Approximate Dynamic Programming for Two-Player Zero-Sum Markov Games

Julien Perolat(1)
Bruno Scherrer(2)
Bilal Piot(1)
Olivier Pietquin(1,3)

(1) Univ. Lille, CRIStAL, SequeL team, France
(2) Inria, Villers-lès-Nancy, F-54600, France
(3) Institut Universitaire de France (IUF), France

Abstract

This paper provides an analysis of error propagation in Approximate Dynamic Programming applied to zero-sum two-player Stochastic Games. We provide a novel and unified error propagation analysis in \( L_p \)-norm of three well-known algorithms adapted to Stochastic Games (namely Approximate Value Iteration, Approximate Policy Iteration and Approximate Generalized Policy Iteration). We show that we can achieve a stationary policy which is \( 2\gamma\epsilon + \epsilon' \) \(-\)-optimal, where \( \epsilon \) is the value function approximation error and \( \epsilon' \) is the approximate greedy operator error. In addition, we provide a practical algorithm (AGPI-Q) to solve infinite horizon \( \gamma \)-discounted two-player zero-sum Stochastic Games in a batch setting. It is an extension of the Fitted-\( Q \) algorithm (which solves Markov Decision Processes from data) and can be non-parametric. Finally, we demonstrate experimentally the performance of AGPI-Q on a simultaneous two-player game, namely Alesia.

1. Introduction

A wide range of complex problems (e.g. computer networks, human-computer interfaces, games as chess or checkers) can be addressed as multi-agent systems. This is why Multi-Agent Reinforcement Learning (MARL) (Busoniu et al., 2008) has received a growing interest in the last few years. While, in decision theory, Markov Decision Processes (MDPs) (Puterman, 1994) are widely used to control a single agent in a complex environment, Markov Games (MGs) (also named Stochastic Games (SGs)) (Shapley, 1953) are an extension of this formal framework to describe multi-agent systems. As for MDPs, SGs constitute a model for MARL. (Littman, 1994) and have been largely studied in past years (Hu & Wellman, 2003; Greenwald et al., 2003; Bowling & Veloso, 2001). In some sense, SGs are a generalization of both the game-theory and the MDP frameworks where the agents may have different pay-offs (rewards in the MDP vocabulary) they aim at maximizing.

This paper contributes to solving large scale games with unknown dynamics. Especially, it addresses the problem of finding the Nash equilibrium for infinite horizon \( \gamma \)-discounted two-player zero-sum SGs in an approximate fashion, possibly from batch data. SGs are modeled as extensions of MDPs where Dynamic Programming (DP) commonly serves as a basis to a wide range of practical solutions (Puterman, 1994). However, when the size of the game is too large or when its dynamics is not perfectly known, exact solutions cannot be computed. In that case, Approximate Dynamic Programming (ADP) approaches are preferred as for MDPs (Bertsekas, 1995). Because DP relies on an iterative schema, interleaving Value Function Approximation and Greedy Operator Approximations, two types of errors (corresponding to each source of approximation) are introduced and can accumulate over iterations when passing to ADP.

More specifically, this paper provides a theoretical error propagation analysis in \( L_p \)-norm of well-known ADP algorithms applied to games such as Approximate Value Iteration (AVI), Approximate Policy Iteration (API) and Approximate Generalized Policy Iteration (AGPI) (described later). It also proposes a novel algorithm, named Approximate Generalized Policy Iteration-Q (AGPI-Q) extending Fitted-\( Q \) iteration (Ernst et al., 2005) which is further analyzed in terms of error and complexity. AGPI-Q is evaluated on a simultaneous two-player game, namely Alesia.
1.1. Dynamic Programming techniques for MDP
ADP for MDPs has been the topic of many studies these last two decades. Bounds in $L_\infty$ can be found in (Bertsekas, 1995) while $L_p$-norm ones were published in (Munos & Szepesvári, 2008) and (Farahmand et al., 2010). Because approximations often come from the use of supervised-learning algorithms, $L_p$-norm bounds are a significant improvement over $L_\infty$ ones. Indeed, they can rely on upper bounds provided in the supervised learning literature to estimate the overall error on the learnt policy. For a unified analysis of PI, VI and MPI in $L_p$-norm see Scherrer et al. (2012). It allows evaluating several practical algorithms like Fitted-Q iteration (Ernst et al., 2005; Antos et al., 2008), Classification-based MPI (Lagoudakis & Parr, 2003b; Lazaric et al., 2010; Gabillon et al., 2011) and LSPI (Lagoudakis & Parr, 2003a).

1.2. Dynamic Programming techniques for Stochastic Games
Zero-sum two-player $\gamma$-discounted SGs are quite close to $\gamma$-discounted MDPs. In fact, the development of DP techniques for SGs has followed closely the development of DP for MDPs (Shapley, 1953). Most of these algorithms are detailed in Patek (1997) in the framework of Stochastic Shortest Path Games (SSPG). Zero-sum two-player discounted SG is a subclass of SSPG. Similar algorithms exist to solve those games including PI (Patek, 1997), VI (Shapley, 1953) and MPI (called also Generalized Policy Iteration (GPI) (Patek, 1997) in the context of SSPG). Yet, there has been very little attention paid to ADP for SGs after Patek’s work except from Lagoudakis & Parr (2002). To our knowledge, no analyses of the approximate version of those algorithms exist in $L_p$-norm.

2. Framework
A SG is a generalization of an MDP to a $n$-player setting. Each player has a control on the SG through a set of actions. At each step of the game, all players simultaneously choose an action. The reward each player gets after one step depends on the state and on the joint action of all players. Furthermore, the dynamics of the SG may depend on the state and on the actions of all players.

Each player is interested in maximizing some criterion on this sequence of rewards. Here, we will consider the problem of computing the minimax equilibrium in a two-player zero-sum SG where each player tries to maximize his own value, defined as the expected $\gamma$-discounted cumulative reward. The value attained at the minimax equilibrium is the optimal value. In the case of simultaneous games, the major difference between those games and MDPs is that each player may have to randomize his strategies to reach this equilibrium (Von Neumann, 1947).

2.1. Two-Player Zero-Sum Discounted Stochastic Games
A two-player $\gamma$-discounted SG can be considered as a tuple $(S, (A^1(s)), s \in S, (A^2(s)), s \in S, p, r, \gamma)$ where $S$ is the state space$^1$, $A^1(s)$ is the finite set of actions player 1 can play in state $s$, $A^2(s)$ is the finite set of actions player 2 can play in state $s$, $p(s'|s, a^1, a^2)$ with $a^1 \in A^1(s)$ and $a^2 \in A^2(s))$ is the probability transition from state $s$ to state $s'$, $r(s, a^1, a^2)$ with $a^1 \in A^1(s)$ and $a^2 \in A^2(s))$ is the reward of both players bounded by $R_{\text{max}}, \gamma$ is the discount factor. A strategy $\pi^i$, where $i \in \{1, 2\}$, maps a state $s \in S$ to a probability distribution $\pi^i(\cdot|s)$ over $A^i(s)$. Those strategies are named policies in the MDP literature.

We will adopt both vocabulary indifferently.

The stochastic kernel characterizing the transition of the game is $P_{\pi^1, \pi^2}(s'|s, a^1, a^2) = \sum_{s''} p(s''|s, a^1, a^2) \pi^1(s''|s, a^1) \pi^2(\cdot|s, a^2)$. The stochastic kernel is the probability to go from $s$ to $s'$ when player 1 is playing strategy $\pi^1$ and player 2 is playing strategy $\pi^2$. The reward function will be noted $r_{\pi^1, \pi^2} = \sum_{s''} p(s''|s, a^1, a^2) r(s, a^1, a^2)$. This quantity represents the reward each player can expect while player 1 plays $\pi^1$ and player 2 plays $\pi^2$.

One can see zero-sum two-player games as two players maximizing opposite rewards. Another way to see it is that the goal of player 1 is to maximize his cumulated $\gamma$-discounted reward while the goal of player 2 is to minimize it. From now on, we will note $\mu$ the policy of the maximizer and $\nu$ the policy of the minimizer.

Let $v_{\mu, \nu}(s) = \mathbb{E}[\sum_{t=0}^{+\infty} \gamma^t r_{\mu, \nu}(s_t)|s_0 = s, s_{t+1} \sim P_{\mu, \nu}(\cdot|s_t)]$ be the cumulative reward if the players 1 and 2 respectively use the stationary strategies $\mu$ and $\nu$. The value function $v_{\mu, \nu}$ maps the state $s$ to its value $v_{\mu, \nu}(s)$.

2.2. Bellman Operators
Let us define the following five operators on value $v$:

$$T_{\gamma, \mu} v = r_{\mu, \nu} + \gamma P_{\mu, \nu} v$$
$$T_{\mu} v = \min_{\nu} T_{\mu, \nu} v$$
$$T_{\nu} v = \max_{\mu} T_{\mu, \nu} v$$
$$T v = \max_{\mu} T_{\mu} v$$
$$\hat{T} v = \min_{\nu} \hat{T} v$$

Here $\mu$ and $\nu$ are random policies. The min and max are well defined because $\mu$ and $\nu$ are in compact sets since we are considering finite sets of actions. All those operators are contractions in $L_{\infty}$-norm with constant $\gamma$. We have the

$^1$We will only consider a finite state space. The case where the state space is continuous is beyond the scope of this paper.
following remarkable property (Patek, 1997):
\[ \forall v, \; T \vDash v = T \vDash v \] (1)
Equation (1) is a consequence of von Neumann’s Minimax theorem (Von Neumann, 1947; Patek, 1997).

2.3. Minimax Equilibrium

In the setting of zero-sum two-player SGs, notions of minimax equilibrium and of Nash equilibrium are equivalent. In this case, the optimal value of the game is:
\[ v^* = \min_{\nu} \max_{\mu} v_{\mu,\nu} = \min_{\nu} \max_{\mu} v_{\mu,\nu} \] (2)
The existence of \( v^* \) is proved in Patek (1997) for a more general class of games. This value can be achieved using mixed stationary strategies (Patek, 1997). Equation (2) is a consequence of Equation (1) and of the contraction property. The value \( v^* \) is the unique fixed point of the operator \( T \). We will note \( v_{\mu,\nu} = \inf_{\nu} v_{\mu,\nu} \) the fixed point of operator \( T_{\mu} \). An optimal counter strategy against \( \mu \) is any strategy \( \nu \) satisfying \( v_{\mu,\nu} = v_{\mu,\nu} \).

2.4. Policy Iteration, Value Iteration and Generalized
Policy Iteration

The three algorithms we intend to analyse are VI, PI and GPI. They all share the same greedy step. A strategy \( \mu \) is greedy with respect to some value \( v \) (noted \( \mu \in \mathcal{G}(v) \)) when \( T \vDash v = T_{\mu} \vDash v = \min_{\nu} T_{\mu,\nu} \vDash v \).

Remark 1. In practice, trying to find \( \mu \) greedy with respect to a value \( v \) means trying to maximize \( (T_{\mu} \vDash v)(s) \) for each state \( s \). Let us fix some state \( s \); we try to find
\[ \left( \min_{\mu} T_{\mu,\nu} \vDash v \right)(s) = \min_{\mu,\nu} \min_{E_{a\sim\mu}(s),a',v_{\mu,\nu}(s)} \left[ r(s, a, a') + \gamma v(\Sigma) \right] \]
For a constant state \( s \), this is equivalent to trying to find the minimax equilibrium of a matrix game. The matrix of the game with stochastic reward is defined by \( r(s, a, a') + \gamma v(\Sigma_{a,a'}) \). From the expectation of this matrix, the minimax equilibrium can be computed with linear programming.

In the case of turn-based games, finding a greedy policy is much simpler. Indeed, since at each state only one player controls the SG, finding a greedy strategy is reduced to finding a maximum over the actions of player 1.

The algorithms differ by the way they do the evaluation step. We describe them by showing how they work from iteration \( k \) to iteration \( k + 1 \). Value Iteration (VI) iterates as follows:
\[ v_{k+1} = T \vDash v_k \]
This can equivalently be written as follows:
\[ \mu_{k+1} = \mathcal{G}(v_k) \]
\[ v_{k+1} = \mu_{k+1} \vDash v_k \]

Policy Iteration (PI) iterates as follows
\[ \mu_{k+1} = \mathcal{G}(v_k) \]
\[ v_{k+1} = \mu_{k+1} \vDash v_k + (T_{\mu_{k+1}} \vDash v_k)^+ \]
Finally, Generalized Policy Iteration (GPI) iterates in a way that interpolates between VI and PI:
\[ \mu_{k+1} = \mathcal{G}(v_k) \]
\[ v_{k+1} = (T_{\mu_{k+1}} \vDash v_k)^m \]

It is clear that GPI generalizes VI and PI. In particular, the error propagation bounds for VI (when \( m = 1 \)) and PI (when \( m = \infty \)) will follow from the analysis we shall provide for GPI.

2.5. Example and Special Cases

In this paper, we consider the game Alesia (Meyer et al., 1997), that is a two-player zero-sum simultaneous game. Both players control a token in the center of a five box board. At the beginning each player has a finite budget \( n \). At each turn, every one has to bet at least 1 if his own remaining budget is not 0. The goal of each player is to put the token on his side of the board. At each turn the token moves toward the objective of the player who bets the most (in case of equality the token doesn’t move). The one winning is the one who can push his token outside of the board.

This game can be modelled as follows. The state space \( S \) is a triplet \( \{1, ... , 5\} \), budget of player 1 \( \in \{0, ..., n\} \), budget of player 2 \( \in \{0, ..., n\} \) and an absorbing state \( \Omega \) where the players fall at the end of the game. The action space in state \( (s, u, v) \) is \( \mathcal{A}^1((s, u, v)) \) for player 1. If \( u \neq 0 \), then \( \mathcal{A}^1((s, u, v)) = \{1, ..., u\} \), else \( \mathcal{A}^1((s, u, v)) = \{0\} \). For player 2, if \( v \neq 0 \), then \( \mathcal{A}^2((s, u, v)) = \{1, ..., v\} \) else \( \mathcal{A}^2((s, u, v)) = \{0\} \). If player 1 bets \( a_1 \) and player 2 bets \( a_2 \) in state \( (s, u, v) \), then the next state is \( (s', u - a_1, v - a_2) \) where \( s' = s - 1_{a_1 \leq a_2} + 1_{a_2 \leq a_1} \) or \( \Omega \) if \( s' \notin \{1, ..., 5\} \). The reward is 1 if the token leaves the board from position 5 and -1 if the token leaves the board from position 1.

Remark 2. In turn-based games, each state is controlled by a single player. In the zero-sum two-player SGs framework, they can be seen as a game where \( \forall s \in S, \text{card}(\mathcal{A}^1(s)) = 1 \) or \( \text{card}(\mathcal{A}^2(s)) = 1 \). In this special case, optimal strategies are deterministic (Hansen et al., 2013). Furthermore, and as we already mentioned, the greedy step (see definition in Sec. 2.4) reduces to finding
a maximum rather than a minimax equilibrium and is thus significantly simpler. Furthermore, one can see an MDP as a zero-sum two-player SG where one player has no influence on both the reward and the dynamics. Therefore, our analysis should be consistent with previous MDP analyses.

3. Stochastic Games and Approximate Generalized Policy Iteration (AGPI)

In this section, we analyse the algorithm in the case of approximations in the greedy and evaluation steps. This analysis was done in $L_{∞}$-norm in Patek (1997) for PI. We generalize it for AGPI in the case of $σ$-weighted $L_{p}$-norm, that is defined for a function $h$ and a distribution $σ$ on the state space as $\|h\|_{p,σ} = \left( \sum_{s ∈ S} |h(s)|^p σ(s) \right)^{1/p}$.

3.1. Approximate Generalized Policy Iteration

In Patek (1997) this algorithm is presented as GPI. It has been first presented in (Van Der Wal, 1978). Similarly to its exact counterpart, an iteration of this algorithm can be divided in two steps: a greedy step and an evaluation step. The main difference is that we account for possible errors in both steps (respectively $ε'_k$ and $ε_k$).

For the greedy step, we shall write $μ_k ← \hat{G}_k(v_{k−1})$ for:

$$\mathcal{T} v_{k−1} ≤ \mathcal{T}_{μ_k} v_{k−1} + ε'_k$$

or, $∀µ \mathcal{T}_µ v_{k−1} ≤ \mathcal{T}_{μ_k} v_{k−1} + ε'_k$ (3)

In other words, the strategy $μ_k$ is not necessarily the best strategy, but it has to be at most $ε'_k$ away from the best strategy.

For the evaluation step, we consider that we may have an additive error $ε_k$:

$$v_k = (\mathcal{T}_{μ_k})^m v_{k−1} + ε_k$$ (4)

**Remark 3.** Evaluation step: In the evaluation step the policy $μ_k$ is fixed and we apply $m$ times the operator $\mathcal{T}_{μ_k} = \min_\nu \mathcal{T}_{μ_k,ν} \cdots$. Since the policy of the maximizer is fixed, the problem solved in the evaluation step consists in finding an optimal $m$-horizon counter-policy for the minimizer.

3.2. Error Propagation

Since errors may accumulate from iterations to iterations, we are interested in bounding the difference

$$l_k = v_∗ - v_{μ_k} ≥ 0$$

where $v_∗$ is the minimax value of the game (obtained when both players play the Nash equilibrium $μ_*$ and $ν_*$) and $v_{μ_k}$ is the value when the maximizer plays $μ_k$ and the minimizer plays the optimal counter-strategy against $μ_k$. This is a natural measure of quality for the strategy $μ_k$ that would be output by the approximate algorithm.

By definition, we have:

$$∀ν, ν_+ ≤ \mathcal{T}_{μ_k,ν} v$$ (5)

In addition, we shall consider a few notations. The minimizer policies $ν^k_0, ν_k, ν_k$ and $\bar{ν}_k$ are policies that respectively satisfy:

$$(\mathcal{T}_{μ_k})^{i+1} v_{k-1} = \mathcal{T}_{μ_k,ν^k_i} \cdots \mathcal{T}_{μ_k,ν_k} v_{k-1},$$ (6)

$$\mathcal{T}_{μ_k} v_k = \mathcal{T}_{μ_k,ν^k_0} v_k,$$

$$\mathcal{T}_{μ_k} v_{μ_k} = \mathcal{T}_{μ_k,ν^k_0} v_{μ_k}.$$ (8)

In order to bound $l_k$, we will study the following quantities similar to those introduced by Scherrer et al. (2012) (recall that $T_µ$ and $T_{µ,ν}$ are defined in Sec. 2.2 and the stochastic kernel is defined in Sec. 2.1):

$$d_k = v_∗ - (\mathcal{T}_{μ_k})^m v_{k−1} = v_∗ - (v_k - ε_k),$$

$$s_k = (\mathcal{T}_{μ_k})^m v_{k−1} - v_{μ_k} = (v_k - ε_k) - v_{μ_k},$$

$$b_k = v_k - \mathcal{T}_{μ_k+1} v_k,$$

$$x_k = (\mathcal{I} - γP_{μ_k,ν_k}) ε_k + ε'_k,$$

$$y_k = γP_{μ_k,ν_k} ε_k + ε'_k.$$ (7)

Notice that $l_k = d_k + s_k$. We shall prove the following relations, similar to the one proved in (Scherrer et al., 2012).

**Lemma 1.** The following linear relations hold:

$$b_k ≤ γP_{μ_k,ν_k} γP_{μ_k,ν_k−1} \cdots γP_{μ_k,ν_k−b_{k−1} + x_k},$$

$$d_{k+1} ≤ γP_{μ_k,ν_k} d_k + y_k + \sum_{j=1}^{m−1} γP_{μ_k,ν_k−1} \cdots γP_{μ_k,ν_k−b_{k−1}} b_k,$$

$$s_{k+1} ≤ (γP_{μ_k,ν_k})^m (\sum_{i=1}^{∞} γP_{μ_k,ν_k−1} \cdots γP_{μ_k,ν_k−b_{k−1}} b_{k−1}).$$

Contrary to the analysis of Scherrer et al. (2012) for MDPs in which the operator $T_µ$ is affine, it is in our case non-linear. The proof that we now develop is thus slightly trickier.

**Proof.** Let us start with $b_k$:

$$b_k = v_k - \mathcal{T}_{μ_k+1} v_k = v_k - \mathcal{T}_{μ_k} v_k + \mathcal{T}_{μ_k} v_k - \mathcal{T}_{μ_k+1} v_k$$
In Equation (3) with $\mu = \mu_k - 1$ and $k \leftarrow k + 1$, we have $T_{\mu_k} v_k \leq T_{\mu_{k+1}} v_k + \epsilon_{k+1}$ then:

$$b_k \leq v_k - T_{\mu_k} v_k + \epsilon_{k+1} = v_k - \epsilon_k - \sum_{i=1}^{\mu_k} v_k + \gamma P_{\mu_k, \nu_k} \epsilon_k + \epsilon_{k+1}$$

From Equation (4), we have $v_k - \epsilon_k = (T_{\mu_k})^m v_{k-1}$. Thus,

$$b_k \leq (T_{\mu_k})^m v_{k-1} - T_{\mu_k, \nu_k} (T_{\mu_k})^m v_{k-1} + x_k \quad \text{(Eq. (6))}$$

Then $d_{k+1}$ becomes:

$$d_{k+1} \leq \gamma P_{\mu_k, \nu_k} (v_k - v_k) + \epsilon_{k+1}$$

To bound $d_{k+1}$, we decompose it in the three following terms:

$$d_{k+1} = T_{\mu_k} v_k - T_{\mu_k} v_k - T_{\mu_k} v_k$$

Finally, bounding term 3 involves the $b_k$ quantity:

$$T_{\mu_k+1} v_k = (T_{\mu_k+1})^m v_k$$

Let us finally bound $s_k$:

$$s_k = (T_{\mu_k})^m v_{k-1} - v_k$$

From linear recursive relations of the kind of Lemma 1, Scherrer et al. (2012) show how to deduce a bound on the
$L_p$-norm of $l_k$. This part of the proof being identical to that of Scherrer et al. (2012), we do not develop it here. For completeness however, we include it in Appendix A.1 of the Supplementary Material.

**Theorem 1.** Let $\rho$ and $\sigma$ be distributions over states. Let $p$, $q$ and $q'$ be such that $\frac{1}{q} + \frac{1}{\sigma} = 1$. Then, after $k$ iterations, we have:

$$
\|l_k\|_{p,\rho} \leq \frac{2(\gamma - \gamma^k)(C_q^{1,k,0})^{\frac{1}{2}}}{(1 - \gamma)^2} \sup_{1 \leq j \leq k-1} \|e_j\|_{pq',\sigma},
$$

$$
+ \frac{(1 - \gamma^k)(C_q^{0,k,0})^{\frac{1}{2}}}{(1 - \gamma)^2} \sup_{1 \leq j \leq k} \|e_j\|_{pq',\sigma},
$$

$$
+ \frac{2\gamma^k}{1 - \gamma} (C_q^{b,k+1,0})^{\frac{1}{2}} \min(\|d_0\|_{pq',\sigma}, \|b_0\|_{pq',\sigma}).
$$

where

$$
C_q^{d,k} = \frac{(1 - \gamma)^2}{\gamma - \gamma^k} \sum_{i=1}^{k-\infty} \sum_{j=i}^{k} \gamma^j c_q(j + d),
$$

with the following norm of a Radon-Nikodym derivative:

$$
c_q(j) = \sup_{\mu_1,\mu_2,\ldots,\mu_j} \left\| \frac{d(\rho \mu_1,\nu_1) \cdots \mu_j,\nu_j)}{d(\sigma \mu_1,\nu_1) \cdots \mu_j,\nu_j)} \right\|_{q,\sigma}
$$

**Remark 4.** The Radon-Nikodym derivative of measure $\rho$ on $S$ with respect to measure $\sigma$ on $S$ is in the discrete case the function $h$ defined by: $h(s) = \frac{\rho(s)}{\sigma(s)}$ when $\sigma(s) \neq 0$, $\infty$ otherwise.

**Remark 5.** If player 2 has no influence on the game, then the $\gamma$-discounted two-player zero-sum SG is simply a $\gamma$-discounted MDP. In that case the bound is the same as in Scherrer et al. (2012).

**Remark 6.** In the case of a SG with no discount factor but with an absorbing state the expression given in lemma 1 is still valid. Instead of having a $\gamma$-discounted transition kernel we would have a simple transition kernel. And if this transition kernel has a property

$$
\exists l, \sup_{\mu_0,\nu_0,\ldots,\mu_i,\nu_i} \left\| \prod_{i=0}^l \mu_{i+1,\nu_i} \right\|_{\infty} \leq \gamma < 1,
$$

we could still compute properties on the propagation of errors.

**Remark 7.** One should notice that when $p$ tends to infinity, the bound becomes:

$$
\lim_{k \to +\infty} \|l_k\|_{\infty} \leq \frac{2\gamma}{(1 - \gamma)^2} \epsilon + \frac{1}{(1 - \gamma)^2} \epsilon',
$$

where $\epsilon$ and $\epsilon'$ are respectively the sup of errors at the evaluation step and the sup of errors at the greedy step in $\infty$-norm. We thus recover the bounds computed by Patek (1997).

### 4. Application

The analysis of error propagation presented in Sect. 3.2 is general enough to develop several implementations. From the moment one can control the error made at each iteration step, the bound presented in Theorem 1 applies.

#### 4.1. Algorithm

In this section, we present the Approximate Generalized Policy Iteration-$Q$ (AGPI-$Q$) algorithm which is an extension for SG of Fitted-$Q$. This algorithm is offline and uses the so-called state-action value function $Q$. The state-action value function extends the value function by adding two degrees of freedom for the first action of each player. More formally, the state-action value function $Q^{\mu,\nu}(s, a, b)$ is defined as

$$
Q^{\mu,\nu}(s, a, b) = E[r(s, a, b)] + \sum_{s' \in S} p(s'|s, a, b) v_{\mu,\nu}(s').
$$

We assume we are given some samples $(x^1, \alpha_1^1, \alpha_2^1, r^1, x^{j'})_{j=1,\ldots,N}$ and an initial $Q$-function (here we chose the null function). As it is an instance of AGPI, each iteration of this algorithm is made of a greedy step and an estimation step. The algorithm is precisely described in Algorithm 1.

#### Algorithm 1 AGPI - $Q$ for Batch sample

**Input:** $(x^1, \alpha_1^1, \alpha_2^1, r^1, x^{j'})_{j=1,\ldots,N}$ some samples, $q_0 = 0$ a $Q$-function, $\mathcal{F}$ an hypothesis space

**for** $k=1,2,\ldots,K$ **do**

**Greedy step:**

**for all** $j$ **do**

$$
\bar{a}_j = \text{arg max}_{\bar{a}} \min_b q_{k-1}(x^j, \bar{a}, \bar{b}) \quad \text{(solving a matrix game)}
$$

**end for**

**Evaluation step:**

$$
q_{k,0} = q_{k-1}
$$

**for** $i=1,\ldots,m$ **do**

**for all** $j$ **do**

$$
q^i = r(x^j, \alpha_1^i, \alpha_2^i) + \gamma \min_b q_{k-1}(x^{j'}, \bar{a}_j, b)
$$

**end for**

$$
q_{k,i} = \text{arg min}_{q \in \mathcal{F}} \sum_{j=1}^N l(q(x^j, \alpha_1^j, \alpha_2^j, q^i))
$$

Where $l$ is a loss function. $q_k = q_{k,m}$

**end for**

**output** $q_k$

For the greedy step, the minimax policy for the maximizer on each matrix game defined by $(q_k(x^j, a, b))_{a,b}$. In general, this step involves solving $N$ linear programs; recall
that in the case of a turn-based game this step reduces to finding a maximum. The evaluation step involves solving the MDP with an horizon \( m \) for the minimizer. This part is similar to fitted-Q iteration. At each step we try to find the best fit over our hypothesis space for the next \( Q \)-function according to some loss function \( l(x,y) \) (often, \( l(x,y) = |x - y|^2 \)).

4.2. Analysis

In this case, we have \( \epsilon_k = 0 \) and \( \epsilon_k \) the error made on \( q_k \) at each iteration. Let us note \( \epsilon_{k,i} \) the error or fitting the \( Q \)-function on the feature space. The Bellman operator in the case of actions value function \( Q(s,a_1,a_2) \) for policy \( \mu \) and \( \nu \) is \( (T_{\mu,\nu} Q)(s,a_1,a_2) = r(s,a_1,a_2) + \gamma \mathbb{E}_{\nu}[Q(s',\mu(\cdot),\nu(\cdot))] \). The other non-linear operators are analogous to those defined in Sec. 2.2. In this section the operator used is the one on \( Q \)-function.

We have \( q_{k,i} \) such as \( T_{\mu,k} q_{k,0} = T_{\mu,k} q_{k,m-1} \ldots T_{\mu,k} q_{k,0} q_{k,0} \). Furthermore we have \( q_{k,i} \) and \( \epsilon_{k,i} \). On the one hand, we have:

\[
\epsilon_k = q_k - T_{\mu,k} q_{k,0} - T_{\mu,k} q_{k,m-1} \ldots T_{\mu,k} q_{k,0} \epsilon_{k,i} + \sum_{i=0}^{m-1} \mathbb{P}_{\mu,k} \epsilon_{k,i}
\]

On the other hand (with \( \tilde{\nu}_{k,i} \) as \( q_{k,i+1} = T_{\mu,k} \tilde{\nu}_{k,i} q_{k,i} + \epsilon_{k,i} \)), we have:

\[
\epsilon_k = q_k - T_{\mu,k} q_{k,0} - T_{\mu,k} q_{k,m-1} \ldots T_{\mu,k} q_{k,0} \epsilon_{k,i} + \sum_{i=0}^{m-1} \mathbb{P}_{\mu,k} \epsilon_{k,i}
\]

From these inequalities, we can provide the following bound (the proof is given in Appendix B):

\[
\|l_k\|_{\rho,\rho} \leq \frac{2(\gamma - \gamma^k)(1 - \gamma^m)}{(1 - \gamma)^3} (C_q^{1,k,0,m,0}) \sup_i \|\epsilon_{k,i}\|_{\rho,\rho} + \frac{2\gamma^k}{1 - \gamma} (C_q^{k,k+1,0}) \min_i \|d_0\|_{\rho,\rho} + \sum_{i=1}^{\infty} \sum_{j=i}^{\infty} \gamma^j c_q(j + d).
\]

4.3. Complexity analysis

At the greedy step, the algorithm solves \( N \) minimax equilibria for a zero-sum matrix game. This is usually done by linear programming. For state \( s \), the complexity of such an operation is the complexity of solving a linear program with \( c_s = 1 + \text{card}(A_1(s)) + \text{card}(A_2(s)) \) and \( c_s = 1 + \text{card}(A_1(s)) \) constrains and with \( \text{card}(A_1(s)) \) variables. Let us note \( L(c_s,\text{card}(A_1(s))) \) this complexity. Then, the complexity of this step is bounded by \( N L(c_s,\text{card}(A_1(s))) \).

Using the simplex method, \( L(c,a) \) may grow exponentially with \( c \) while with the interior point method, \( L(c,a) \) is \( O(a^5) \) (Karmarkar, 1984). The time to compute \( q_j \) in the evaluation step depends on finding a maximum over \( A^2(x^j) \). And the regression complexity to find \( q_{k,i} \) depends on the regression technique. Let us note this complexity \( R(N) \). Finally, the complexity of this step is \( mR(N) \).

The overall complexity is thus \( O(NL(c,a) + mR(N)) \); in general, the complexity of solving the linear program will be the limiting factor.

5. Experiments

In this section, AGPI-Q is tested on the Alesia game described in Sec. 2.5 where we assume that both players start with a budget \( n = 20 \). As a baseline, we use the exact solution of the problem provided by VI. We have run the algorithm for \( K = 10 \) iterations and for \( m \in \{1, 2, 3, 4, 5\} \) evaluation steps. We have considered sample set sizes \( N = 2500, 5000, 10000 \). Each experiment is repeated 20 times. First, we generate \( N \) uniform samples \( (x^j) \) over the state space. Then, for each state, we draw uniformly the actions of each player in the set of their own action space. Then, we set \( a_1^j \sim \mathcal{U}(A_1(x^j)) \), \( a_2^j \sim \mathcal{U}(A_2(x^j)) \), with \( r_j = r(x^j, a_1^j, a_2^j) \). As hypothesis space, we use CART trees (Breiman et al., 1984) which exemplifies the non-parametric property of the algorithm.

The performance of the algorithm is measured as the mean-squared error between the value function \( v_K(s) = \min_{a_1} \max_{a_2} q_K(s, a_1, a_2) \) where \( q_K \) is the output of the algorithm AGPI-Q and the actual value function computed via VI. Figure 1 shows the evolution of performance along iterations for \( N = 10000 \) for the different values of the parameter \( m \). Figure 2 shows the exact value function (Fig. 2(a)) and the approximated one \( v_K \) (Fig. 2(b)). The complete list of experiments results can be found in the supplementary.
For each size of sample set, the asymptotic convergence is better for small values of \( m \). This is conform to Eq. (11), in which the term \( \frac{2(\gamma - \gamma^k)(1-\gamma^m)}{(1-\gamma)^3} \) increases with \( m \). However, for small values of \( k \), the mean-squared error is reducing when \( m \) is increasing. This is coherent with experimental results when using MPI for MDP: the bigger \( m \), the higher the convergence rate. The price to pay for this acceleration of convergence towards the optimal value is an heavier evaluation step. This is similar to results in the exact case Puterman (1994). Overall, this suggests to use large values of \( m \) at the beginning of the algorithm and to reduce it as \( k \) grows to get a smaller asymptotic error.

It appears that the provided bound is highly sensitive to \( \gamma \) (which is a common problem of ADP). This is critical and further work should concentrate on reducing the impact of \( \gamma \) in the final error bound. Since non-stationary policies can reduce the impact of \( \gamma \) in MDP (Scherrer & Lesner, 2012), extensions of this work to zero-sum two-player SGs is forecasted. Moreover, we empirically demonstrate that AGPI-Q performs well on a simultaneous two-player game, namely Alesia.

6. Conclusion and Perspectives

This work provides a novel and unified error propagation analysis in \( L_p \)-norm of well-known algorithms (API, AVI and AGPI) for zero-sum two-player SGs. It extends the error propagation analyses of Scherrer et al. (2012) for MDPs to zero-sum two-player SGs and of Patek (1997) which is an \( L_\infty \)-norm analysis for only API. In addition, we provide a practical algorithm (AGPI-Q) which learns a good approximation of the Nash Equilibrium from batch data provided in the form of transitions sampled from actual games (the dynamics is not known). This algorithm is an extension of Fitted-Q for zero-sum two-player SGs and can thus be non-parametric. No features need to be provided or hand-crafted for each different application which is a significant advantage. Finally, we empirically demonstrate that AGPI-Q performs well on a simultaneous two-player game, namely Alesia.

References


A. Appendix: Demonstration of Lemma 2

The following proof is similar to the one for MDP (Scherrer et al., 2012) we write it for the sake of completeness. Furthermore, equation (14) is the starting point of demonstration of (11).

We will use the same abusive but simplifying notation as in (Scherrer et al., 2012). We will note $\Gamma^n$ any product of $n$ discounted transition Kernel. Then $\Gamma^n$ represents the set \{ $\gamma P_{\mu_i, \nu_i}, \ldots, \gamma P_{\mu_n, \nu_n}$, with $\mu_i, \nu_i$ random strategies}. Then $\Gamma^n$ is the represent a class of discounted stochastic matrix. One should read $\Gamma^n$ as there exist $\gamma P_{1}, \ldots, \gamma P_{n}$ such as $\Gamma^n$ represent the product $\gamma P_{1} \ldots \gamma P_{n}$. For example we have the following property:

$$\alpha_1 \Gamma^i, \alpha_2 \Gamma^j + \alpha_3 \Gamma^k = \alpha_1 \alpha_2 \Gamma^{i+j} + \alpha_3 \Gamma^k.$$ 

We can rewrite Lemma 1 in this Simple way:

\begin{align*}
  b_k &\leq \Gamma^m b_{k-1} + x_k, \\
  d_{k+1} &\leq \Gamma d_k + y_k + \sum_{j=1}^{m-1} \Gamma^j b_k, \\
  s_k &\leq \Gamma^m (\sum_{i=1}^{\infty} \Gamma^i b_{k-1}).
\end{align*}

We prove these three inequalities:

\begin{align*}
  b_k &\leq \sum_{i=1}^{k} \Gamma^{m(k-i)} x_k + \Gamma^{mk} b_0, \\
  d_k &\leq \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{i=1}^{k-1} \sum_{j=i}^{m(i-1)} \Gamma^j x_{k-i} + \sum_{i=k}^{m(k-1)} \Gamma^i b_0 + \Gamma^k d_0 \\
  s_k &\leq \sum_{i=1}^{k-1} \sum_{j=mi}^{\infty} \Gamma^{j+m(k-i)} x_i + (\sum_{j=mk}^{\infty} \Gamma^j) b_0
\end{align*}

First inequality. After expanding (12) we get:

\begin{align*}
  b_k &\leq \sum_{i=1}^{k} \Gamma^{m(k-i)} x_k + \Gamma^{mk} b_0
\end{align*}

(13)
Second inequality:

\[d_k \leq \Gamma^k d_0 + \sum_{j=0}^{k-1} \Gamma^{k-1-j} (y_j + \sum_{l=1}^{m-1} \Gamma^l b_j)\]

\[\leq \Gamma^k d_0 + \sum_{j=0}^{k-1} \Gamma^{k-1-j} [y_j + \sum_{l=1}^{m} \Gamma^l (\sum_{i=1}^{j} \Gamma^{m(j-i)} x_i + \Gamma^{m} b_0)]\]

\[= \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{j=0}^{k-1-m} \sum_{l=1}^{j} \Gamma^{k-1-j+l+m(j-i)} x_i + z_k\]

\[= \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{i=1}^{k-1} \sum_{j=1}^{m(k-i)-1} \Gamma^{j} x_i + z_k\]

\[= \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{i=1}^{k-1} \sum_{j=i}^{m-1} \Gamma^{j} x_{k-i} + z_k\]

With \(z_k\):

\[z_k = \sum_{j=0}^{k-1-m} \sum_{l=1}^{j} \Gamma^{k-1-j+l+m(j-i)} b_0 + \Gamma^k d_0\]

\[= \sum_{i=k}^{mk-1} \Gamma^i b_0 + \Gamma^k d_0\]

last inequality (we replace \(b_{k-1}\) with it’s expression (13)):

\[s_k \leq \sum_{i=1}^{k} \sum_{j=mi}^{\infty} \Gamma^{j+m(k-i)} x_i + \left( \sum_{j=mk}^{\infty} \Gamma^j b_0 \right)\]

And finally bounding \(l_k\)

\[l_k = d_k + s_k\]

\[\leq \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^{j} x_{k-i} + z_k + \sum_{i=k}^{mk-1} \Gamma^i b_0 + \Gamma^k d_0\]  \hspace{1cm} (14)

Depend on errors

Depend on initial condition

\[\eta_k = z_k + z_k' = \sum_{i=k}^{\infty} \Gamma^i b_0 + \Gamma^k d_0\]

Then we prove the following lemma which is similar to that for MDP given by (Scherrer et al., 2012):
Lemma 2. \( \forall k \geq 1 \)

\[
|l_k| \leq 2 \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon_{k-i}| + \sum_{i=0}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon'_{k-i}| + h(k),
\]

with \( h(k) = 2 \sum_{i=k}^{\infty} \Gamma^i |b_0| \) or \( 2 \sum_{i=k}^{\infty} \Gamma^i |d_0| \).

Relation between \( b_0 \) and \( d_0 \):

\[
b_0 = v_0 - T_{\mu_1} v_0
\]
\[
= v_0 - v_\pi + T_{\pi_0} v_\pi - T_{\pi_0} v_0 = \bar{T}_{\mu_1} v_0
\]
\[
\leq -d_0 + \gamma P_{\mu_1, \nu_0} d_0 + \epsilon'_1
\]
\[
\leq (I - \gamma P_{\mu_1, \nu_0})^{-1}(\epsilon'_1 - b_0)
\]

One should notice that in a discrete state space \( P_{\mu, \nu} \) is a stochastic matrix. Then \( I - \gamma P_{\mu, \nu_0} \) is invertible since \( \gamma P_{\mu_1, \nu_0} \) has a spectral radius \( < 1 \).

then:

\[
|\eta_k| \leq \sum_{i=k}^{\infty} \Gamma^i (I - \gamma P_{\mu_1, \nu_0}) |d_0| + |\epsilon'_1| + \Gamma^k |d_0|
\]
\[
\leq \sum_{i=k}^{\infty} \Gamma^i (I + \Gamma) |d_0| + |\epsilon'_1| + \Gamma^k |d_0|
\]
\[
\leq 2 \sum_{i=k}^{\infty} \Gamma^i |d_0| + \sum_{i=k}^{\infty} \Gamma^i |\epsilon'_1|
\]

and also:

\[
|\eta_k| \leq \sum_{i=k}^{\infty} \Gamma^i |b_0| + \Gamma^k (I - \gamma P_{\mu_1, \nu_0})^{-1}(|\epsilon'_1| + |b_0|)
\]
\[
\leq \sum_{i=k}^{\infty} \Gamma^i |b_0| + \Gamma^k \sum_{i=0}^{\infty} \Gamma (|\epsilon'_1| + |b_0|)
\]
\[
\leq 2 \sum_{i=k}^{\infty} \Gamma^i |b_0| + \sum_{i=k}^{\infty} \Gamma^i |\epsilon'_1|
\]

because \( l_k \) is positive we have then:

\[
|l_k| \leq \sum_{i=1}^{k} \Gamma^{i-1} |y_{k-i}| + \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |x_{k-i}| + |\eta_k|
\]
\[
\leq \sum_{i=1}^{k} \Gamma^{i-1} (|\epsilon_{k-i}| + |\epsilon'_{k+1-i}|) + \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j (|I + \Gamma| |\epsilon_{k-i}| + |\epsilon'_{k+1-i}|) + |\eta_k| \quad \text{note that } \epsilon_0 = 0
\]
\[
\leq \sum_{i=1}^{k-1} (\Gamma^i + \sum_{j=i}^{\infty} (\Gamma^j + \Gamma^{j+1})) |\epsilon_{k-i}| + \sum_{i=1}^{k-1} \Gamma^{i-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon'_{k+1-i}| + \Gamma^{k-1} |\epsilon'_1| + \sum_{i=k}^{\infty} \Gamma^i |\epsilon'_1| + h(k)
\]
\[
\leq 2 \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon_{k-i}| + \sum_{i=0}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon'_{k-i}| + h(k) \quad \text{with } h(k) = 2 \sum_{i=k}^{\infty} \Gamma^i |b_0| \text{ or } 2 \sum_{i=k}^{\infty} \Gamma^i |d_0|
A.1. Demonstration of Theorem 1

Let us recall Scherrer’s lemma (demonstration can be found in (Scherrer et al., 2012)).

**Lemma 3.** Let $\mathcal{I}$ and $(J_i)_{i \in \mathcal{I}}$ be a sets of positive integers, $\{I_1, \ldots, I_n\}$ a partition of $\mathcal{I}$. Let $f$ and $(g_i)_{i \in \mathcal{I}}$ be function such as:

$$|f| \leq \sum_{i \in \mathcal{I}} \sum_{j \in J_i} \Gamma^j |g_i| = \sum_{l=1}^n \sum_{i \in I_l} \sum_{j \in J_i} \Gamma^j |g_i|.$$  

Then for all $p$, $q$ and $q'$ such as $\frac{1}{q} + \frac{1}{q'} = 1$ and for all distribution $\rho$ and $\sigma$ we have

$$\|f\|_{p,\rho} \leq \sum_{l=1}^n (C_q(l))^\frac{1}{p} \sup_{i \in I_l} \|g_i\|_{pq',\sigma} \sum_{i \in I_l} \sum_{j \in J_i} \gamma^j.$$  

with the concentrability coefficient written:

$$C_q(l) = \frac{\sum_{i \in I_l} \sum_{j \in J_i} \gamma^j c_q(j)}{\sum_{i \in I_l} \sum_{j \in J_i} \gamma^j}.$$  

Theorem 1 can be proven by applying lemma 3 with:

$$\mathcal{I} = \{1, ..., 2k\}$$
$$\mathcal{I} = \{I_1, I_2, I_3\}, I_1 = \{1, 2, ..., k - 1\}, I_2 = \{k, ..., 2k - 1\}, I_3 = \{2k\}$$

$\forall i \in I_1$
$g_i = 2\epsilon_{k-i}$
$J_i = \{i, i+1, \ldots\}$

$\forall i \in I_2$
$g_i = \epsilon'_{k-(i-k)}$
$J_i = \{i-k, i-k+1, \ldots\}$

$\forall i \in I_3$
$g_i = 2d_0$ or $2b_0$
$J_i = \{k, k+1, \ldots\}$
B. Appendix: Bound for AGPI-Q

From (14) and with \( x_k = (I - \Gamma)\epsilon_k \) and \( y_k = -\Gamma\epsilon_k \) we can compute:

\[
I_k = d_k + s_k \\
\leq \sum_{i=1}^{k} \Gamma^{i-1} \Gamma (-\epsilon_{k-i}) + \sum_{i=1}^{k-1} \sum_{j=1}^{\infty} \Gamma^{j} (I - \Gamma) \epsilon_{k-i} + z_k + z_k' \\
\leq \sum_{i=1}^{k} \Gamma^{i-1} \Gamma \left[ \sum_{l=0}^{m-1} \Gamma^{m-l-1} (-\epsilon_{k-i,l}) \right] + \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^{j} (I - \Gamma) \left[ \sum_{l=0}^{m-1} \Gamma^{m-l-1} \epsilon_{k-i,l} \right] + z_k + z_k' 
\]

In (15) we use (10) to bound \(-\epsilon_{k-i}\) and (9) to bound \(\epsilon_{k-i}\). Since \(l_k \geq 0\):

\[
|l_k| \leq \sum_{i=1}^{k} \Gamma^{i} \left[ \sum_{l=0}^{m-1} \Gamma^{m-l-1} |\epsilon_{k-i,l}| \right] + \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^{j} (I + \Gamma) \left[ \sum_{l=0}^{m-1} \Gamma^{m-l-1} |\epsilon_{k-i,l}| \right] + h(k) \quad \text{with } \epsilon_{0,l} = 0 \\
\leq 2 \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^{j} \left[ \sum_{l=0}^{m-1} \Gamma^{l} |\epsilon_{k-i,m-l-1}| \right] + h(k) \\
\leq 2 \sum_{i=1}^{k-1} \sum_{l=0}^{m-1} \sum_{j=i+l}^{\infty} \Gamma^{j} |\epsilon_{k-i,m-l-1}| + h(k)
\]

with lemma 3 applied to:

\[
\mathcal{I} = \{1, \ldots, (k-1)m + 1\} \\
\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2\}, \mathcal{I}_1 = \{1, 2, \ldots, (k-1)m\}, \mathcal{I}_2 = \{(k-1)m + 1\} \\
\forall \xi \in \mathcal{I}_1, \xi = i + (k-1)l \\
g_\xi = 2\epsilon_{k-i,m-l-1} \\
\mathcal{J}_\xi = \{i + l, i + l + 1, \ldots\} \\
\forall i \in \mathcal{I}_2, \quad g_i = 2d_0 \text{ or } 2b_0 \\
\mathcal{J}_i = \{k, k+1, \ldots\}
\]

The result is:

\[
\| l_k \|_{p,\rho} \leq \frac{2(\gamma - \gamma^k)(1 - \gamma^m)}{(1 - \gamma^3)} (C_{q,k,0,m,0}^{l})^{\frac{1}{p}} \sup_{i,l} \|\epsilon_{i,l}\|_{pq',\sigma} + \frac{2\gamma^k}{1 - \gamma} (C_{q}^{k, k+1, 0})^{\frac{1}{p}} \min(\|d_0\|_{pq',\sigma}, \|b_0\|_{pq',\sigma})
\]

With:

\[
C_{q}^{l,k',k,d} = \frac{(1 - \gamma^3)}{(\gamma^l - \gamma^k)(\gamma'^l - \gamma'^k)} \sum_{i=1}^{k-1} \sum_{i'=0}^{k'} \sum_{j=i'+l}^{\infty} \gamma^j c_q(j + d).
\]
C. Appendix: Experiments

Results for experiments for $N = 2500$ (Figure 3(a)) and $N = 5000$ (Figure 3(b))

C.1. Mean square error between approximate value and exact value

(a) Mean square error (y-axis) between the estimated value function and the true value function at step $k$ (x-axis). For $n = 20$ and $N = 2500$

(b) Mean square error (y-axis) between the estimated value function and the true value function at step $k$ (x-axis). For $n = 20$ and $N = 5000$

C.2. Exact value function, approximate value function and error for $N = 10000$

Figure 3. Exact value for the game of ALESIA. From left to right the value of the game (for player’s budget = 0…20) for token’s position 5 to 1

Figure 4. Value computed by AGPI-Q for the game of ALESIA. From left to right the approximate value of the game (for player’s budget = 0…20) for token’s position 5 to 1
Figure 5. Absolute difference between value computed by AGPIL-$Q$ and the exact value. From left to right the error of the game (for player’s budget = 0...20) for token’s position 5 to 1