Implicit Finite-Horizon Approximation and Efficient Optimal Algorithms for Stochastic Shortest Path

Liyu Chen, Mehdi Jafarnia-Jahromi, Rahul Jain, Haipeng Luo

University of Southern California

October 8, 2021

Motivation

Many MDP models have been studied:

- Infinite horizon average reward model (Bartlett & Tewari, 2009; Jaksch et al., 2010)
- Infinite horizon discounted model (Even-Dar et al., 2003; Strehl et al., 2006)
- Finite horizon model (Osband and Van Roy, 2016; Azar et al., 2017; Jin et al., 2018)

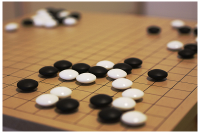
Motivation

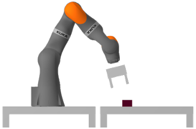
Many MDP models have been studied:

- Infinite horizon average reward model (Bartlett & Tewari, 2009; Jaksch et al., 2010)
- Infinite horizon discounted model (Even-Dar et al., 2003; Strehl et al., 2006)
- Finite horizon model (Osband and Van Roy, 2016; Azar et al., 2017; Jin et al., 2018)

However, there are many real-world applications not modelled well by the above:

- Games (such as Go)
- Car navigation
- Robotic manipulation





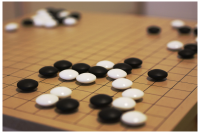
Motivation

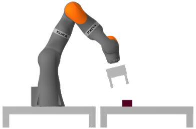
Many MDP models have been studied:

- Infinite horizon average reward model (Bartlett & Tewari, 2009; Jaksch et al., 2010)
- Infinite horizon discounted model (Even-Dar et al., 2003; Strehl et al., 2006)
- Finite horizon model (Osband and Van Roy, 2016; Azar et al., 2017; Jin et al., 2018)

However, there are many real-world applications not modelled well by the above:

- Games (such as Go)
- Car navigation
- Robotic manipulation





For these, Stochastic Shortest Path (SSP) is a better model.

- Episodic MDP with a goal state.
- Ends interaction only when the goal state is reached

Related Works

S: #states, A: #actions, K: #episodes, D: SSP-diameter
$c_{\min}$: minimum cost, $B_\star$: maximum expected cost of optimal policy over all states
$T_\star$: maximum expected hitting time of optimal policy starting from any state

- UC-SSP (Tarbouriech et al., 2020): $\tilde{\mathcal{O}}\left(DS\sqrt{\frac{D}{c_{\min}}AK} + S^2AD^2\right)$
- Bernstein-SSP (Cohen et al., 2020): $\tilde{\mathcal{O}}\left(B_\star S\sqrt{AK} + \sqrt{\frac{B_\star^3 S^2 A^2}{c_{\min}}}\right)$
- ULCVI (Cohen et al., 2021): $\tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + T_\star^4 S^2 A\right)$ **(Minimax Optimal)**
- EB-SSP (Tarbouriech et al., 2021): $\tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + B_\star S^2 A\right)$ **(Minimax Optimal)**
- Lower Bound (Cohen et al., 2020): $\Omega(B_\star\sqrt{SAK})$

Related Works

S: #states, A: #actions, K: #episodes, D: SSP-diameter
$c_{\min}$: minimum cost, $B_\star$: maximum expected cost of optimal policy over all states
$T_\star$: maximum expected hitting time of optimal policy starting from any state

- UC-SSP (Tarbouriech et al., 2020): $\tilde{\mathcal{O}}\left(DS\sqrt{\frac{D}{c_{\min}}AK} + S^2AD^2\right)$
- Bernstein-SSP (Cohen et al., 2020): $\tilde{\mathcal{O}}\left(B_\star S\sqrt{AK} + \sqrt{\frac{B_\star^3S^2A^2}{c_{\min}}}\right)$
- ULCVI (Cohen et al., 2021): $\tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + T_\star^4S^2A\right)$ **(Minimax Optimal)**
- EB-SSP (Tarbouriech et al., 2021): $\tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + B_\star S^2A\right)$ **(Minimax Optimal)**
- Lower Bound (Cohen et al., 2020): $\Omega(B_\star\sqrt{SAK})$

Techniques applied in previous works are quite different from each other, and some of these algorithms are fairly complicated.

Our Results

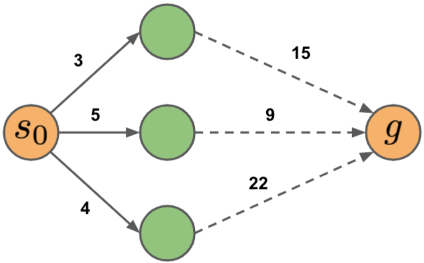
Our contribution: A generic template for regret minimization algorithms in SSP. Using this template, we develop two algorithms:

S: #states, A: #actions, D: SSP-diameter, K: #episodes
$T_\star$: expected hitting time of optimal policy, $c_{\min}$: minimum cost

	SVI-SSP	LCB-ADVANTAGE-SSP
Regret	$\tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + B_\star S^2 A\right)$	$\tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + B_\star^5 S^2 A / c_{\min}^4\right)$
Algorithm type	Model-based	Model-free (the first)

Problem Formulation

SSP Model: MDP $M = (\mathcal{S}, \mathcal{A}, s_{\text{init}}, g, c, P)$, only P is unknown.



Problem Formulation

SSP Model: MDP $M = (\mathcal{S}, \mathcal{A}, s_{\text{init}}, g, c, P)$, only P is unknown.

Learning Protocol

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

  learner starts in state $s_1^k = s_{\text{init}}$, $i \leftarrow 1$

  while $s_i^k \neq g$ **do**

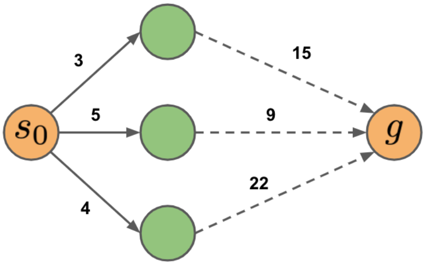
    learner chooses action $a_i^k \in \mathcal{A}$ and observes states $s_{i+1}^k \sim P(\cdot|s_i^k, a_i^k)$

    $i \leftarrow i + 1$

  end

  learner suffers cost $\sum_{i=1}^{I_k} c(s_i^k, a_i^k)$

end



Problem Formulation

SSP Model: MDP $M = (\mathcal{S}, \mathcal{A}, s_{\text{init}}, g, c, P)$, only P is unknown.

Notations:

- Policy π: maps state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}$
  - Proper: reaches g with probability 1

Problem Formulation

SSP Model: MDP $M = (\mathcal{S}, \mathcal{A}, s_{\text{init}}, g, c, P)$, only P is unknown.

Notations:

- Policy π: maps state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}$
  - Proper: reaches g with probability 1
- Value function $V^{\pi}(s) = \mathbb{E}[\sum_{i=1}^{I} c(s^i, a^i) | P, \pi, s^1 = s]$

Problem Formulation

SSP Model: MDP $M = (\mathcal{S}, \mathcal{A}, s_{\text{init}}, g, c, P)$, only P is unknown.

Notations:

- Policy π: maps state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}$
    - Proper: reaches g with probability 1
- Value function $V^\pi(s) = \mathbb{E}[\sum_{i=1}^{I} c(s^i, a^i) | P, \pi, s^1 = s]$

Objective: minimize regret w.r.t. the best proper policy in hindsight

$$R_K = \sum_{k=1}^{K} \left(\sum_{i=1}^{I_k} c(s_k^i, a_k^i) - V^{\pi^\star}(s_0) \right),$$

where $\pi^\star = \operatorname{argmin}_{\pi \in \Pi_{\text{proper}}} V^\pi(s_0)$.

Generic Template

A General Algorithmic Template for SSP

Initialize: $t \leftarrow 0$, $s_1 \leftarrow s_{\text{init}}$, $Q(s,a) \leftarrow c(s,a)$ for all (s,a).
for $k = 1, \ldots, K$ **do**
  repeat
    Increment time step $t \overset{+}{\leftarrow} 1$.
    Take action $a_t = \operatorname{argmin}_a Q(s_t, a)$, suffer cost $c(s_t, a_t)$, and transit to s'_t.
    Update Q (so that it satisfies Property 1 and Property 2).
    if $s'_t \neq g$ **then** $s_{t+1} \leftarrow s'_t$; **else** $s_{t+1} \leftarrow s_{\text{init}}$, **break**.
end
Record $T \leftarrow t$ (that is, the total number of steps).

Generic Template

A General Algorithmic Template for SSP

Initialize: $t \leftarrow 0$, $s_1 \leftarrow s_{\text{init}}$, $Q(s,a) \leftarrow c(s,a)$ for all (s,a).
for $k = 1, \ldots, K$ **do**
  repeat
    Increment time step $t \overset{+}{\leftarrow} 1$.
    Take action $a_t = \text{argmin}_a Q(s_t, a)$, suffer cost $c(s_t, a_t)$, and transit to s'_t.
    Update Q (so that it satisfies Property 1 and Property 2).
    if $s'_t \neq g$ **then** $s_{t+1} \leftarrow s'_t$; **else** $s_{t+1} \leftarrow s_{\text{init}}$, **break**.
end
Record $T \leftarrow t$ (that is, the total number of steps).

The key of analysis: Bounding the estimation error $Q^\star(s_t, a_t) - Q(s_t, a_t)$.

Generic Template

A General Algorithmic Template for SSP

Initialize: $t \leftarrow 0$, $s_1 \leftarrow s_{\text{init}}$, $Q(s,a) \leftarrow c(s,a)$ for all (s,a).
for $k = 1, \ldots, K$ **do**
  repeat
    Increment time step $t \overset{+}{\leftarrow} 1$.
    Take action $a_t = \text{argmin}_a\, Q(s_t, a)$, suffer cost $c(s_t, a_t)$, and transit to s'_t.
    Update Q (so that it satisfies Property 1 and Property 2).
    if $s'_t \neq g$ **then** $s_{t+1} \leftarrow s'_t$; **else** $s_{t+1} \leftarrow s_{\text{init}}$, **break**.
end
Record $T \leftarrow t$ (that is, the total number of steps).

The key of analysis: Bounding the estimation error $Q^\star(s_t, a_t) - Q(s_t, a_t)$.
Issue: Relatively straightforward in a discounted setting or a finite-horizon setting, but becomes highly non-trivial in SSP.

Implicit Finite Horizon Approximation

Solution: approximate an SSP instance M with a finite-horizon counterpart $\widetilde{M}$.

- It corresponds to interacting with M for H steps, and then teleporting to the goal state.

Implicit Finite Horizon Approximation

Solution: approximate an SSP instance M with a finite-horizon counterpart $\widetilde{M}$.

- It corresponds to interacting with M for H steps, and then teleporting to the goal state.
- We only need the optimal value functions of $\widetilde{M}$ in the analysis:

$$Q^\star_h(s,a) = c(s,a) + P_{s,a}V^\star_{h-1}, \qquad V^\star_h(s) = \min_a Q^\star_h(s,a),$$

with $Q^\star_0(s,a) = 0$ for all (s,a).

Implicit Finite Horizon Approximation

Solution: approximate an SSP instance M with a finite-horizon counterpart $\widetilde{M}$.

- It corresponds to interacting with M for H steps, and then teleporting to the goal state.
- We only need the optimal value functions of $\widetilde{M}$ in the analysis:

$$Q^\star_h(s,a) = c(s,a) + P_{s,a}V^\star_{h-1}, \qquad V^\star_h(s) = \min_a Q^\star_h(s,a),$$

with $Q^\star_0(s,a) = 0$ for all (s,a).

Lemma

For any value of H, $Q^\star_H(s,a) \le Q^\star(s,a)$ holds for all (s,a). For any $\delta \in (0,1)$, if $H \ge \frac{4B_\star}{c_{\min}}\ln(2/\delta) + 1$, then $Q^\star(s,a) \le Q^\star_H(s,a) + B_\star\delta$ holds for all (s,a).

Implicit Finite Horizon Approximation

Solution: approximate an SSP instance M with a finite-horizon counterpart $\widetilde{M}$.

- It corresponds to interacting with M for H steps, and then teleporting to the goal state.
- We only need the optimal value functions of $\widetilde{M}$ in the analysis:

$$Q^\star_h(s,a) = c(s,a) + P_{s,a} V^\star_{h-1}, \qquad V^\star_h(s) = \min_a Q^\star_h(s,a),$$

with $Q^\star_0(s,a) = 0$ for all (s,a).

Lemma

For any value of H, $Q^\star_H(s,a) \leq Q^\star(s,a)$ holds for all (s,a). For any $\delta \in (0,1)$, if $H \geq \frac{4B_\star}{c_{\min}} \ln(2/\delta) + 1$, then $Q^\star(s,a) \leq Q^\star_H(s,a) + B_\star \delta$ holds for all (s,a).

Similar approximation has been done explicitly before (Chen et al., 2021a; Chen et al., 2021b; Cohen et al., 2021)

Implicit Finite Horizon Approximation

To perform approximation implicitly, we need the following two properties of estimate Q (let Q_t be the value of Q at the beginning of time step t):

- **Property 1** (Optimism): with high probability, $Q_t(s, a) \leq Q^\star(s, a)$ for all (s, a), $t \geq 1$.

Implicit Finite Horizon Approximation

To perform approximation implicitly, we need the following two properties of estimate Q (let Q_t be the value of Q at the beginning of time step t):

- **Property 1** (Optimism): with high probability, $Q_t(s,a) \le Q^\star(s,a)$ for all (s,a), $t \ge 1$.
- **Property 2** (Recursion): There exists a "bonus overhead" $\xi_H > 0$ and an absolute constant $d > 0$ such that the following holds with high probability:

$$\sum_{t=1}^{T}(Q_h^\star(s_t,a_t) - Q_t(s_t,a_t)) \le \xi_H + \left(1+\frac{d}{H}\right)\sum_{t=1}^{T}(V_{h-1}^\star(s_t) - Q_t(s_t,a_t))_+,$$
$$\sum_{t=1}^{T}(Q^\star(s_t,a_t) - Q_t(s_t,a_t)) \le \xi_H + \left(1+\frac{d}{H}\right)\sum_{t=1}^{T}(V^\star(s_t) - Q_t(s_t,a_t))_+,$$

where $(x)_+ = \max\{x, 0\}$.

Implicit Finite Horizon Approximation

Theorem

For any $\delta \in (0,1)$, if $H \geq \frac{4B_\star}{c_{\min}} \ln(2/\delta) + 1$, then the template ensures (with high probability)
$R_K = \tilde{\mathcal{O}}\left(\sqrt{B_\star C_K} + B_\star + \delta C_K + \xi_H\right)$*, where* $C_K = \sum_{k=1}^{K} \sum_{i=1}^{I_k} c(s_i^k, a_i^k)$.

Implicit Finite Horizon Approximation

Theorem

For any $\delta \in (0,1)$, *if* $H \geq \frac{4B_\star}{c_{\min}} \ln(2/\delta) + 1$, *then the template ensures (with high probability)* $R_K = \tilde{\mathcal{O}}\left(\sqrt{B_\star C_K} + B_\star + \delta C_K + \xi_H\right)$, *where* $C_K = \sum_{k=1}^{K} \sum_{i=1}^{I_k} c(s_i^k, a_i^k)$.

Now if we ensure $\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K}\right)$ (with appropriate bonus), then $R_K = \tilde{\mathcal{O}}\left(B_\star\sqrt{SAK}\right)$.

No explicit implementation of $\widetilde{M}$ is required!

Optimal and Efficient Model-based Algorithm: SVI-SSP

Update $Q(s, a)$ for logarithmically many times for each (s, a).

Optimal and Efficient Model-based Algorithm: SVI-SSP

Update $Q(s,a)$ for logarithmically many times for each (s,a).
When updating $Q(s,a)$, we apply the following update rule:

$$Q(s,a) \leftarrow \max\left\{c(s,a) + \bar{P}_{s,a}V - b, Q(s,a)\right\},$$

where $\bar{P}$ is the empirical transition, $b \approx \max\left\{7\sqrt{\frac{\mathbb{V}(\bar{P}_{s,a},V)}{n}}, \frac{49B_\star}{n}\right\}$ (Zhang et al., 2021).

Optimal and Efficient Model-based Algorithm: SVI-SSP

Theorem

SVI-SSP *satisfies Property 1 and Property 2 with* $d = 1$ *and*
$\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + B_\star S^2 A + \delta C_K\right)$.

Optimal and Efficient Model-based Algorithm: SVI-SSP

Theorem

SVI-SSP *satisfies Property 1 and Property 2 with* $d = 1$ *and*
$\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + B_\star S^2 A + \delta C_K\right)$.

Theorem

SVI-SSP *ensures* $R_K = \tilde{\mathcal{O}}(B_\star\sqrt{SAK} + B_\star S^2 A)$.

Optimal and Efficient Model-based Algorithm: SVI-SSP

Theorem

SVI-SSP *satisfies Property 1 and Property 2 with* $d = 1$ *and*
$\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + B_\star S^2 A + \delta C_K\right)$.

Theorem

SVI-SSP *ensures* $R_K = \tilde{\mathcal{O}}(B_\star\sqrt{SAK} + B_\star S^2 A)$.

- Minimax optimal, matching the result of EB-SSP (Tarbouriech et al., 2021).

Optimal and Efficient Model-based Algorithm: SVI-SSP

Theorem

SVI-SSP *satisfies Property 1 and Property 2 with* $d = 1$ *and*
$\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + B_\star S^2 A + \delta C_K\right)$.

Theorem

SVI-SSP *ensures* $R_K = \tilde{\mathcal{O}}(B_\star\sqrt{SAK} + B_\star S^2 A)$.

- Minimax optimal, matching the result of EB-SSP (Tarbouriech et al., 2021).
- Can be made parameter-free using doubling trick (Tarbouriech et al., 2021).

Optimal and Efficient Model-based Algorithm: SVI-SSP

Theorem

SVI-SSP *satisfies Property 1 and Property 2 with* $d = 1$ *and* $\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + B_\star S^2 A + \delta C_K\right)$.

Theorem

SVI-SSP *ensures* $R_K = \tilde{\mathcal{O}}(B_\star\sqrt{SAK} + B_\star S^2 A)$.

- Minimax optimal, matching the result of EB-SSP (Tarbouriech et al., 2021).
- Can be made parameter-free using doubling trick (Tarbouriech et al., 2021).
- Lower time complexity of updates: SVI-SSP: $\tilde{\mathcal{O}}(B_\star S^2 A/c_{\min})$, EB-SSP: $\tilde{\mathcal{O}}(B_\star^2 S^5 A/c_{\min}^2)$, ULCVI: $\tilde{\mathcal{O}}(S^2 A T_\star K)$

Optimal and Efficient Model-based Algorithm: SVI-SSP

Theorem

SVI-SSP *satisfies Property 1 and Property 2 with* $d = 1$ *and*
$\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + B_\star S^2 A + \delta C_K\right)$.

Theorem

SVI-SSP *ensures* $R_K = \tilde{\mathcal{O}}(B_\star\sqrt{SAK} + B_\star S^2 A)$.

- Minimax optimal, matching the result of EB-SSP (Tarbouriech et al., 2021).
- Can be made parameter-free using doubling trick (Tarbouriech et al., 2021).
- Lower time complexity of updates: SVI-SSP: $\tilde{\mathcal{O}}(B_\star S^2 A/c_{\min})$, EB-SSP: $\tilde{\mathcal{O}}(B_\star^2 S^5 A/c_{\min}^2)$, ULCVI: $\tilde{\mathcal{O}}(S^2 A T_\star K)$

Our implicit finite horizon analysis is the key to achieve sparse updates.

The First Model-free Algorithm: LCB-ADVANTAGE-SSP

Update $Q(s, a)$ for logarithmically many times for each (s, a).

The First Model-free Algorithm: LCB-ADVANTAGE-SSP

Update $Q(s,a)$ for logarithmically many times for each (s,a).
Inspired by (Zhang et al., 2020), we update $Q(s,a)$ with the following variance reduced update rule (approximately)

$$Q(s,a) \leftarrow \max\left\{ c(s,a) + \frac{1}{n}\sum_{i=1}^{n} V^{\text{ref}}(s'_{t_i}) + \frac{1}{m}\sum_{i=1}^{m}\left(V(s'_{t'_i}) - V^{\text{ref}}(s'_{t'_i})\right) - b, Q(s,a) \right\},$$

where m is the number of samples in current stage, and n is the number of samples up to current stage, and $V(s) = \min_a Q(s,a)$.

The First Model-free Algorithm: LCB-Advantage-SSP

Theorem

LCB-Advantage-SSP *satisfies Property 1 and Property 2 with* $d = 3$ *and*
$\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + \frac{B_\star^2 H^3 S^2 A}{c_{\min}}\right)$.

The First Model-free Algorithm: LCB-Advantage-SSP

Theorem

LCB-Advantage-SSP *satisfies Property 1 and Property 2 with* $d = 3$ *and* $\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + \frac{B_\star^2 H^3 S^2 A}{c_{\min}}\right)$.

Theorem

LCB-Advantage-SSP *ensures* $R_K = \tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + \frac{B_\star^5 S^2 A}{c_{\min}^4}\right)$.

The First Model-free Algorithm: LCB-ADVANTAGE-SSP

Theorem

LCB-ADVANTAGE-SSP *satisfies Property 1 and Property 2 with* $d = 3$ *and*
$\xi_H = \tilde{\mathcal{O}}\left(\sqrt{B_\star SAC_K} + \frac{B_\star^2 H^3 S^2 A}{c_{\min}}\right)$.

Theorem

LCB-ADVANTAGE-SSP *ensures* $R_K = \tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + \frac{B_\star^5 S^2 A}{c_{\min}^4}\right)$.

- To make it parameter-free, we try logarithmically many different values of parameters simultaneously, each leading to a different update rule for Q and V^{ref}.

Summary

Our contribution: A generic template for regret minimization algorithms in SSP. Using this template, we develop two algorithms:

S: #states, A: #actions, D: SSP-diameter, K: #episodes
$T_\star$: expected hitting time of optimal policy, $c_{\min}$: minimum cost

	SVI-SSP	LCB-ADVANTAGE-SSP
Regret	$\tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + B_\star S^2 A\right)$	$\tilde{\mathcal{O}}\left(B_\star\sqrt{SAK} + B_\star^5 S^2 A/c_{\min}^4\right)$
Algorithm type	Model-based	Model-free (the first)