Sketched Newton Value Iteration for Large-Scale Markov Decision Processes

Jinsong Liu[1], **Chenghan Xie**[2], **Qi Deng**[1], **Dongdong Ge**[1], **Yinyu Ye**[3]
[1]Shanghai University of Finance and Economics
[2]Fudan University
[3]Stanford University
liujinsong@163.sufe.edu.cn, 20307130043@fudan.edu.cn, qideng@sufe.edu.cn,
ge.dongdong@mail.shufe.edu.cn, yyye@stanford.edu

Abstract

Value Iteration (VI) is one of the most classic algorithms for solving Markov Decision Processes (MDPs), which lays the foundations for various more advanced reinforcement learning algorithms, such as Q-learning. VI may take a large number of iterations to converge as it is a first-order method. In this paper, we introduce the **N**ewton **V**alue **I**teration (NVI) algorithm, which eliminates the impact of action space dimension compared to some previous second-order methods. Consequently, NVI can efficiently handle MDPs with large action spaces. Building upon NVI, we propose a novel approach called **S**ketched **N**ewton **V**alue **I**teration (SNVI) to tackle MDPs with both large state and action spaces. SNVI not only inherits the stability and fast convergence advantages of second-order algorithms, but also significantly reduces computational complexity, making it highly scalable. Extensive experiments demonstrate the superiority of our algorithms over traditional VI and previously proposed second-order VI algorithms.

Introduction

The Markov Decision Process (MDP) (Puterman 2014) is a classical approach used to model sequential decision problems and finds wide application in various fields such as finance (Hambly, Xu, and Yang 2023), automatic driving (Kiran et al. 2021), and more. In scenarios where the the parameters of MDP model is unknown, Reinforcement Learning (RL) algorithms sample trajectories to estimate the optimal value function and policy. Many RL algorithms either directly originate from or can be perceived as stochastic approximation variants of the Bellman equation (Bellman 1966). For instance, the Q-learning (Watkins and Dayan 1992) algorithm can be viewed as a stochastic fixed-point iteration for solving the Q-Bellman equation (Kamanchi, Diddigi, and Bhatnagar 2021).

This motivates us to delve into algorithms for solving MDPs with full model information, as they can offer valuable insights and theoretical support for the design of new RL algorithms. The Bellman equation, being interpretable as a fixed-point iteration, has prompted the application of several acceleration techniques to solve MDPs, including Anderson acceleration (Zhang, O'Donoghue, and Boyd 2020), Nesterov's acceleration and Polyak's momentum (Nesterov 2003). Furthermore, viewing traditional MDP algorithms, such as value iteration and policy iteration, from the optimization perspective, has been a source of inspiration, leading to significant improvements (Goyal and Grand-Clement 2023). An intriguing viewpoint is to consider value iteration and policy iteration as first-order and second-order optimization algorithms, respectively (Bertsekas 2012; KALABA 1957; Pollatschek and Avi-Itzhak 1969; Puterman and Brumelle 1979). Among the classic second-order algorithms, Newton's method stands out as it is invariant to rescaling and coordinate transformations, which means Newton's method needs little or no tuning of hyperparameters. However, for large-scale problems, Newton's method becomes computationally expensive. A recent work by Kamanchi, Diddigi, and Bhatnagar (2021) applies the Newton-Raphson method to the Q-Bellman operator and proposes the G-SOVI algorithm. Nevertheless, due to its computational cost of $\mathcal{O}(|S|^3|A|^3)$, G-SOVI becomes impractical for large MDPs. Consequently, our objective is to investigate the Newton value iteration algorithm, incorporating dimension reduction techniques to strike a balance between computational cost and solution quality. By doing so, we aim to develop a more efficient approach for solving MDPs on a larger scale.

Initially, we introduce the Newton Value Iteration (NVI) algorithm, wherein we apply the Newton-Raphson method to the smooth V-Bellman operator instead of the Q-Bellman operator (Kamanchi, Diddigi, and Bhatnagar 2021). This modification reduces the computational cost significantly, from $\mathcal{O}(|S|^3|A|^3)$ to $\mathcal{O}(|S|^3)$. Additionally, we demonstrate that the approximate error of the smooth Bellman operator approaches zero as the soft-max parameter $\beta \to \infty$. Despite these advancements, the computational cost of NVI remains prohibitive for MDPs with large state spaces. To address this, we aim to leverage dimension reduction techniques from second-order optimization methods (Gower et al. 2019; Zhang et al. 2022; Hanzely et al. 2020). However, the Jacobian matrix of the smooth Bellman operator poses challenges due to its asymmetry, making it difficult to analyze the convergence of general second-order dimension reduction algorithms like those proposed in (Gower et al. 2019). Hence, we explore the application of the recently proposed Sketched Newton-Raphson method (Yuan, Lazaric,

and Gower 2022) to NVI, resulting in the novel dimension-reduced second-order value iteration algorithm—Sketched Newton Value Iteration. In this work, we not only provide convergence guarantees for SNVI but also conduct experiments to illustrate the superiority of SNVI over value iteration and G-SOVI. Our findings highlight the practical benefits of SNVI, making it a promising solution for MDPs with both large state and action spaces.

Our contributions can be summarized as follows:

- We propose the Newton Value Iteration algorithm, effectively reducing the computational complexity of G-SOVI from $\mathcal{O}(|S|^3|A|^3)$ to $\mathcal{O}(|S|^3)$. Additionally, we provide rigorous proofs for the global quadratic convergence of NVI.
- Building upon NVI, we introduce the Sketched Newton Value Iteration (SNVI) algorithm, which goes a step further in reducing computational complexity from $\mathcal{O}(|S|^3)$ to $\mathcal{O}(k^3)$, where $k \ll |S|$ is the sketch size. We establish that SNVI exhibits a sublinear convergence rate, given certain mild assumptions.
- To validate the effectiveness of our proposed algorithms, we conduct extensive numerical experiments on various MDP instances. The experimental results demonstrate that SNVI achieves faster convergence compared to traditional Value Iteration (VI) and the G-SOVI method.

In conclusion, our contributions entail the development of efficient algorithms for solving MDPs, significantly reducing computational complexity and demonstrating fast convergence rates through both theoretical analysis and practical experiments. These findings have the potential to advance the field of MDP-solving algorithms and can find applications in various real-world scenarios.

Related Work

This paper is related to the connections between MDP solving algorithms (Tamar et al. 2016; Puterman 2014; Ye 2011) and optimization theory (Boyd and Vandenberghe 2004), especially second order optimization methods (Nesterov and Polyak 2006; Conn, Gould, and Toint 2000; Yuan 2015) and corresponding dimension reduction techniques (Gower et al. 2019; D'ambrosio et al. 2020; Zhang et al. 2022; Hanzely et al. 2020). It seems more natural to combine MDP solving algorithms with first-order optimization algorithms (Nesterov 2003; Ghadimi, Feyzmahdavian, and Johansson 2015), as value iteration method can be seen as gradient descent of an unknown function (Bertsekas 2012), upon which Goyal and Grand-Clement (2023) define first-order methods for MDPs, extend Nesterov's acceleration and Polyak's momentum to MDPs and present novel lower bounds on the performances of value iteration algorithms. On the other hand, policy iteration can be seen as a second-order method (KALABA 1957; Pollatschek and Avi-Itzhak 1969; Puterman and Brumelle 1979). We refer to (Grand-Clément 2021) for a comprehensive review about the connection between MDP and optimization algorithms.

For problems with small or medium size, second-order methods are typically favored over first-order methods due to their faster convergence rates (Nesterov 2003; Boyd and Vandenberghe 2004). Among second-order methods, Newton-type methods stand out as the most popular choice. This preference stems from their invariance to rescaling and coordinate transformations, effectively minimizing the need for hyperparameter tuning (Gower et al. 2019). Kamanchi, Diddigi, and Bhatnagar (2021) apply Newton-Raphson method to the smooth Q-bellman operator and prove that their algorithm has quadratic convergence rate. However, the computational cost of Newton type methods is unacceptable for large problems, for example, the algorithm in (Kamanchi, Diddigi, and Bhatnagar 2021) has a $\mathcal{O}(|S|^3|A|^3)$ computational complexity, which is only practical for MDPs with both small action and state spaces. Inspired by the recent success of dimension reduced second-order algorithms (Gower et al. 2019; Zhang et al. 2022; D'ambrosio et al. 2020; Yuan, Lazaric, and Gower 2022), we believe that the combination of dimension reduction techniques and smooth Bellman operator has great potential for developing efficient second-order algorithms that can be applied to MDPs with both large state and action spaces.

Preliminaries

Consider the Markov decision process $\mathfrak{M} = \{\mathcal{S}, \mathcal{A}, P, r, \gamma\}$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P_{sa} \in \mathbb{R}^{|S|}$ is the probability of transition from state s to the next state when choosing action a. $r(s, a, s')$ is the reward obtained in state s when action a is chosen and the next state is s', and $\gamma \in [0, 1)$ is the discount factor. Let $|S|$ be the size of state space, $|A|$ be the size of action space. The objective of an MDP is to find an optimal policy $\pi^* \in \mathbb{R}_+^{|S|\times|A|}$, where a policy π maps each state to a probability distribution over the set of actions. The value function associated with the policy π is

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, \pi\left(s_t\right), s_{t+1}\right) \mid s_0\right],$$

where the expectation is taken over the policy π and the transition kernel P. Let v^* denote the optimal value function, i.e. the value function following the optimal policy π^*. The optimal value function can be obtained by solving the bellman operator $T : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$, where T is defined as:

$$T(v)_s := \max_{a\in\mathcal{A}}\{r_{sa} + \gamma P_{sa}^\top v\}, \quad \forall s \in \mathcal{S}, \tag{1}$$

and $r_{sa} = \sum_{s'\in\mathcal{S}} r(s, a, s')$. It is easy to show that T is a contraction map in sup-norm with factor γ:

$$\|T(v) - T(w)\|_\infty \le \lambda \cdot \|v - w\|_\infty.$$

The value iteration (VI) algorithm generates a sequence of value function:

$$v_0 \in \mathbb{R}^{|S|}, v_{t+1} = T\left(v_t\right), \forall t \ge 0,$$

and the optimal value function v^* is the unique fixed point of the bellman operator T:

$$v^* = Tv^*.$$

Denote the Q-value function as:

$$Q(s, a) = r_{sa} + \gamma P_{sa}^\top v, \forall (s, a) \in (|S| \times |A|). \tag{2}$$

It is easy to see that:

$$v(s) = \max_a Q(s,a).$$

Then we obtain the Q-bellman equation:

$$Q^*(s,a) = r_{sa} + \gamma P_{sa}^\top \max_a Q^*(s,a). \tag{3}$$

Smoothed Maximum Operation. Considering that $f(x) = \max_{i\in\{1,2,\dots,d\}} x_i, x \in \mathbb{R}^d$ is inherently non-smooth, an intuitive way (Kamanchi, Diddigi, and Bhatnagar 2021) to smoothly approximate the $\max$ operator is by using a function $f_\beta(x) = \frac{1}{\beta}\log\sum_{i=1}^d e^{\beta x_i}$. This approach yields the following observations:

$$|f(x) - f_\beta(x)| = \left|\frac{1}{\beta}\log\left(\sum_{i=1}^d e^{\beta(x_i - f(x))}\right)\right| \le \left|\frac{\log d}{\beta}\right|,$$

which tends to 0 as $\beta \to \infty$.

Notation. When dealing with a square matrix $A \in \mathbb{R}^{n\times n}$, the matrix norm is defined as $\|A\| = \sqrt{\lambda_M}$, where λ_M denotes the largest eigenvalue of the matrix product $A^\top A$. When it comes to a vector $v \in \mathbb{R}^n$, $\|v\|$ denotes the standard Euclidean norm. Additionally, we employ $\|v\|_Q = \sqrt{v^T Q v}$, where Q represents a positive-definite matrix.

Newton Value Iteration

In this section, we present the Newton Value Iteration (NVI) algorithm. As a starting point, it is worth recalling that the conventional value iteration algorithm can be viewed as a fixed-point iteration involving the Bellman operator. Building upon this concept, Reetz (1973) introduced the concept of successive relaxation Bellman equation:

$$T^w(v) := wT(v) + (1-w)v, \tag{4}$$

where w is the relaxation parameter. It is easy to show that the fixed points of T^w and T coincide, and the contraction factor of T^w is $1-\gamma+\gamma w$, which is smaller than that of T (Kamanchi, Diddigi, and Bhatnagar 2021). Using the successive relaxation Bellman operator (4), the modified value iteration can be seen as the fixed point problem:

$$(I - T^w)(v) = 0. \tag{5}$$

It is natural to consider applying Newton-Raphson method to solve (5). However, an inherent non-smoothness is associated with T^w due to the presence of the $\max$ operation within the Bellman operator. To address this challenge, a viable strategy is to leverage the log-sum-exp operator, which offers a smooth approximation of $\max$ and consequently yields a differentiable surrogate operator.

A Smooth Bellman Operator

$$T_\beta(v)_s = \frac{1}{\beta}\log\left(\sum_{a\in\mathbb{A}} e^{\beta\left(r_{sa}+\gamma P_{sa}^\top v\right)}\right),\ \forall s \in \mathcal{S}. \tag{6}$$

To surmount the non-smoothness challenge, we introduce the concept of the smooth Bellman Operator (6).

Algorithm 1: Calculation of the Jacobian Matrix

Input: the MDP model $\mathfrak{M}$,
v: vector of value functions,
β: Softmax approximation factor,
Output: $T_\beta(v)$

1: Calculate the Q-value function $Q \in \mathbb{R}^{|S|\times|A|}$ using (2).
2: Calculate $E = e^{\beta Q}$, applying the exponential operation element-wise.
3: Calculate the denominator $N = \sum_a E$.
4: Calculate the numerator $D = \gamma\sum_a(P \times E)$, with the cross product performed on the corresponding $|S|\times|A|$ dimensions.
5: Calculate $T_\beta(v) = D/N$, performing the division row-wise.
6: **return** $T_\beta(v)$

By calculating the Jacobian matrix of $T_\beta(v)$, we obtain:

$$\nabla T_\beta(v)_{(s,s')} = \frac{\gamma\sum_{a\in\mathbb{A}} P(s'|s,a)e^{\beta\left(r_{sa}+\gamma P_{sa}^\top v\right)}}{\sum_{a\in\mathbb{A}} e^{\beta(r_{sa}+\gamma P_{sa}^\top v)}}. \tag{7}$$

Employing this smoothed approach, we construct the successive relaxation smooth Bellman operator as follows:

$$\begin{aligned} T_\beta^w(v) &= wT_\beta(v) + (1-w)v \\ &= \frac{w}{\beta}\log\left(\sum_{a\in\mathbb{A}} e^{\beta\left(r_{sa}+\gamma P_{sa}^\top v\right)}\right) + (1-w)v, \forall s \in \mathcal{S}. \end{aligned}$$

The corresponding Jacobian matrix of the successive relaxation smooth Bellman operator becomes:

$$\nabla T_\beta^w(v) = w\nabla T_\beta(v) + (1-w)I. \tag{8}$$

It's noteworthy that the efficient computation of the Jacobian matrix (7) can be accomplished through vectorized operations, facilitated by modern matrix computation libraries like Numpy (Harris et al. 2020). We outline our specific approach for the computation of (7) as follows. The computational complexity of Algorithm 1 is evident as $\mathcal{O}(|S|^2|A|)$, which aligns with the computational complexity of value iteration.

Remark 1. We employ several tricks to guarantee numerical stability in Algorithm 1. For example, we subtracted a baseline from the Q-value function to prevent overflow.

The NVI Algorithm

We apply Newton-Raphson method to the successive relaxation smooth Bellman operator $T_\beta^w(v)$, leading to the iteration:

$$\begin{aligned} v_{t+1} &= v_t - (I - \nabla T_\beta^w(v_t))^{-1}(v_t - T_\beta^w(v_t)) \\ &= v_t - \frac{1}{w}(I - \nabla T_\beta(v_t))^{-1}w(v_t - T_\beta(v_t)) \\ &= v_t - (I - \nabla T_\beta(v_t))^{-1}(v_t - T_\beta(v_t)) \end{aligned} \tag{9}$$

Indeed, the variable w does not exert any influence on the NVI algorithm, as it is present on both sides of the linear system in the Newton-Raphson method (9). This distinguishes

it from G-SOVI. To maintain numerical stability, direct inversion is circumvented in favor of solving linear equations. Consequently, it becomes apparent that the computational complexity per step for (9) is $\mathcal{O}(|S|^3)$.

Convergence Analysis

Let us begin by demonstrating that the successive relaxation smooth Bellman operator is a contraction operator in the max-norm, rendering it appealing for fixed-point problems (Agarwal, Meehan, and O'regan 2001).

Lemma 1. *$T_\beta^w(v) : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ is a max-norm contraction operator:*

$$\|T_\beta^w(v_1) - T_\beta^w(v_2)\|_\infty \leq (w\gamma + |1-w|)\,\|v_1 - v_2\|_\infty$$

Proof Sketch: We begin by considering the discrepancy between the operators applied to two distinct vectors. Then we formulate the soft-max term as a well-defined probability distribution. Leveraging the property that the max-norm of a vector surpasses its expectation, we then derive the sought-after result.

We now introduce the theorem presented in (Ortega and Rheinboldt 2000) to establish the global convergence of NVI.

Theorem 1 (Global Convergence of Newton-Raphson Method)**.** *Suppose that $F : \mathbb{R}^d \to \mathbb{R}^d$ is continuous, component-wise concave on $\mathbb{R}^d$, differentiable and that $\nabla F(x)$ is non-singular and $\nabla F(x)^{-1} \geq 0$, i.e. each entry of $\nabla F(x)^{-1}$ is non-negative, for all $x \in \mathbb{R}^d$. Assume, further, that $F(x) = 0$ has a unique solution x^* and that ∇F is continuous on $\mathbb{R}^d$. Then for any $x_0 \in \mathbb{R}^d$ the Newton iterates converge to x^*.*

It suffices to apply Theorem 1 with the choice of $F = I - T_\beta^w$ to get the global convergence of NVI.

Theorem 2. *Let v^* be the fixed point of operator T_β^w. NVI converges to v^* for any choice of initial point v_0*

Applying Theorem 1 might appear less straightforward. The primary challenge lies in demonstrating the invertibility of $I - T_\beta^w$ and the non-negativity of each element in $(I - T_\beta^w)^{-1}$. This can be achieved by establishing a clear connection with the probability distribution matrix. The uniqueness of the fixed point can be derived through the contraction property of $I - T_\beta^w$.

Theorem 3. *NVI exhibits a quadratic rate of convergence:*

$$\|v_{t+1} - v^*\| \leq K\|v_t - v^*\|^2$$

where $K = \frac{3}{2}\beta$ and $\beta = L\|\nabla F(v)^{-1}\| \leq \frac{L}{w(1-\gamma)}$, L is Lipschitz constant.

The proof of Theorem 3 resembles the standard analysis for Newton's method.

Up to this point, we have demonstrated that NVI possesses a unique solution. Moreover, the global quadratic convergence rate, established through our analysis, furnishes NVI with a robust theoretical underpinning. This convergence rate serves as a definitive assurance for the efficacy of NVI in practice.

Algorithm 2: Sketched Newton Value Iteration

Input: the MDP model $\mathfrak{M}$
v_0: initial value function vector
β : Softmax approximation factor
n: the number of iterations
$\mathcal{D}$: distribution of sketching matrix
Output: v_t

```
Let t = 0.
while t ≤ n do
    Calculate the Jacobian according to Algorithm 1.
    Sample a fresh sketching matrix S_t ~ D_{v_t}
    Apply Sketched Newton step according to 10.
    t = t + 1
    if Stopping criteria is satisfied then
        Break
    end if
end while
return v_t
```

Sketched Newton Value Iteration

In contrast to G-SOVI, NVI effectively mitigates the impact of action space dimension on the matrix inverse for Newton-Raphson method. However, the challenge of an extensive state space frequently arises, resulting in an impractical computation cost for matrix inversion. Thus, analogous to second-order optimization algorithms, our objective is to curtail the matrix inversion dimension within the Newton system. In this section, we introduce the Sketched Newton Value Iteration algorithm to strike a balance between computational expenses and solution quality.

The SNVI Algorithm

Given the asymmetry of the Jacobian matrix (8), a direct application of dimension-reduced Newton methods, such as those proposed by Gower et al. (2019), for analyzing convergence properties becomes challenging. Therefore, we explore the integration of the sketched Newton-Raphson method introduced by Yuan, Lazaric, and Gower (2022).

In this approach, utilizing a random sketching matrix $S \in \mathbb{R}^{|S|\times k}$ with $k \ll |S|$, we employ the sketched Newton step at each iteration:

$$\begin{aligned} &v_{t+1} = v_t - \alpha d_t, \\ &d_t = \nabla F(v_t) S_t \big(S_t^\top \nabla F(v_t)^\top \nabla F(v_t) S_t\big)^\dagger S_t^\top F(v_t), \end{aligned} \tag{10}$$

This step notably reduces the computational complexity associated with solving linear systems from $\mathcal{O}(|S|^3)$ to $\mathcal{O}(k^3)$. As mentioned earlier, explicit computation of the Jacobian matrix incurs a computational cost of $\mathcal{O}(|S|^2|A|)$. However, within the framework of SNVI, there may not be a compelling need to compute the Jacobian matrix explicitly. Instead, we can obtain $\nabla F(v_t)S_t$ by performing k Jacobian-vector product operations, which can be efficiently accomplished by using automatic differentiation techniques (Paszke et al., 2017) or finite difference approximation (Jorge and Stephen 2006).

Convergence Analysis

Our insight into interpreting and analyzing the SNVI in is through its connection to the SGD. For the ease of notation, we define

$$H_S(v) \stackrel{\text{def}}{=} S\left(S^\top \nabla F(v)^\top \nabla F(v) S\right)^\dagger S^\top,$$
$$f_{S,y}(x) \stackrel{\text{def}}{=} \frac{1}{2}\|F(x)\|^2_{H_S(y)},$$
$$f_y(x) \stackrel{\text{def}}{=} \mathbb{E}\left[f_{S,y}(x)\right] = \frac{1}{2}\|F(x)\|^2_{\mathbb{E}[H_S(y)]},$$
$$f_t(v) \stackrel{\text{def}}{=} f_{v_t}(v).$$

Under the above notations, (10) is equivalent to:

$$v_{t+1} = v_t - \alpha \nabla f_{S_t, v_t}(v_t). \tag{11}$$

The objective function $f_{S,y}(x)$ boasts several favorable properties that render it highly amenable to optimization. Notably, it satisfies the interpolation condition and exhibits a beneficial smoothness characteristic. Specifically, for any $x^* \in \mathbb{R}^p$ such that $F(x^*) = 0$, we observe that the stochastic gradient is zero, denoted as $\nabla f_{S,y}(x^*) = 0$. This condition, known as the interpolation condition, is pivotal.

It is noteworthy, however, that the approach depicted in (11) deviates from classical SGD methods. In essence, the transition from the t-th iteration to the $(t+1)$-th iteration entails a switch in the objective function from $f_{x^t}(x)$ to $f_{x^{t+1}}(x)$, alongside a corresponding alteration in the distribution from $\mathcal{D}_{x^t}$ to $\mathcal{D}_{x^{t+1}}$. This nuanced aspect signifies that the method operates as an online SGD variant. Rest assured, we exercise meticulous care in addressing this aspect within our forthcoming convergence proofs. Initially, we introduce a specific type of smoothness property that plays a crucial role in our proofs.

Lemma 2. *For every $v \in \mathbb{R}^{|S|}$ and any realization $S \sim \mathcal{D}_v$ associated with any distribution $\mathcal{D}_v$,*

$$\frac{1}{2}\left\|\nabla f_{S,v}(v)\right\|^2 = f_{S,v}(v). \tag{12}$$

Since $\nabla f_{S,v}(v^*) = 0$ and $f_v(v^*) = 0$, (12) implies that

$$\left\|\nabla f_{S,v}(v) - \nabla f_{S,v}(v^*)\right\|^2 \leq 2\left(f_{S,v}(v) - f_{S,v}(v^*)\right),$$

which typically emerges as a result of assuming that $f_{S,v}(v)$ is convex and 1-smooth, as outlined in (Nesterov 2003). In our specific context, this property serves as a pivotal element in establishing global convergence.

In our pursuit of establishing the global convergence of SNVI, we proceed by assuming that f_t exhibits star-convexity, which represents an extensive class of nonconvex functions that encompasses convexity as a particular instance (Hinder, Sidford, and Sohoni 2020; Lee and Valiant 2016; Nesterov and Polyak 2006; Zhou et al. 2019).

Assumption 1 (Star-Convexity)**.** Let v^* be the solution of $F(v) = 0$. For every v_t generated by (10), we have:

$$f_t(v^*) \geq f_t(v_t) + \langle \nabla f_t(v_t), v^* - v_t \rangle$$

With the foundation laid by the aforementioned assumptions and properties, we are now ready to present our primary convergence result for SNVI.

Theorem 4. *Let v^* satisfy Assumption 1, we have:*

$$\mathbb{E}\left[\min_{t=0,\ldots,n-1} \|F(v_t)\|^2_{\mathbb{E}[H_S(v_t)]}\right] \leq \frac{\|v_0 - v^*\|^2}{n\alpha(1-\alpha)}$$

Furthermore, if the stochastic function $f_{S,v}(v)$ is star-convex along the iterates v_t:

Assumption 2. $f_{S,y}(x)$ is star-convex along the iterates v_t, i.e.

$$f_{S_t,v_t}(v^*) \geq f_{S_t,v_t}(v_t) + \langle \nabla f_{S_t,v_t}(v_t), v^* - v_t \rangle.$$

Together with the assumption that $\mathbb{E}\left[H_S(x)\right]$ is invertible, we can establish the sublinear convergence rate of both the successive relaxation Bellman operator and the value function for SNVI.

Lemma 3. *Under the assumptions of 1 and 2, there exists $\lambda > 0$ s.t*

$$\lambda \mathbb{E}\left[\min_{t=0,\ldots,n-1} \|F(v_t)\|^2_\infty\right] \leq \frac{\|v_0 - v^*\|^2}{n\alpha(1-\alpha)}.$$

Theorem 5 (Sublinear convergence)**.** *Under Assumption 1 and 2, from Lemma 3, we know that when $\omega < 1$, $n \geq \frac{\|v_0 - v^*\|^2}{\lambda\alpha(1-\alpha)(\omega-\omega\gamma)\epsilon^2}$, we have:*

$$\mathbb{E}\left[\min_{t=0,\ldots,n-1} \|(v_t - v^*)\|_\infty\right] \leq \epsilon. \tag{13}$$

We have now established that SNVI possesses a global sublinear convergence rate. This insight, drawn from our comprehensive analysis, serves as a strong theoretical foundation, affirming the practical effectiveness and efficiency of SNVI in real-world scenarios. It is worth noting that deriving Assumption 1 and 2 theoretically is challenging, fortunately, we can empirically verify these two assumptions.

Experiments

In this section, we present experimental results that highlight the advantages of our proposed algorithms compared to both value iteration and G-SOVI.

Experiment Setup

We have partially drawn inspiration from the experimental setups outlined in (Goyal and Grand-Clement 2023) and (Kamanchi, Diddigi, and Bhatnagar 2021). Specifically, we utilize the MDP toolbox framework (Cordwell, Gonzalez, and Tulabandhula 2015) to implement our algorithms and create a variety of MDP instances for our experiments. The stopping criterion for all the algorithms is based on achieving a difference between the current value function and the optimal value function within a threshold of $\epsilon(1-\gamma)$, given by the inequality:

$$e_t = \|v_t - v^*\|_\infty \leq \epsilon(1-\gamma).$$

Throughout our experiments, we set ϵ to 0.1 and initialize all algorithms with $v_0 = \mathbf{0}$. In order to effectively portray the results, we utilize logarithmic scaling for both the vertical axis representing the error and the horizontal axis representing the running time in seconds. Notably, for the purpose of preserving numerical stability, we gradually increase

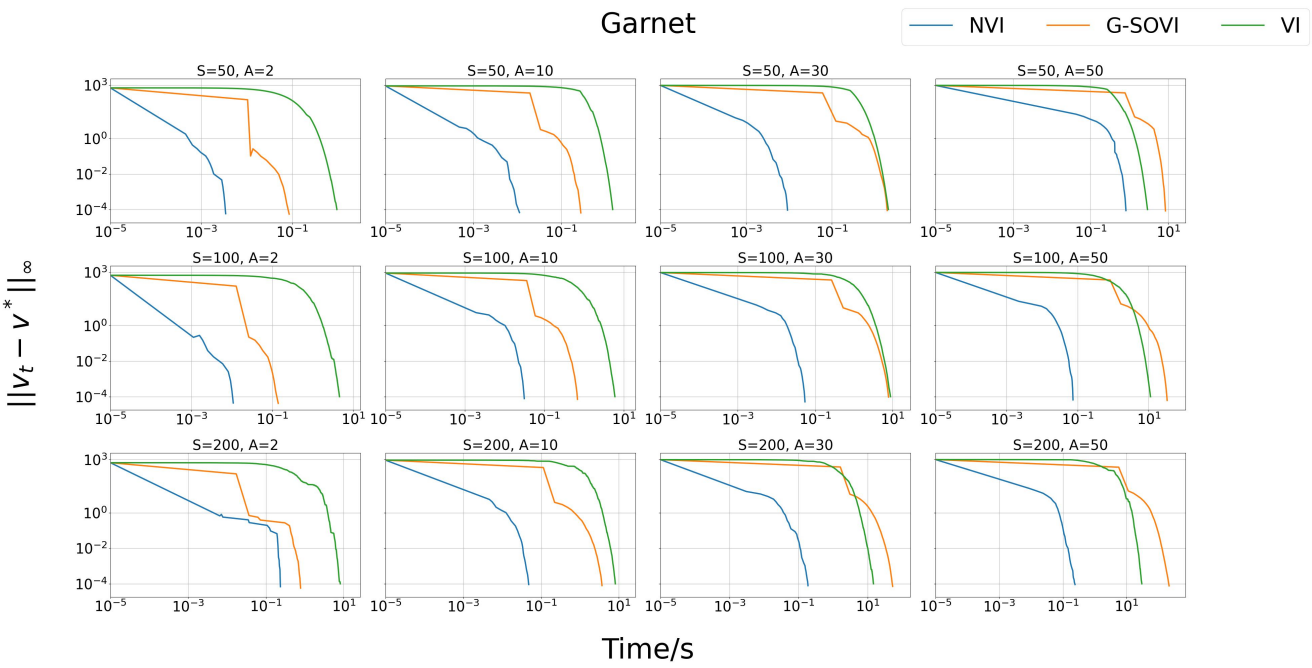

Figure 1: Comparison with G-SOVI on Garnet MDP instances with different action and state space dimensions.

the soft max parameter β. We apply the sketching matrix $S_t \in \{0,1\}^{d\times s}$ as described in (Gower et al. 2019), which contains precisely one non-zero entry per row and column.

All experiments are conducted in Python on a Mac OS laptop equipped with a 3.2 GHz 8-Core Apple M1 processor. These experimental settings ensure consistency and reliability in evaluating the performance of our algorithms and comparing them against existing methods.

MDP Instances

We use two structured (*forest* and *machine replacement*) and some random (*Garnet*) MDP instances as benchmark problems. The *forest* MDP is influenced by the application of dynamic programming in optimizing fire management strategies (Possingham and Tuck 1997). Machine replacement is another widely used MDP representative instance (Delage and Mannor 2010; Wiesemann, Kuhn, and Rustem 2013; Clement and Kroer 2021). And Generalized Average Reward Non-stationary Environment Test-bench (Garnet MDPs) (Archibald, McKinnon, and Thomas 1995) offers a representative set of abstract MDPs that serve as a common benchmark. The Garnet MDPs are widely used as benchmarks in both MDP-solving (Goyal and Grand-Clement 2023) and reinforcement learning (Qian et al. 2019; Tarbouriech and Lazaric 2019) algorithm evaluations.

Verification of Assumptions

It suffices to verify Ass. 2, which implies Ass. 1. We verify that $\Delta_t = f_{S_t,v_t}(v^*) - f_{S_t,v_t}(v_t) - \langle \nabla f_{S_t,v_t}(v_t), v^* - v_t\rangle \geq 0$ along the iterates. We plot the result on Forest and Machine Replacement MDP instances both with $S = 5000$ as a representative in Figure 2, showing that Ass. 2 generally holds, except for very few violation points of small sketching size.

Evaluation of Advantages

Scalability on large action space. In order to highlight the advantages of our algorithm over G-SOVI in MDPs with high-dimensional action spaces, we have randomly generated a set of 12 *Garnet* MDP instances. These instances encompass varying dimensions for both the state and action spaces, including 3 different state dimensions and 4 different action dimensions.Upon examining Figure 1, it is evident that NVI consistently outperforms G-SOVI and VI across these instances. Furthermore, as the dimension of the action space increases, the disparity in running time between NVI and G-SOVI becomes more pronounced. Notably, when the action space dimension is small, G-SOVI demonstrates a superior performance compared to VI. However, as the action space dimension becomes substantial—even with just a few tens of dimensions—G-SOVI's performance declines to a level lower than that of VI.

Scalability on large state space. To effectively demonstrate that SNVI exhibits superior scalability in larger state space dimensions compared to NVI and converges faster than VI, we conducted experiments on *Forest* MDP instances. It is worth noting that *Forest* MDP instances have a fixed action dimension. In Figure 3, the numerical value following 'SNVI' represents the sketching rate, denoted as $\frac{k}{|S|}$.Upon analyzing the results, it is evident that both NVI and SNVI converge significantly faster than VI across these *Forest* MDP instances. Furthermore, with an appropriate sketch size, SNVI demonstrates faster convergence than NVI. This computational advantage becomes increasingly pronounced as the dimension of the state space grows. Ad-

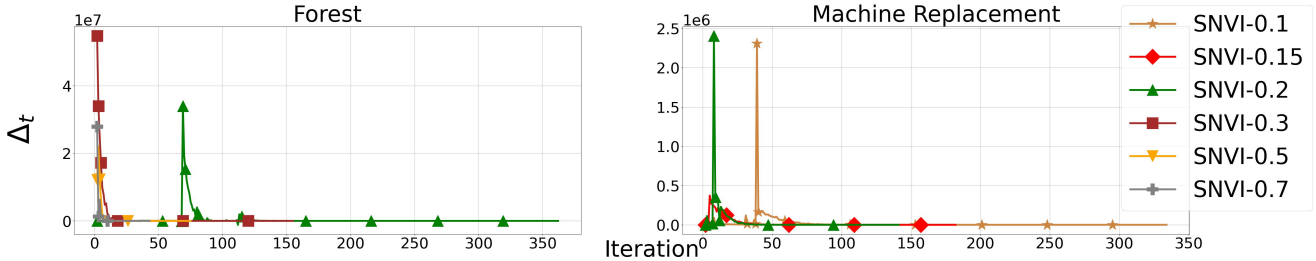

Figure 2: Empirical verification of assumption 1 and 2 on Forest and Machine Replacement MDP instances.

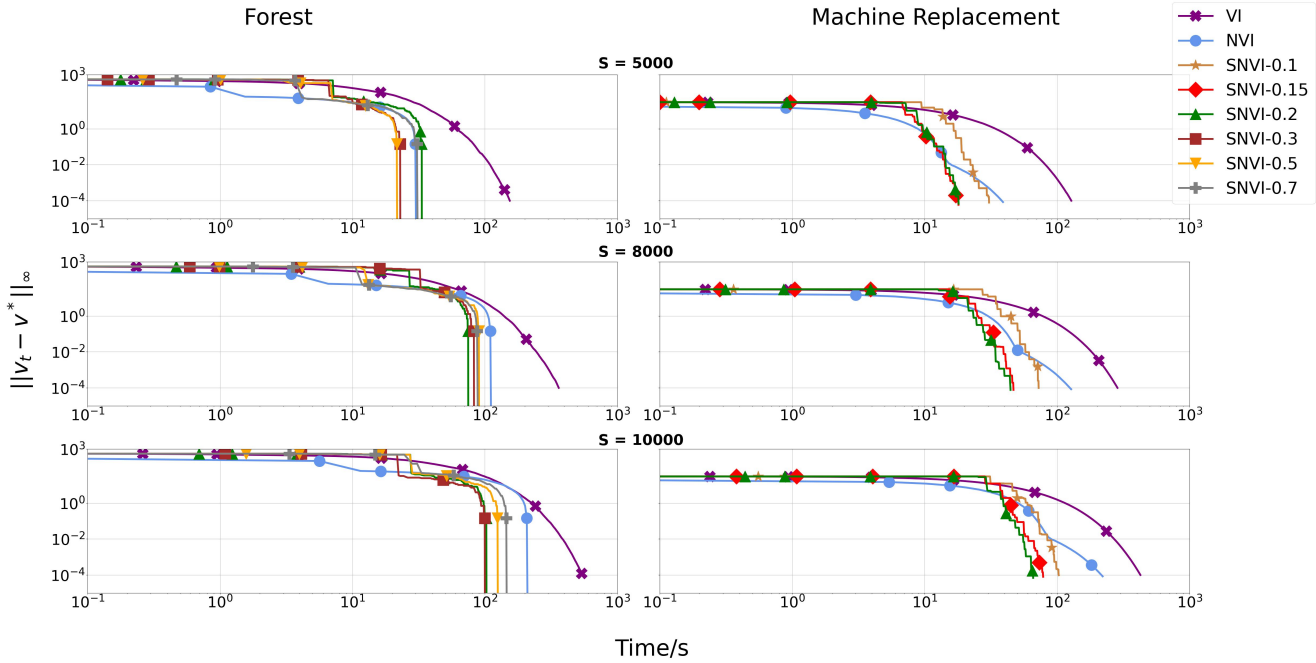

Figure 3: Comparison with VI on Forest and Machine Replacement MDP instances with different state space dimension.

ditionally, the observed trend indicates that as the dimension of the state space increases, a smaller proper sketch ratio suffices. This aligns with our earlier analysis, wherein SNVI's reduction of computational complexity from $\mathcal{O}(|S|^3)$ to $\mathcal{O}(|k|^3)$ is substantiated.

Based on these observations, we can observe that NVI exhibits significantly better scalability in scenarios involving large action spaces compared to G-SOVI. This result demonstrates the practical advantage of NVI in handling high-dimensional action spaces. Moreover, The computational results substantiate the notion that a judicious balance between computational cost and solution quality can indeed lead to a reduction in the total running time. This observation underscores the practical significance of SNVI in handling larger state space dimensions with enhanced efficiency and effectiveness.

In essence, our findings not only confirm the theoretical analysis of NVI and SNVI but also highlights their practical effectiveness in the realm of addressing MDP instances with large state and action spaces.

Conclusion

In this paper, we introduce the Newton Value Iteration (NVI) algorithm, which effectively tackles the challenge posed by action space dimensions and paves the way for handling MDPs with large action spaces. Building upon NVI, we present a Sketched Newton Value Iteration (SNVI) approach, showcasing robust performance both in theoretical analyses and experimental evaluations.

It should be mentioned that SNVI's theoretical convergence rate, being sublinear, still leaves room for enhancement. This is evident as its convergence rate is slower than the linear rate of VI, even though its practical performance surpasses VI's in experiments. We posit that the incorporation of techniques such as restarts could potentially accelerate the convergence of SNVI to a linear rate.

A promising avenue for future exploration involves extending the benefits of SNVI to model-free reinforcement learning algorithms. While our current work primarily focuses on solving model-based MDPs, extending SNVI to model-free MDPs holds the promise of further enriching the applicability and impact of our findings.

Acknowledgments

This research is partially supported by the National Natural Science Foundation of China (NSFC) [Grant NSFC-72150001, 72225009, 72394360, 72394365].

References

Agarwal, R. P.; Meehan, M.; and O'regan, D. 2001. *Fixed point theory and applications*, volume 141. Cambridge university press.

Archibald, T.; McKinnon, K.; and Thomas, L. 1995. On the generation of markov decision processes. *Journal of the Operational Research Society*, 46(3): 354–361.

Bellman, R. 1966. Dynamic programming. *Science*, 153(3731): 34–37.

Bertsekas, D. 2012. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific.

Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Clement, J. G.; and Kroer, C. 2021. First-order methods for Wasserstein distributionally robust MDP. In *International Conference on Machine Learning*, 2010–2019. PMLR.

Conn, A. R.; Gould, N. I.; and Toint, P. L. 2000. *Trust region methods*. SIAM.

Cordwell, S.; Gonzalez, Y.; and Tulabandhula, T. 2015. Markov Decision Process (MDP) toolbox for python. https://github.com/sawcordwell/pymdptoolbox.

Delage, E.; and Mannor, S. 2010. Percentile optimization for Markov decision processes with parameter uncertainty. *Operations research*, 58(1): 203–213.

D'ambrosio, C.; Liberti, L.; Poirion, P.-L.; and Vu, K. 2020. Random projections for quadratic programs. *Mathematical Programming*, 183(1-2): 619–647.

Ghadimi, E.; Feyzmahdavian, H. R.; and Johansson, M. 2015. Global convergence of the heavy-ball method for convex optimization. In *2015 European control conference (ECC)*, 310–315. IEEE.

Gower, R.; Kovalev, D.; Lieder, F.; and Richtárik, P. 2019. RSN: randomized subspace Newton. *Advances in Neural Information Processing Systems*, 32.

Goyal, V.; and Grand-Clement, J. 2023. A first-order approach to accelerated value iteration. *Operations Research*, 71(2): 517–535.

Grand-Clément, J. 2021. From convex optimization to MDPs: A review of first-order, second-order and quasi-Newton methods for MDPs. *arXiv preprint arXiv:2104.10677*.

Hambly, B.; Xu, R.; and Yang, H. 2023. Recent advances in reinforcement learning in finance. *Mathematical Finance*, 33(3): 437–503.

Hanzely, F.; Doikov, N.; Nesterov, Y.; and Richtarik, P. 2020. Stochastic subspace cubic Newton method. In *International Conference on Machine Learning*, 4027–4038. PMLR.

Harris, C. R.; Millman, K. J.; Van Der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; et al. 2020. Array programming with NumPy. *Nature*, 585(7825): 357–362.

Hinder, O.; Sidford, A.; and Sohoni, N. 2020. Near-optimal methods for minimizing star-convex functions and beyond. In *Conference on learning theory*, 1894–1938. PMLR.

Jorge, N.; and Stephen, J. W. 2006. *Numerical optimization*. Spinger.

KALABA, R. E. 1957. *On nonlinear differential equations, the maximum operation, and monotone convergence*. New York University.

Kamanchi, C.; Diddigi, R. B.; and Bhatnagar, S. 2021. Generalized Second-Order Value Iteration in Markov Decision Processes. *IEEE Transactions on Automatic Control*, 67(8): 4241–4247.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 4909–4926.

Lee, J. C.; and Valiant, P. 2016. Optimizing star-convex functions. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, 603–614. IEEE.

Nesterov, Y. 2003. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Nesterov, Y.; and Polyak, B. T. 2006. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1): 177–205.

Ortega, J. M.; and Rheinboldt, W. C. 2000. *Iterative solution of nonlinear equations in several variables*. SIAM.

Pollatschek, M.; and Avi-Itzhak, B. 1969. Algorithms for stochastic games with geometrical interpretation. *Management Science*, 15(7): 399–415.

Possingham, H.; and Tuck, G. 1997. Application of stochastic dynamic programming to optimal fire management of a spatially structured threatened species. In *Proceedings international congress on modelling and simulation*, volume 2. Modeling and Simulation Society of Australia Canberra.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Puterman, M. L.; and Brumelle, S. L. 1979. On the convergence of policy iteration in stationary dynamic programming. *Mathematics of Operations Research*, 4(1): 60–69.

Qian, J.; Fruit, R.; Pirotta, M.; and Lazaric, A. 2019. Exploration bonus for regret minimization in discrete and continuous average reward mdps. *Advances in Neural Information Processing Systems*, 32.

Reetz, D. 1973. Solution of a Markovian decision problem by successive overrelaxation. *Zeitschrift für Operations Research*, 17: 29–32.

Tamar, A.; Wu, Y.; Thomas, G.; Levine, S.; and Abbeel, P. 2016. Value iteration networks. *Advances in neural information processing systems*, 29.

Tarbouriech, J.; and Lazaric, A. 2019. Active exploration in markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 974–982. PMLR.

Watkins, C. J.; and Dayan, P. 1992. Q-learning. *Machine learning*, 8: 279–292.

Wiesemann, W.; Kuhn, D.; and Rustem, B. 2013. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1): 153–183.

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

Yuan, R.; Lazaric, A.; and Gower, R. M. 2022. Sketched Newton–Raphson. *SIAM Journal on Optimization*, 32(3): 1555–1583.

Yuan, Y.-x. 2015. Recent advances in trust region algorithms. *Mathematical Programming*, 151: 249–281.

Zhang, C.; Ge, D.; Jiang, B.; and Ye, Y. 2022. DRSOM: A Dimension Reduced Second-Order Method and Preliminary Analyses. *arXiv preprint arXiv:2208.00208*.

Zhang, J.; O'Donoghue, B.; and Boyd, S. 2020. Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations. *SIAM Journal on Optimization*, 30(4): 3170–3197.

Zhou, Y.; Yang, J.; Zhang, H.; Liang, Y.; and Tarokh, V. 2019. Sgd converges to global minimum in deep learning via star-convex path. *arXiv preprint arXiv:1901.00451*.