Achieving $\tilde{O}(1/\epsilon)$ Sample Complexity for Constrained Markov Decision Process

Jiashuo Jiang (HKUST)

Joint work with Yinyu Ye

22 October, 2024

Constrained MDP

- A MDP problem with constraints to be satisfied

Constrained MDP

- A MDP problem with constraints to be satisfied

- In this work, we consider a data-driven setting.

Constrained MDP

- A MDP problem with constraints to be satisfied
- In this work, we consider a data-driven setting.
  - Model parameters are unknown and need to be learned from the data.

Constrained MDP

- A MDP problem with constraints to be satisfied
- In this work, we consider a data-driven setting.
  - Model parameters are unknown and need to be learned from the data.
  - An approach for multi-objective or safe reinforcement learning.
- Enjoys a very wide applications.
  - Prophet inequality with Markovian arrival (Jia et al. (2023)).

Constrained MDP

- A MDP problem with constraints to be satisfied

- In this work, we consider a data-driven setting.
  - Model parameters are unknown and need to be learned from the data.
  - An approach for multi-objective or safe reinforcement learning.

- Enjoys a very wide applications.
  - Prophet inequality with Markovian arrival (Jia et al. (2023)).
  - Network revenue management problem with Markovian arrival (Jiang (2023) and Li et al. (2023)).

Constrained MDP

- A MDP problem with constraints to be satisfied
- In this work, we consider a data-driven setting.
  - Model parameters are unknown and need to be learned from the data.
  - An approach for multi-objective or safe reinforcement learning.
- Enjoys a very wide applications.
  - Prophet inequality with Markovian arrival (Jia et al. (2023)).
  - Network revenue management problem with Markovian arrival (Jiang (2023) and Li et al. (2023)).
  - Markovian modulated demand process in inventory literature (e.g. Song and Zipkin (1993)).

Constrained MDP

- A MDP problem with constraints to be satisfied
- In this work, we consider a data-driven setting.
  - Model parameters are unknown and need to be learned from the data.
  - An approach for multi-objective or safe reinforcement learning.
- Enjoys a very wide applications.
  - Prophet inequality with Markovian arrival (Jia et al. (2023)).
  - Network revenue management problem with Markovian arrival (Jiang (2023) and Li et al. (2023)).
  - Markovian modulated demand process in inventory literature (e.g. Song and Zipkin (1993)).
  - Other applications in autonomous driving, robotics, financial management, etc.

Problem Formulation

- The tabular setting: a finite set of states $\mathcal{S}$ and a finite state of actions $\mathcal{A}$.

Problem Formulation

- The tabular setting: a finite set of states $\mathcal{S}$ and a finite state of actions $\mathcal{A}$.
- A transition kernel $P : (a, s) \rightarrow s'$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.

Problem Formulation

- The tabular setting: a finite set of states $\mathcal{S}$ and a finite state of actions $\mathcal{A}$.
- A transition kernel $P : (a, s) \to s'$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.
- A reward function $r : (s, a) \to [0, 1]$, allowed to be stochastic.

Problem Formulation

- The tabular setting: a finite set of states $\mathcal{S}$ and a finite state of actions $\mathcal{A}$.
- A transition kernel $P : (a, s) \rightarrow s'$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.
- A reward function $r : (s, a) \rightarrow [0, 1]$, allowed to be stochastic.
- K resource consumption functions, $c_k : (s, a) \rightarrow [0, 1]$, allowed to be stochastic for each $k \in [K]$.

Problem Formulation

- The tabular setting: a finite set of states $\mathcal{S}$ and a finite state of actions $\mathcal{A}$.
- A transition kernel $P : (a, s) \to s'$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.
- A reward function $r : (s, a) \to [0, 1]$, allowed to be stochastic.
- K resource consumption functions, $c_k : (s, a) \to [0, 1]$, allowed to be stochastic for each $k \in [K]$.
- A discount factor $\gamma \in (0, 1)$.

Problem Formulation

- The tabular setting: a finite set of states $\mathcal{S}$ and a finite state of actions $\mathcal{A}$.
- A transition kernel $P : (a, s) \to s'$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.
- A reward function $r : (s, a) \to [0, 1]$, allowed to be stochastic.
- K resource consumption functions, $c_k : (s, a) \to [0, 1]$, allowed to be stochastic for each $k \in [K]$.
- A discount factor $\gamma \in (0, 1)$.
- Goal: find a Markovian policy to maximize

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[r(a_t^\pi, s_t^\pi)]$$

subject to the resource constraints

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[c_k(a_t^\pi, s_t^\pi)] \le \alpha_k, \forall k \in [K].$$

Problem Formulation

- The tabular setting: a finite set of states $\mathcal{S}$ and a finite state of actions $\mathcal{A}$.
- A transition kernel $P : (a, s) \to s'$ for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.
- A reward function $r : (s, a) \to [0, 1]$, allowed to be stochastic.
- K resource consumption functions, $c_k : (s, a) \to [0, 1]$, allowed to be stochastic for each $k \in [K]$.
- A discount factor $\gamma \in (0, 1)$.
- Goal: find a Markovian policy to maximize

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[r(a_t^\pi, s_t^\pi)]$$

subject to the resource constraints

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[c_k(a_t^\pi, s_t^\pi)] \leq \alpha_k, \forall k \in [K].$$

- Unknown parameters: the transition kernel P, reward function r, and the cost function c_k for each $k \in [K]$.

Performance Measure

- Generative model: for each (s, a), we can obtain a sample of the state transition, reward, and costs, following true distributions.

Performance Measure

- Generative model: for each (s, a), we can obtain a sample of the state transition, reward, and costs, following true distributions.
- Denote by π^* the optimal policy and OPT the optimal value.

Performance Measure

- Generative model: for each (s, a), we can obtain a sample of the state transition, reward, and costs, following true distributions.
- Denote by π^* the optimal policy and OPT the optimal value.
- Sample complexity: for an arbitrary $\epsilon > 0$, how many samples we need in order to construct a policy π such that

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[r(a_t^\pi, s_t^\pi)] \geq \text{OPT} - \epsilon$$

and

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[c_k(a_t^\pi, s_t^\pi)] \leq \alpha_k + \epsilon, \forall k \in [K].$$

Performance Measure

- Generative model: for each (s, a), we can obtain a sample of the state transition, reward, and costs, following true distributions.
- Denote by π^* the optimal policy and OPT the optimal value.
- Sample complexity: for an arbitrary $\epsilon > 0$, how many samples we need in order to construct a policy π such that

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[r(a_t^\pi, s_t^\pi)] \geq \text{OPT} - \epsilon$$

and

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[c_k(a_t^\pi, s_t^\pi)] \leq \alpha_k + \epsilon, \forall k \in [K].$$

- Sample complexity for constrained MDP.

Performance Measure

- Generative model: for each (s, a), we can obtain a sample of the state transition, reward, and costs, following true distributions.

- Denote by π^* the optimal policy and OPT the optimal value.

- Sample complexity: for an arbitrary $\epsilon > 0$, how many samples we need in order to construct a policy π such that

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[r(a_t^\pi, s_t^\pi)] \geq \mathrm{OPT} - \epsilon$$

  and

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[c_k(a_t^\pi, s_t^\pi)] \leq \alpha_k + \epsilon, \forall k \in [K].$$

- Sample complexity for constrained MDP.
  - the worst-case $\tilde{O}(1/\epsilon^2)$ sample complexity known (e.g. Efroni et al. (2020)).

Performance Measure

- Generative model: for each (s, a), we can obtain a sample of the state transition, reward, and costs, following true distributions.
- Denote by π^* the optimal policy and OPT the optimal value.
- Sample complexity: for an arbitrary $\epsilon > 0$, how many samples we need in order to construct a policy π such that

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[r(a_t^{\pi}, s_t^{\pi})] \geq \text{OPT} - \epsilon$$

and

$$\sum_{t=1}^{\infty} \gamma^t \cdot \mathbb{E}[c_k(a_t^{\pi}, s_t^{\pi})] \leq \alpha_k + \epsilon, \forall k \in [K].$$

- Sample complexity for constrained MDP.
  - the worst-case $\tilde{O}(1/\epsilon^2)$ sample complexity known (e.g. Efroni et al. (2020)).
  - Whether we can achieve instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity?

Our Main Results

- We are the first to achieve the instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity with a new algorithm.

Our Main Results

- We are the first to achieve the instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity with a new algorithm.
  - The $\tilde{O}(\cdot)$ term hides a instance-dependent gap Δ.

Our Main Results

- We are the first to achieve the instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity with a new algorithm.
  - The $\tilde{O}(\cdot)$ term hides a instance-dependent gap Δ.
  - All our approach can be extended to finite horizon episodic setting, online learning setting, and offline learning setting.

Our Main Results

- We are the first to achieve the instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity with a new algorithm.
  - The $\tilde{O}(\cdot)$ term hides a instance-dependent gap Δ.
  - All our approach can be extended to finite horizon episodic setting, online learning setting, and offline learning setting.
- Contribution 1: characterize of Δ via the *corner points* of a feasible region.

Our Main Results

- We are the first to achieve the instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity with a new algorithm.
  - The $\tilde{O}(\cdot)$ term hides a instance-dependent gap Δ.
  - All our approach can be extended to finite horizon episodic setting, online learning setting, and offline learning setting.

- Contribution 1: characterize of Δ via the *corner points* of a feasible region.
  - the first characterization of instance hardness for CMDP problems.

Our Main Results

- We are the first to achieve the instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity with a new algorithm.
  - The $\tilde{O}(\cdot)$ term hides a instance-dependent gap Δ.
  - All our approach can be extended to finite horizon episodic setting, online learning setting, and offline learning setting.

- Contribution 1: characterize of Δ via the *corner points* of a feasible region.
  - the first characterization of instance hardness for CMDP problems.

- Contribution 2: a resolving method for solving CMDP problems with instance optimality.

Our Main Results

- We are the first to achieve the instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity with a new algorithm.
  - The $\tilde{O}(\cdot)$ term hides a instance-dependent gap Δ.
  - All our approach can be extended to finite horizon episodic setting, online learning setting, and offline learning setting.

- Contribution 1: characterize of Δ via the *corner points* of a feasible region.
  - the first characterization of instance hardness for CMDP problems.

- Contribution 2: a resolving method for solving CMDP problems with instance optimality.
  - Introduce the online LP framework and borrow the resolving algorithmic idea.

Our Main Results

- We are the first to achieve the instance-dependent $\tilde{O}(1/\epsilon)$ sample complexity with a new algorithm.
  - The $\tilde{O}(\cdot)$ term hides a instance-dependent gap Δ.
  - All our approach can be extended to finite horizon episodic setting, online learning setting, and offline learning setting.
- Contribution 1: characterize of Δ via the *corner points* of a feasible region.
  - the first characterization of instance hardness for CMDP problems.
- Contribution 2: a resolving method for solving CMDP problems with instance optimality.
  - Introduce the online LP framework and borrow the resolving algorithmic idea.
  - Our resolving method relaxes the non-degeneracy assumption.

LP Reformulation

- The *occupancy measure*: the total expected discounted time spent on a state-action pair, under a policy (Altman 1999).

LP Reformulation

- The *occupancy measure*: the total expected discounted time spent on a state-action pair, under a policy (Altman 1999).
- A LP formulation of the optimal policy.

$$
\begin{aligned}
V^* = \quad & \max \sum_{(s,a)} r(s,a) \cdot q(s,a) \\
\text{s.t.} & \sum_{(s,a)} c_k(s,a) \cdot q(s,a) \le \alpha_k, \forall k \in [K] \\
& \sum_{(s,a)} q(s,a) \cdot (1_{s=s'} - \gamma \cdot P(s'|s,a)) = (1-\gamma) \cdot \mu(s'), \forall s' \in \mathcal{S} \\
& q(s,a) \ge 0, \forall s \in \mathcal{S}, a \in \mathcal{A}
\end{aligned}
$$

$q(s,a)$: the total expected discounted time spent on (s,a).

LP Reformulation

- The *occupancy measure*: the total expected discounted time spent on a state-action pair, under a policy (Altman 1999).
- A LP formulation of the optimal policy.

$$
\begin{aligned}
V^* = \quad & \max \sum_{(s,a)} r(s,a) \cdot q(s,a) \\
\text{s.t.} & \sum_{(s,a)} c_k(s,a) \cdot q(s,a) \le \alpha_k, \forall k \in [K] \\
& \sum_{(s,a)} q(s,a) \cdot (1_{s=s'} - \gamma \cdot P(s'|s,a)) = (1-\gamma) \cdot \mu(s'), \forall s' \in \mathcal{S} \\
& q(s,a) \ge 0, \forall s \in \mathcal{S}, a \in \mathcal{A}
\end{aligned}
$$

$q(s,a)$: the total expected discounted time spent on (s,a).

- However, the LP parameters unknown hence cannot be directly solved.

Characterization via Optimal Basis

- The feasible region for the policy is a *polytope*.

Characterization via Optimal Basis

- The feasible region for the policy is a *polytope*.
- Feasible solution is "continuous" over the feasible region.
  - Hence no positive gap between the optimal solution and the sub-optimal one.

Characterization via Optimal Basis

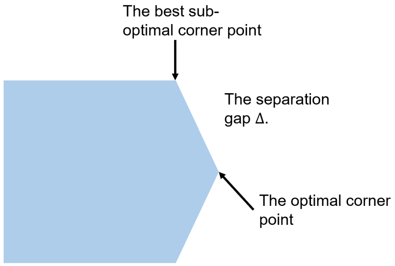
- The feasible region for the policy is a *polytope*.
- Feasible solution is "continuous" over the feasible region.
  - Hence no positive gap between the optimal solution and the sub-optimal one.
- There always exists one *corner point* to be optimal (basic solution).

Characterization via Optimal Basis

- The feasible region for the policy is a *polytope*.
- Feasible solution is "continuous" over the feasible region.
  - Hence no positive gap between the optimal solution and the sub-optimal one.
- There always exists one *corner point* to be optimal (basic solution).
  - If restrict to corner point solutions, then there exists a positive gap between the optimal one and the sub-optimal one.

Characterization via Optimal Basis

- The feasible region for the policy is a *polytope*.
- Feasible solution is "continuous" over the feasible region.
  - Hence no positive gap between the optimal solution and the sub-optimal one.
- There always exists one *corner point* to be optimal (basic solution).
  - If restrict to corner point solutions, then there exists a positive gap between the optimal one and the sub-optimal one.



Finding the Optimal Basis

- Restricting to corner points requires us to characterize the LP basis.

Finding the Optimal Basis

- Restricting to corner points requires us to characterize the LP basis.
  - $I \subset \mathcal{S} \times \mathcal{A}$: the index set of basic variables (the optimal actions to take for each state).

Finding the Optimal Basis

- Restricting to corner points requires us to characterize the LP basis.
  - $I \subset \mathcal{S} \times \mathcal{A}$: the index set of basic variables (the optimal actions to take for each state).
  - $J \subset [K]$: the set of constraints being binding.

Finding the Optimal Basis

- Restricting to corner points requires us to characterize the LP basis.
  - $I \subset \mathcal{S} \times \mathcal{A}$: the index set of basic variables (the optimal actions to take for each state).
  - $J \subset [K]$: the set of constraints being binding.

- General idea: *lexicographically* restrict the variables to zero to check whether the optimal LP value changes.

Finding the Optimal Basis

- Restricting to corner points requires us to characterize the LP basis.
  - $I \subset \mathcal{S} \times \mathcal{A}$: the index set of basic variables (the optimal actions to take for each state).
  - $J \subset [K]$: the set of constraints being binding.
- General idea: *lexicographically* restrict the variables to zero to check whether the optimal LP value changes.
  - For the primal LP: obtain the set of basic variables.

Finding the Optimal Basis

- Restricting to corner points requires us to characterize the LP basis.
  - $I \subset \mathcal{S} \times \mathcal{A}$: the index set of basic variables (the optimal actions to take for each state).
  - $J \subset [K]$: the set of constraints being binding.

- General idea: *lexicographically* restrict the variables to zero to check whether the optimal LP value changes.
  - For the primal LP: obtain the set of basic variables.
  - For the dual LP: obtain the set of binding constraints.

Finding the Optimal Basis

- Restricting to corner points requires us to characterize the LP basis.
  - $I \subset \mathcal{S} \times \mathcal{A}$: the index set of basic variables (the optimal actions to take for each state).
  - $J \subset [K]$: the set of constraints being binding.

- General idea: *lexicographically* restrict the variables to zero to check whether the optimal LP value changes.
  - For the primal LP: obtain the set of basic variables.
  - For the dual LP: obtain the set of binding constraints.

Finding the Optimal Basis

- Restricting to corner points requires us to characterize the LP basis.
  - $I \subset \mathcal{S} \times \mathcal{A}$: the index set of basic variables (the optimal actions to take for each state).
  - $J \subset [K]$: the set of constraints being binding.

- General idea: *lexicographically* restrict the variables to zero to check whether the optimal LP value changes.
  - For the primal LP: obtain the set of basic variables.
  - For the dual LP: obtain the set of binding constraints.

Theorem

When the sample size $n \geq \Omega(\frac{1}{\Delta} \cdot \log(1/\epsilon))$, we can identify one optimal I^* and J^* with probability at least $1 - \epsilon$.

Finding the Optimal Distributions

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.
  - update the remaining resources.

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \dots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.
  - update the remaining resources.
- A logarithmic regret ($O(\log T)$) can be obtained.

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.
  - update the remaining resources.
- A logarithmic regret $(O(\log T))$ can be obtained.
  - A crucial step in previous analysis is to stabilize the optimal basis!

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.
  - update the remaining resources.
- A logarithmic regret ($O(\log T)$) can be obtained.
  - A crucial step in previous analysis is to stabilize the optimal basis!
  - Non-degeneracy assumption: the underlying LP has a unique optimal basis.

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.
  - update the remaining resources.
- A logarithmic regret ($O(\log T)$) can be obtained.
  - A crucial step in previous analysis is to stabilize the optimal basis!
  - Non-degeneracy assumption: the underlying LP has a unique optimal basis.
- Our innovation: we resolve the LP while sticking to the optimal basis I^* and J^* that we have identified.

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \dots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.
  - update the remaining resources.
- A logarithmic regret ($O(\log T)$) can be obtained.
  - A crucial step in previous analysis is to stabilize the optimal basis!
  - Non-degeneracy assumption: the underlying LP has a unique optimal basis.
- Our innovation: we resolve the LP while sticking to the optimal basis I^* and J^* that we have identified.
  - We resolve a set of linear equations with only basic variables and binding constraints involved.

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.
  - update the remaining resources.
- A logarithmic regret ($O(\log T)$) can be obtained.
  - A crucial step in previous analysis is to stabilize the optimal basis!
  - Non-degeneracy assumption: the underlying LP has a unique optimal basis.
- Our innovation: we resolve the LP while sticking to the optimal basis I^* and J^* that we have identified.
  - We resolve a set of linear equations with only basic variables and binding constraints involved.

Finding the Optimal Distributions

- We adopt the resolving algorithm from the online LP literature (e.g. Agrawal et al. (2014), Kesselheim et al. (2014) and Li and Ye (2022)).
- At each iteration $t = 1, \ldots, T$,
  - construct an empirical LP V^* using available samples to obtain $\boldsymbol{q}^t$.
  - use the new sample and $\boldsymbol{q}^t$ to compute resource consumption.
  - update the remaining resources.
- A logarithmic regret ($O(\log T)$) can be obtained.
  - A crucial step in previous analysis is to stabilize the optimal basis!
  - Non-degeneracy assumption: the underlying LP has a unique optimal basis.
- Our innovation: we resolve the LP while sticking to the optimal basis I^* and J^* that we have identified.
  - We resolve a set of linear equations with only basic variables and binding constraints involved.

Theorem

Our algorithm enjoys a sample complexity of $\tilde{O}(\frac{1}{\Delta} \cdot \frac{1}{\epsilon})$.

Numerical Experiments

- We set $|\mathcal{S}| = |\mathcal{A}| = 10$ and $\gamma = 0.7$.
- We randomly generate the transition kernel P, and reward and cost functions, r and c_k.

Numerical Experiments

- We set $|\mathcal{S}| = |\mathcal{A}| = 10$ and $\gamma = 0.7$.
- We randomly generate the transition kernel P, and reward and cost functions, r and c_k.
- We consider the error term

$$\text{Err}(N) = \|\boldsymbol{q}^N - \boldsymbol{q}^*\|_1 / \|\boldsymbol{q}^*\|_1$$

where $\boldsymbol{q}^N$ denotes the occupancy measure computed by our algorithm with N samples.

Numerical Experiments

- We set $|\mathcal{S}| = |\mathcal{A}| = 10$ and $\gamma = 0.7$.
- We randomly generate the transition kernel P, and reward and cost functions, r and c_k.
- We consider the error term

$$\text{Err}(N) = \|\boldsymbol{q}^N - \boldsymbol{q}^*\|_1 / \|\boldsymbol{q}^*\|_1$$

where $\boldsymbol{q}^N$ denotes the occupancy measure computed by our algorithm with N samples.