hand side of \(8\) respectively. Then we have,

\[
LHS = \gamma^2 \sum_{s'} P^{\pi(0)}(s'|s)\text{var}\left[\sum_{t=0}^{\infty} \gamma^t r(s^{t+1}, \pi(t+1)(s)) \bigg| s^1 = s'\right] + \gamma^2 \text{var}(v[\pi^{\infty(1)}])_{\pi^{(0)}}
\]

\[
= \gamma^4 \sum_{s', s''} P^{\pi(0)}(s'|s')P^{\pi(1)}(s''|s')\text{var}\left[\sum_{t=0}^{\infty} \gamma^t r(s^{t+2}, \pi(t+2)(s)) \bigg| s^2 = s''\right] + \gamma^4 P^{\pi(0)}\text{var}(v[\pi^{\infty(2)}])_{\pi^{(1)}}
\]

\[+ \gamma^2 \text{var}(v[\pi^{\infty(1)}])_{\pi^{(0)}}. \]

Applying the above equality recursively for \(\text{var}(v[\pi^{\infty(1)}])\) completes the proof.
Based on the above two lemmas, we immediately obtain the following worst-case bound for the error accumulation.

**Corollary A.11.**

\[
\sqrt{R \sum_{i=1}^{R} \gamma^{2(R-i)} P^{(R)}_{\text{aux}} \cdot P^{(R-1)}_{\text{aux}} \cdot \ldots \cdot P^{(i+1)}_{\text{aux}} \var(\sigma^{\infty(i-1)}_{\text{aux}}))_{\sigma_{\text{aux}}}} \leq \sqrt{\frac{R}{\gamma^2 (1 - \gamma)^2}}.
\]

**Proof.** We use that

\[
\sum_{i=0}^{\infty} \gamma^i R(s, \sigma^{(i)}_{\text{aux}}(s)) \leq (1 - \gamma)^{-1}
\]

Since \(\sum_{i=0}^{\infty} \gamma^i R(s, \sigma^{(i)}_{\text{aux}}(s)) \leq (1 - \gamma)^{-1}\), we have var(\(\sigma^{\infty(i-1)}_{\text{aux}}\)) \(\leq (1 - \gamma)^{-2}\) as desired. \(\square\)

### A.3.3 Putting Everything Together

**Proof of Proposition A.4.** By Corollary A.8, we have,

\[
v^{(R)} - v^{*} \leq v^{(R)} - v^{(\sigma(0))} \leq \sum_{i=1}^{R} \gamma^{R-i} P^{(R)}_{\text{aux}} \cdot P^{(R-1)}_{\text{aux}} \cdot \ldots \cdot P^{(i+1)}_{\text{aux}} (v^{(0)} - v^{*}) + \sum_{i=1}^{R} \gamma^{R-i} P^{(R)}_{\text{aux}} \cdot P^{(R-1)}_{\text{aux}} \cdot \ldots \cdot P^{(i+1)}_{\text{aux}} \epsilon^{(i)}_{\sigma_{\text{aux}}}
\]

where the first inequality holds for sufficiently large \(R\). Consider the second term. Since

\[
e^{(i)} = \sqrt{L \cdot \text{var}(v^{(0)})/m + \beta \cdot (L/m)^{3/4} + u/(CR)}.
\]

We bound

\[
\sum_{i=1}^{R} \gamma^{R-i} P^{(R)}_{\text{aux}} \cdot P^{(R-1)}_{\text{aux}} \cdot \ldots \cdot P^{(i+1)}_{\text{aux}} \cdot \beta \cdot (L/m)^{3/4} \leq R\beta \cdot (L/m)^{3/4}
\]

and

\[
\sum_{i=1}^{R} \gamma^{R-i} P^{(R)}_{\text{aux}} \cdot P^{(R-1)}_{\text{aux}} \cdot \ldots \cdot P^{(i+1)}_{\text{aux}} \cdot u/(CR) \leq Ru/(CR).
\]

We thus have,

\[
\gamma^{R-i} P^{(R)}_{\text{aux}} \cdot P^{(R-1)}_{\text{aux}} \cdot \ldots \cdot P^{(i+1)}_{\text{aux}} \sqrt{L \cdot \text{var}(v^{(0)})/m + R\beta \cdot (L/m)^{3/4} + R \cdot u/(CR)}.
\]

Note that

\[
\sqrt{\text{var}(v^{(0)})/m} \leq \sqrt{\text{var}(v^{(i-1)})/m} + \|v^{(i-1)} - v^{(0)}\|\infty.
\]

Now consider

\[
\|v^{(i-1)} - v^{(0)}\|\infty \leq \|v^{(0)} - v^{(R)}\|\infty + \|v^{(i-1)} - v^{(i-1)}\|\infty.
\]
We bound \( \|v(0) - v^R\|_\infty \leq u \). Applying Corollary A.8 again, we have
\[
\|v^{(i-1)} - v^{(i-1)}\|_\infty \leq u/4 + \sum_{i=1}^{R} \gamma^{R-i} P_{\text{aux}}^{(R)} \cdot P_{\text{aux}}^{(R-1)} \cdot \ldots \cdot P_{\text{aux}}^{(i+1)} \sigma_{\text{aux}}^{(i)}
\]
\[
\leq u/4 + \sum_{i=1}^{R} \gamma^{R-i} P_{\text{aux}}^{(R)} \cdot P_{\text{aux}}^{(R-1)} \cdot \ldots \cdot P_{\text{aux}}^{(i+1)} \sqrt{L \cdot \text{var}(v(0))_{\sigma_{\text{aux}}} / m}
\]
\[
+ R \beta \cdot (L/m)^{3/4} + R \cdot u/(CR)
\]

With a natural bound, \( \text{var}[v(0)] \leq \beta^2 \cdot 1 \), we have
\[
\|v^{(i-1)} - v^{(0)}\|_\infty \leq u + \|v^{(i-1)} - v^{(i)}\|_\infty \leq R \beta \sqrt{L/m} + R \beta (L/m)^{3/4} + u/C'
\]
for some constant \( C' > 0 \). Therefore,
\[
v^{(R)} - v^* \leq \sum_{i=1}^{R} \gamma^{R-i} P_{\text{aux}}^{(R)} \cdot P_{\text{aux}}^{(R-1)} \cdot \ldots \cdot P_{\text{aux}}^{(i+1)} \sqrt{L \cdot \text{var}(v^{(i-1)})_{\pi_{\text{aux}}} / m}
\]
\[
+ R \left( R \beta \sqrt{L/m} + R \beta \left( L/m \right)^{3/4} + \frac{u}{C'} \right) \cdot \sqrt{L/m} + \frac{u}{4} + \frac{u}{C} + R \beta \left( L/m \right)^{3/4}
\]
\[
\leq \sqrt{LR \beta^2 / \gamma^2 m} + R \left( R \beta \sqrt{L/m} + R \beta \left( L/m \right)^{3/4} + \frac{u}{C'} \right) \cdot \sqrt{L/m} + \frac{u}{4} + \frac{u}{C} + R \beta \left( L/m \right)^{3/4}
\]
\[
\leq u/2
\]
for large enough constants, \( C \), and that in \( m \).

A.4 Algorithm that Computes a Monotone Sequence

Here we show that Algorithm 1 or Algorithm 2 computes an MDVSS or MIVSS respectively.

Proposition A.12. Let \( u \in (0, \beta] \), \( \beta = (1 - \gamma)^{-1}, \delta \in (0, 1) \), and \( R = \Theta(\beta \log(\beta/u)) \). Further, let \( L = \Theta(\log(\delta^{-1}) R |S||A|) \), and \( m = \Omega(\beta^3 \cdot \max(u^{-2}, 1) \cdot \log(|S||A| \delta^{-1})^2) \). Then there exists an algorithm, on input a stochastic game with a generative model with a sampling oracle, \( \mathcal{G} = (S := S_{\min} \cup S_{\max}, P, r, \gamma) \), a value-strategy pair \((\pi(0), v(0))\) satisfying \( T_{\pi(0)}[v(0)] \leq v(0), T[u(0)] \leq v(0) \) (or \( T_{\pi(0)}[v(0)] \geq v(0), T[u(0)] \geq v(0) \)), and \( \|v(0) - v^*\|_\infty \leq u \) for some \( u > 0 \), outputs, with probability at least \( 1 - \delta \), an MDVSS (or MIVSS) \{\( v(i), Q(i), \pi(i), \epsilon(i) \}_{i=0}^{R} \} by querying
\[
Z = O[|S||A| \cdot (m + R \beta^2 \log|R|S||A| \delta^{-1})]
\]
samples, where
\[
\epsilon(i) = \sqrt{L \sigma_{v(0)} / m} + \beta \cdot (L/m)^{3/4} + u/(CR),
\]
for some large constant \( C > 1 \) and \( \sigma_{v(0)} \) is the variance vector for vector \( v(0) \). The algorithms uses space \( O(|S||A|) \) and halts in time \( O(Z) \).

This section is devoted to proving Proposition A.12 The algorithm of obtaining MDVSS and MIVSS is provided in Algorithm 1 and 2.

The Good Events  Suppose we are given an arbitrary input vector \( v^{-}(0), v^{+}(0) \in [0, (1 - \gamma)^{-1} S \text{ with } v^* - u \leq v^{-}(0) \leq v^+ + u, v^{-}(0) \leq T[v^{-}(0)], v^{+}(0) \leq T[v^+(0)], T[v^+(0)] \leq v^{+}(0) \text{ and } T[v^{-}(0)] \leq v^{-}(0). \]

Since the algorithm is randomized, to begin our analysis, we define a sequence of events for the iterates. We will show that these events happen with high probability via concentration inequalities.

Definition A.13. Let \( \tilde{w}^- \) and \( \tilde{w}^+ \) be the estimate defined in Line 16 (of Algorithm 1 or Algorithm 2 respectively). Denote \( \alpha_{1} \leftarrow L/m_{1} \leq 1 \). Let \( \mathcal{E}_0 \) be the event that
\[
|\tilde{w}^\pm - P v^{\pm}(0)| \leq \sqrt{\alpha_{1} \sigma_{v^{\pm}(0)} + \alpha_{1}^{3/4} \cdot \|v^{\pm}(0)\|_\infty \cdot 1}. \quad (9)
\]
For each $i > 0$, let $g^{\pm(i)}$ be given in Line 32. Let $E_i$ be the event that
\[
|g^{\pm(i)} - P[v^{\pm(i)} - v^{\pm(0)}]| \leq C(1 - \gamma)u \cdot 1.
\]
for some sufficiently small constant $C > 0$.

**Lemma A.14.** For some sufficiently large constant $c$ in $L$, $Pr[E_0] \geq 1 - O(\delta/R)$.

**Proof.** Note that $\|v^{\pm(0)}\|_\infty \leq (1 - \gamma)^{-1}$. By a straightforward application of a Hoeffding bound and Bernstein inequality and a union bound over all $(s, a)$, we reach the desired inequality. More details can be found in the proof of Lemma 5.1 in [Sidford et al. (2018a)].

**The Implications of the Good Events** We now illustrate the consequences of these good events.

**Lemma A.15** (Implications of $E_0$). On $E_0$, we have, for all $(s, a) \in S \times A$,
\[
0 \leq r(s, a) + \gamma P[v(s, a)] = 2 \sqrt{\alpha_1 \sigma_{v^{\pm(0)}}^2 + 2\alpha_1^{3/4}} \|v^{\pm(0)}\|_\infty \quad \text{and}
\]
\[
0 \leq Q^{-}(s, a) \leq Q^{+}(s, a),
\]
and
\[
0 \leq Q^{-}(s, a) - r(s, a) - \gamma P[v(s, a)] \|v^{\pm(0)}\|_\infty \leq 2 \sqrt{\alpha_1 \sigma_{v^{\pm(0)}}^2 + 2\alpha_1^{3/4}} \|v^{\pm(0)}\|_\infty \quad \text{and}
\]
\[
Q^{-}(s, a) \leq Q^{+}(s, a) \leq \beta,
\]
where $\alpha_1 = L/m_1$.

**Proof.** We prove the first inequality and the secondary inequality follows similarly. Condition on $E_0$, we have
\[
|r + \gamma w - r - \gamma P[v^{\pm(0)}]| \leq \sqrt{\alpha_1 \sigma_{v^{\pm(0)}}^2 + \alpha_1^{3/4}} \|v^{\pm(0)}\|_\infty.
\]
Since
\[
Q^{-}(0) = \max [r + \gamma w - \sqrt{\alpha_1 \sigma_{v^{\pm(0)}}^2 + \alpha_1^{3/4}} \|v^{\pm(0)}\|_\infty, 0],
\]
we have
\[
0 \leq r + \gamma P[v^{\pm(0)}] - Q^{-}(0) \leq 2 \sqrt{\alpha_1 \sigma_{v^{\pm(0)}}^2 + \alpha_1^{3/4}} \|v^{\pm(0)}\|_\infty.
\]
Moreover, since $v^{0} \leq v^{\ast}$, we have
\[
Q^{-}(0) \leq r(s, a) + \gamma P[v(s, a)] = \|v^{-}(0) \leq Q^{+},
\]
completing the proof.

**Lemma A.16** (Implications of $E_i$, (1)). Then for any $i > 0$, conditioning on $E_0, E_1, \ldots, E_R$, we have $\{v^{-}(i), Q^{-}(i), \pi^{-}(i), \xi^{-}(i)\} R_{i=0}$ is an MIVSS and $\{v^{+}(i), Q^{+}(i), \pi^{+}(i), \xi^{+}(i)\} R_{i=0}$ is an MDVSS where
\[
\xi^{\pm(i)} = 2 \sqrt{\alpha_1 \sigma_{v^{\pm(0)}}^2 + 2\alpha_1^{3/4}} \beta + C(1 - \gamma)u \cdot 1
\]
for some sufficiently small $C > 0$.

**Proof.** We prove the first part of the lemma, i.e., $\{v^{-}(i), Q^{-}(i), \pi^{-}(i), \xi^{-}(i)\} R_{i=0}$ is an MIVSS. Then the second part follows similarly. It is clear from the definition (Line 32) that
\[
v^{-}(0) \leq v^{-}(1) \ldots \leq v^{-}(R).
\]
To prove property 1 of MIVSS, we need additionally to show
\[
v^{-}(R) \leq v^{\ast}.
\]
This follows if property 2, i.e.,
\[
\forall i \in [0, R] : \quad v^{-}(i) \leq \mathcal{T}_{\pi^{-}(i)}[v^{-}(i)], \quad v^{-}(i) \leq \mathcal{T}[v^{-}(i)].
\]
Indeed,
\[ v^{(i)} \leq T[v^{(i)}] \leq T^2[v^{(i)}] \leq T^\infty[v^{(i)}] = v^*. \]

We now prove property 2 by induction on \( i \). It immediately follows from the initial condition that
\[ v^{(0)} \leq T_{\pi_{(0)}}[v^{(0)}], \quad v^{(0)} \leq T[v^{(0)}]. \]

Suppose this property holds for all 0, 1, \ldots, \( i-1 \) for some \( i > 1 \). We now consider the case \( i \). Let \( \tilde{Q}^{(i)} = r + \gamma P v^{(i-1)} \). Since \( \|\tilde{g}^{(i)} - P(v^{(i-1)} - v^{(0)})\|_\infty \leq C(1 - \gamma)\epsilon \), we have
\[ |\tilde{Q}^{(i)} - r - \gamma w^{(i-1)} - \gamma g^{(i-1)}| \leq \xi^{(i)}/2. \tag{11} \]

Since
\[ Q^{(i)}(s,a) = \max \left[ r(s,a) + \gamma w^{(i-1)}(s,a) + \gamma g^{(i-1)}(s,a), 0 \right] \leq r(s,a) + \gamma P(\pi|s,a)^T v^{(i-1)} \]
we have, for any \( s \in S \)
\[ \min_a Q^{(i)}(s,a) \leq \min_a \tilde{Q}^{(i)}(s,a) \quad \text{and} \quad \max_a Q^{(i)}(s,a) \leq \max_a \tilde{Q}^{(i)}(s,a). \]

For each \( s \in S \), denote
\[ \tilde{v}^{(i)}(s) = Q^{(i)}(s, \tilde{\pi}^{(i)}(s)), \]
where \( \tilde{\pi}^{(i)} \) is given in Line 26 (that achieves \( \max_a Q^{(i)}(s,a) \) or \( \min_a Q^{(i)}(s,a) \)). To show \( v^{(i)} \leq T[v^{(i)}] \) and \( v^{(i)} \leq T_{\pi^{(i)}}[v^{(i)}] \), we do a case analysis for a state in \( S \). Firstly, we consider state \( s \in S_{\min} \). For each state \( s \in S_{\min} \), note that \( \tilde{\pi}^{(i)}(s) := \arg\min_a Q^{(i)}(s,a) \). By Line 32 \( v^{(i)}(s) \) and \( \pi^{(i)}(s) \) have the following two possibilities,

1. \( v^{(i-1)}(s) \leq \tilde{v}^{(i)}(s) \Rightarrow v^{(i)}(s) = \tilde{v}^{(i)}(s) \) and \( \pi^{(i)}(s) = \tilde{\pi}^{(i)}(s) \);
2. \( v^{(i-1)}(s) > \tilde{v}^{(i)}(s) \Rightarrow v^{(i)}(s) = v^{(i-1)}(s) \) and \( \pi^{(i)}(s) = \pi^{(i-1)}(s) \).

Considering case 1, we have,
\[ v^{(i)}(s) = Q^{(i-1)}(s, \pi^{(i-1)}(s)) \leq \min_a \tilde{Q}^{(i-1)}(s,a) = T[v^{(i-1)}](s) \leq T[v^{(i)}](s) \quad \text{and} \]
\[ v^{(i)}(s) = Q^{(i-1)}(s, \pi^{(i-1)}(s)) \leq \tilde{Q}^{(i)}(s, \pi^{(i)}(s)) = T_{\pi^{(i)}}[v^{(i-1)}](s) \leq T_{\pi^{(i)}}[v^{(i)}](s). \]

Considering case 2., we have, by induction hypothesis \( v^{(i-1)}(s) \leq T[v^{(i-1)}](s) \), \( v^{(i-1)}(s) \leq T_{\pi^{(i-1)}}[v^{(i-1)}](s) \). Thus
\[ v^{(i)}(s) \leq T[v^{(i-1)}](s) \leq T[v^{(i)}](s), \]
\[ v^{(i)}(s) \leq T_{\pi^{(i-1)}}[v^{(i-1)}](s) \leq T_{\pi^{(i-1)}}[v^{(i)}](s) = T_{\pi^{(i)}}[v^{(i)}](s). \]

It follows similarly for the case of \( s \in S_{\max} \) (by just replacing the min by max in the above argument). This completes the induction step and hence the property 2.

We now prove property 3 and 4. By Equation 11 we immediately have,
\[ Q^{(i)} \geq r + \gamma P(v^{(i)}) - \xi^{(i)} \]
proving property 3. Lastly, by Line 28 we have
\[ v^{(i)} \geq T[Q^{(i)}], \]
completing the proof of property 4 and the lemma. 
\[ \square \]
The Probability of Good Events  Note that the random samples in the successive improvement phase are independent with event $E_0$. We then have the following lemma.

**Lemma A.17.** Suppose the algorithm does not halt at iteration $i \geq 1$, then,

$$\Pr[E_i|E_0, E_1, \ldots, E_{i-1}] \geq 1 - O(\delta/R).$$

*Proof.* On $E_0, E_1, \ldots, E_{i-1}$, we have $v^{-}(0) \leq v^{-}(i) \leq v^{*} \leq v^{+}(i) \leq v^{+}(0)$, thus $\|v^{\pm}(i) - v^{\pm}(0)\| \leq u$. Applying Hoeffding bound, Bernstein’s inequality and a union bound over all $(s,a)$, we have that with probability at least $1 - O(\delta/R)$, $\|g^{(i)} - P[v^{(i)} - v^{(0)}]\|_{\infty} \leq (1 - \gamma)e/32$, completing the proof.

Therefore, we have the following lemma.

**Lemma A.18.** Let $R = \lceil c_1 \beta \ln[\beta e^{-1}] \rceil$ be an integer for some constant $c_1$. Then, with probability at least $1 - O(\delta)$, $E_0, \{E_i\}_{i=1}^{R}$ all happen.

*Proof.* By a straightforward calculation, we have

$$\Pr[\cap_{i=0}^{R} E_i] = \Pr[E_0] \Pr[E_1|E_0] \Pr[E_2|E_0, E_1] \ldots \Pr[E_R|E_0, E_1, \ldots, E_{R-1}] \geq 1 - O(\delta).$$

Putting It Together

*Proof of Proposition A.12* Let $E = \cap_{i=0}^{R} E_i$, we have shown, on $E$, the outputs of Algorithm 1 and 2

$\{v^{-(i)}, Q^{-(i)}, \pi^{-(i)}, \xi^{-(i)}\}_{i=0}^{R}$ is an MIVSS and $\{v^{+(i)}, Q^{+(i)}, \pi^{+(i)}, \xi^{+(i)}\}_{i=0}^{R}$ is an MDVSS. By the above lemmas, $E$ happens with probability at least $1 - \Theta(\delta)$. This completes the proof of the proposition.

*Proof of Theorem 1.1* The theorem is proved by combining Proposition A.3, A.4 and A.12.
Algorithm 2 QVI-MIVSS: algorithm for computing monotone increasing value-strategy sequences.

1: \textbf{Input:} A generative model for stochastic game \( \mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \gamma) \);
2: \textbf{Input:} Precision parameter \( u \in [0, (1 - \gamma)^{-1}] \) and error probability \( \delta \in (0, 1) \);
3: \textbf{Input:} Initial values \( v^{-(0)}, \pi^{-(0)} \) that satisfies monotonicity:
   \[ v^* - u \mathbf{1} \leq v^{-(0)} \leq v^*, \quad v^{-(0)} \leq T v^{-(0)}, \quad \text{and} \quad v^{-(0)} \leq T_{m-a} v^{-(0)}; \]
4: \textbf{Output:} \( \{v^{-(i)}, Q^{-(i)}, \pi^{-(i)}, \xi^{-(i)}\}_i \).

5: \textbf{INITIALIZATION:}
6: Let \( c_1, c_2, c_3, c \) be some tunable absolute constants;
7: \textbf{Initialize constants:}
8: \[ \beta \leftarrow (1 - \gamma)^{-1}, \text{and } R \leftarrow [c_1 \beta \ln[b u^{-1}]]; \]
9: \[ m_1 \leftarrow c_2 \beta^2 \cdot \text{min}(1, u^{-2}) \cdot \log(8|\mathcal{S}||\mathcal{A}|\delta^{-1}); \]
10: \[ m_2 \leftarrow c_3 \beta^2 \log(2|\mathcal{S}||\mathcal{A}|\delta^{-1}); \]
11: \[ \alpha_1 \leftarrow L/m_1 \text{ where } L = c \log(|\mathcal{S}| |\mathcal{A}|^{-1}(1 - \gamma)^{-1} \cdot \epsilon_1); \]
12: \textbf{Obtain an initial batch of samples:}
13: For each \( (s, a) \in \mathcal{S} \times \mathcal{A} \): obtain independent samples \( s^{(1)}_{s,a}, s^{(2)}_{s,a}, \ldots, s^{(m_1)}_{s,a} \) from \( P(|s,a); \)
14: Initialize: \( w^* = \bar{w} = \bar{\sigma} = Q^{-(0)} = Q^{-(0)} \leftarrow 0; \)
15: end for
16: for each \( (s, a) \in \mathcal{S} \times \mathcal{A} \) do
17: \textbf{Compute empirical estimates of } \( P_{s,a} [v^{-(0)} \text{ and var}(v^{-(0)}); s,a]; \)
18: \[ \bar{w}^{-(s,a)} = \frac{1}{m_1} \sum_{j=1}^{m_1} v^{-(0)}(s^{(j)}_{s,a}); \]
19: \[ \bar{\sigma}^{-(s,a)} = \frac{1}{m_1} \sum_{j=1}^{m_1} (v^{-(0)}(s^{(j)}_{s,a}) - (\bar{w})^2(s,a); \]
20: \textbf{Shift the empirical estimate to have one-sided error and guarantee monotonicity:}
21: \[ w^{-(s,a)} = \bar{w}^{-(s,a)} - \sqrt{\alpha_1 \bar{\sigma}^{-(s,a)} - \alpha_1^{3/4} \beta}; \]
22: \textbf{Compute coarse estimate of the }Q\text{-function and make sure its value is in } [0, \beta]:
23: \[ Q^{-(1)}(s,a) = \text{clip}[r(s,a) + \gamma w^{-(s,a)}, 0, \beta]; \]
24: end for
25: end for
26: \textbf{REPEAT: } \textbf{successively improve:}
27: for \( i = 1 \) to \( R \) do
28: \textbf{Compute the one-step dynamic programming:}
29: Let \( v^{-(i)} \leftarrow \bar{v}^{-(i)} \leftarrow T Q^{-(i-1)}, \quad \pi^{-(i)} \leftarrow \pi^{-(i)}(Q^{-(i-1)}); \)
30: \textbf{Compute strategy and value and maintain monotonicity:}
31: For each \( s \in \mathcal{S} \):
32: if \( v^{-(i)}(s) \leq v^{-(i-1)}(s) \), then \( v^{-(i)}(s) \leftarrow v^{-(i-1)}(s) \) and \( \pi^{-(i)}(s) \leftarrow \pi^{-(i-1)}(s); \)
33: \textbf{Obtaining a small batch of samples:}
34: For each \( (s, a) \in \mathcal{S} \times \mathcal{A} \): draw independent samples \( s^{(1)}_{s,a}, s^{(2)}_{s,a}, \ldots, s^{(m_2)}_{s,a} \) from \( P(|s,a); \)
35: \textbf{Compute the expected value, }\( g^{-(i)}(s,a) \), the estimate of }\( P[s_{s,a}^{(m_2)}] \text{ with one-sided error:}
36: Let \( \bar{g}^{-(i)}(s,a) \leftarrow \frac{1}{m_2} \sum_{j=1}^{m_2} [v^{-(i)}(s^{(j)}_{s,a}) - v^{-(0)}(s^{(j)}_{s,a})]; \)
37: Let \( g^{-(i)}(s,a) \leftarrow \bar{g}^{-(i)}(s,a) - C(1 - \gamma)u, \text{ where } C \text{ is sufficiently small;} \)
38: \textbf{Estimate the approximation error:}
39: \[ \xi^{-(i)} \leftarrow 2\sqrt{\alpha_1 \bar{\sigma}^{-(i-1)} + 2[a_1^{3/4} \beta + C(1 - \gamma)u] \cdot 1}; \]
40: \textbf{Improve }\( Q^{-(i)} \) \text{ make sure its value is in } [0, \beta]:
41: \[ Q^{-(i+1)} \leftarrow \text{clip}[r + \gamma \cdot [w^{-(i)} + g^{-(i)}], 0, \beta]; \]
42: end for
43: return \( \{v^{-(i)}, Q^{-(i)}, \pi^{-(i)}, \xi^{-(i)}\}_i \).