逐次意思決定における諸問題設定と問題に関する事前知識が性能保証に及ぼす影響について

On Various Problem Settings of Sequential Decision Making and How Prior Knowledge of Problems Affects Theoretical Guarantees

小津野 将 [*1] Tadashi Kozuno　北村 俊徳 [*2] Toshinori Kitamura　市原 有生希 [*3] Yuki Ichihara　萩原 誠 [*4] Makoto Hagiwara

[*1]オムロンサイニックエックス Omron Sinic X　[*2]東京大学 Univ. of Tokyo　[*3]NAIST, ATR NAIST, ATR　[*4]pluszero pluszero

Recently, various problem settings of sequential decision making are proposed, and various types of performance guarantees are given. Examples are non-stationary MDPs and constrained MDPs. In this paper, we summarize and report recent development in performance guarantees for those settings, and we explain what kind of prior knowledge is useful for a better performance guarantee. Then, finally, we discuss open problems and interesting future study directions in the theory of sequential decision making.

1 序論

近年，さまざまな逐次意思決定の問題設定が考えられ，それらに対するさまざまな性能保証が示されている．非定常 MDP や制約付き MDP などがその例となる．本論文では，諸設定に対する性能保証の最近の発展をまとめ，どういった事前知識（問題のパラメータ）が性能向上に有益となるかを説明する．そして最後に，現在未解決の問題と将来の逐次意思決定理論の方向に関し議論する．[*1]

記法． 任意の有限集合 $\mathbf{S}$ に対し，$\Delta(\mathbf{S})$ を $\mathbf{S}$ 上の確率分布全体の集合とする．また，$[0,1]^{\mathbf{S}}$ や $\mathbb{R}^{\mathbf{S}}$ は $\mathbf{S}$ から $[0,1]$ または $\mathbb{R}$ への写像全体を表す．本論文では問題に関係ない定数などを無視する．そのための表記として $\mathcal{O}$ や Ω という表記をするが，$\mathcal{O}(x)$ はサンプル複雑性やリグレットの上限が定数を無視すると x となることを意味する．一方，$\Omega(x)$ は下限がそうなることを意味する．また，$\widetilde{\mathcal{O}}$ と $\widetilde{\Omega}$ は問題に関係した量の内，対数の多項式（例えば $(\ln x)^2$ 等）となる項を無視する．ただし，上限と下限は数学的意味の上限と下限ではなく，単に upper bound と lower bound の直訳である．

2 マルコフ決定過程（MDP）

逐次意思決定の諸問題設定はマルコフ決定過程（MDP）を基にしていることが多い．MDP には有限時間 MDP と無限時間 MDP が存在するが，有限時間 MDP の方が多くの理論結果が得られているため，本論文では有限時間 MDP の結果のみをまとめ，有限時間 MDP を単に MDP と呼ぶ．

MDP は $(\mathbf{S}, \mathbf{A}, \{r_t\}_{h=1}^H, \{P_t\}_{h=1}^H, \mu, H)$ の五つの要素で定義される．ここで，$\mathbf{S}$ は要素数が S の有限状態集合，$\mathbf{A}$ は要素数が A の有限状態集合，$r_h \in [0,1]^{\mathbf{S}\times\mathbf{A}}$ は報酬関数，$P_h \in \Delta(\mathbf{S})^{\mathbf{S}\times\mathbf{A}}$ は遷移確率関数，$\mu \in \Delta(\mathbf{S})$ は初期状態分布，$H \in \mathbb{N}$ は時間長となる．MDP のうち，r_h と P_h が時間に依存しないものを均一 MDP と呼び，その他のものを不均一 MDP と呼ぶ．

これらの要素は次のように用いられる．時刻 $h\,(\notin \{1, H\})$ において，エージェントは S_h を観測し，それに基づいて行動 A_h を決定，実行する．その結果，報酬 $r_h(S_h, A_h)$ がエージェントへと与えられ，次の状態 S_{h+1} が $P_h(S_h, A_h)$ からサンプルされる．時刻が 1 のときは S_1 が μ よりサンプルされ，時刻が H のときは，$r_H(S_H, A_H)$ がエージェントへと与えられた後に，時刻が 1 へとリセットされ，あらたに上記のことを繰り返す．時刻 1 から時刻 H までの一連の MDP・エージェントの相互作用をエピソードと呼ぶ．[*2]

エージェントが行動を決定するルールを方策と呼び，$\pi := (\pi_h)_{h=1}^H$ で表すことが多い．ただし，$\pi_h \in \Delta(\mathbf{A})^{\mathbf{S}}$．方策 π に対し，行動価値関数 $(Q_h^\pi)_{h=1}^H$ と状態価値関数 $(V_h^\pi)_{h=1}^H$ を以下のように定める．時刻 H に対しては $Q_H^\pi = r_H$ で，その他の時刻に関しては，$Q_h^\pi(s,a) := r_h(s,a) + \sum_{s'\in\mathbf{S}} P_h(s'|s,a) V_{h+1}^\pi(s')$．ただし，$V_h^\pi(s) := \sum_{a\in\mathbf{A}} \pi_h(a|s) Q_h^\pi(s,a)$．また期待リターンを $\rho^\pi = \sum_{s\in\mathbf{S}} \mu(s) V_1^\pi(s)$ と定める．最適方策の価値関数および期待リターンを Q_h^*，V_h^* および ρ^* と記す．

3 諸問題設定および目的関数

通常の設定における学習アルゴリズムの性能指標としては，以下のものが考えられることが多い [4]．

定義 1 (無後悔学習)**.** アルゴリズムが各エピソード k の初めに方策 π_k を出力し，それに従うこととする．また，ある正の定数 $\delta \in (0,1)$ を固定する．このとき（疑）リグレットを $R(K) := \sum_{k=1}^K (\rho^* - \rho^{\pi_k})$ と定義し，アルゴリズムが任意の *MDP* に対して以下の不等式を満たす多項式 F_{HPR} を持つ場合，それを性能の指標とする：

$$\mathbb{P}\left(R(K) \le F_{\mathrm{HPR}}\left(S, A, H, K, \ln\frac{1}{\delta}\right)\right) > 1-\delta\,.$$

定義 2 (PAC 学習)**.** アルゴリズムが各エピソード k の初めに方策 π_k を出力するが，必ずしもそれに従う必要はないとする．また，ある正の定数 $\delta \in (0,1)$ と $\varepsilon \in (0, H)$ を固定する．このとき最適性ギャップを $\Delta(k) := \rho^* - \rho^{\pi_k}$ で定義し，失敗回数を $N_\varepsilon := \sum_{k=1}^\infty \mathbb{I}\{\Delta_k > \varepsilon\}$ とする．このとき，アルゴリズムが任意の *MDP* に対して以下の不等式を満たす多項式 F_{PAC}

連絡先: 小津野 将, オムロンサイニックエックス株式会社, 東京都文京区本郷五丁目 24 番 5 号 ナガセ本郷ビル 3F, tadashi.kozuno@sinicx.com

*1 詳細版はつぎの URL より利用可能：temp

*2 理論解析においては確率的報酬を考えないことが多い．学習においては価値推定の分散が問題となるが，確率的報酬に起因する分散よりも確率的状態遷移に起因する分散の方が大きく，報酬が確率的であっても問題の難しさは本質的に変わらないためだ．

表 1: さまざまな MDP の問題設定．報酬関数と状態遷移確立の空欄は，単一の固定されたものを使うことを意味する． [Tadashi: 目的関数については本文に書いて、それを ref した方がいい]. [Toshinori: 性能保証についても同じ表にまとめますか？][Tadashi: 入りそうならですかね．入らなさそうな気がします．そもそも表を横向きにして一ページ丸まる使ってもいいですが...]

設定名称	報酬関数 r	状態遷移確立 P	目的関数	文献
Adversarial MDP	エピソード依存		リグレット	?
Constrained MDP			制約付きリグレット or 制約付き準最適性	?
Horizon-Free 学習			リグレット	?
Nonstationary MDP	エピソード依存	エピソード依存	リグレット	?
Reward-Free 学習	全報酬関数		リグレット or exploitation gap	?
Robust MDP		P の確率分布	準最適性	?

を持つ場合，それを性能の指標とする：

$$\mathbb{P}\left(N_\varepsilon \le F_{\mathrm{PAC}}\left(S, A, H, \frac{1}{\varepsilon}, \ln\frac{1}{\delta}\right)\right) > 1-\delta\,.$$

定義 3 (一様 PAC 学習)．アルゴリズムが各エピソード k の初めに方策 π_k を出力し，それに従うこととする．また，ある正の定数 $\delta \in (0,1)$ を固定し，失敗回数 N_ε を定義 2と同様に定義する．このとき，アルゴリズムが任意の MDP に対して以下の不等式を満たす多項式 F_{UPAC} を持つ場合，それを性能の指標とする：

$$\mathbb{P}\left(\forall\varepsilon : N_\varepsilon \le F_{\mathrm{UPAC}}\left(S, A, H, \frac{1}{\varepsilon}, \ln\frac{1}{\delta}\right)\right) > 1-\delta\,.$$

これらの違いについて説明を加える．もとの定義より $R(K) \le KH$ であるため，F_{HPR} は必ず存在するが，そのような保証は無意味である．最悪ケースを考える場合，情報理論的に達成可能な F_{HPR} の下限については $\widetilde{\Omega}(H\sqrt{SAK})$ であることが知られており [11]，実際に F_{HPR} が $\widetilde{\mathcal{O}}(H\sqrt{SAK})$ であるアルゴリズムも存在するため [1]，MDP から派生した問題設定においては $\widetilde{\mathcal{O}}(H\sqrt{SAK})$（またはそれに近いもの）を達成することが大抵の目標となる．一方，F_{PAC} と F_{PAC} が存在するかは不明である．

下限についても $\sqrt{K}$ のオーダーであることが知られているため [11],

4 無報酬強化学習

4.1 準備

[Toshinori: TODO: とりあえず全部書いたら表記や用語を 3 章の内容をベースに修正しよう．citep, citet 使いたい.]

この章では, 2章から派生させた有限ホライズンかつステップに非定常なマルコフ決定過程を扱う．具体的には, $H \in \mathbb{N}$ をホライズンの長さ, $h \in [H]$ ステップ目の報酬関数と遷移確率関数をそれぞれ r_h と P_h として, $M_{\mathbf{r}} := \{\mathbf{S}, \mathbf{A}, \gamma, \mathbf{r}, \mathbf{P}, \mu\}$ で定義されたマルコフ決定過程を考える．ここで, $\mathbf{r} := \{r_h\}_{h\in[H]}$, $\mathbf{P} := \{P_h\}_{h\in[H]}$ とした．また, ステップ h の方策を $\pi_h \in \mathbb{R}^{S\times A}$ として, 方策を $\pi := \{\pi_h\}_{h\in[H]}$ で表す．$V_h^\pi(s;\mathbf{r}) := \mathbb{E}_\pi\left[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) \middle| s_h = s\right]$ を $M_{\mathbf{r}}$ における π の状態価値関数とする．$M_{\mathbf{r}}$ の最適方策 $\pi^*_{\mathbf{r}}$ は任意の π に対して $V_h^\pi(s;\mathbf{r}) \le V_h^{\pi^*_{\mathbf{r}}}(s;\mathbf{r})$ を $\forall(s,h) \in \mathbf{S}\times[H]$ で満たす．

4.2 概要

無報酬強化学習 (Reward Free Reinforcement Learning, [6]) では, 予め報酬情報を使わずに環境を十分に探索してデータを収集し, 任意の報酬関数における最適な方策を新たな探索をせずに近似する．正確には, 無報酬強化学習では次の流れに従ってアルゴリズムが進行する．

1. 報酬の情報を使わずに, 環境を $h=1$ から $h=H$ ステップまで探索する．これを K エピソード繰り返し, 訪問した状態と行動のデータセット $\mathbf{D} = \{s_h^{(k)}, a_h^{(k)}\}_{(k,h)\in[K]\times[H]}$ を構築する．
2. $\mathbf{D}$ を使い, 各 h ステップ目の遷移確率関数を近似する．近似した遷移確率関数の組を $\widehat{\mathbf{P}} := \{\widehat{P}_h\}_{h\in[H]}$ とする．
3. 任意の報酬関数の組 $\mathbf{r}$ に対して, 近似されたマルコフ決定過程 $(\widehat{M}_{\mathrm{RF}} = \{\mathbf{S}, \mathbf{A}, \gamma, \mathbf{r}, \widehat{\mathbf{P}}, \mu\})$ における最適方策 $\widehat{\pi}^*_{\mathbf{r}}$ を計算する．

無報酬強化学習の性能は, 次で定義される (ε,δ)-PAC 学習を達成するために必要なエピソード数 K で解析されることが多い．

定義 4. アルゴリズムが出力する $\mathbf{D}$ によって計算された $\widehat{\pi}^*_r$ が次を満たすとき, そのアルゴリズムを無報酬強化学習における (ε,δ)-PAC アルゴリズムと呼ぶ．

$$\mathbb{P}\left(\text{任意の } \mathbf{r} \text{ で}, \left|\mathbb{E}_{s_1\sim\mu}\left[V_1^{\pi^*_{\mathbf{r}}}(s_1;\mathbf{r}) - V_1^{\widehat{\pi}^*_{\mathbf{r}}}(s_1;\mathbf{r})\right]\right| \le \varepsilon\right) \ge 1-\delta\,.$$

4.3 事前知識と性能保証

無報酬強化学習の問題設定は [6] によって定式化された．4.1と4.2に記載した問題設定では, (ε,δ)-PAC 学習を達成するために必要なエピソード数のオーダーの下界が $\widetilde{\Omega}\left(\frac{S^2AH^2}{\varepsilon^2}\right)$ であることが示されている [6]．一方で, 必要なエピソード数のオーダーの上界が $\widetilde{\mathcal{O}}\left(\frac{S^2AH^2}{\varepsilon^2}\right)$ であるアルゴリズムが [15] らによって提案されており, このアルゴリズムは上界と下界のオーダーが一致している点で最適オーダーなアルゴリズムである．

[6] が提案した問題設定では任意の報酬関数に対して

[Tadashi: 以下、stationary と non-stationary が time-homogeneous か time-inhomogeneous を意味するために使われています。一方、エピソード毎に r と P が少しずつ変わる設定を non-stationary MDP ともいうので、time-homogeneous か time-inhomogeneous に統一しましょう。][Toshinori: この辺の話をまとめる (オーダーはエピソード数のサンプル効率):]

- [6]: 基礎の論文．設定は time-inhomogeneous: P_h, r_h を考える。エピソードを K 回繰り返してデータセット $\mathcal{D} = \{s_h^{(k)}, a_h^{(k)}\}_{(k,h}} \in [K]\times[H]$ を集める．K の上界 (各

状態とステップの組 (s, h) に到達する確率を最大化するような方策を EULER で獲得するアルゴリズム．獲得した方策の集合からサンプルすることでデータセットを集める)：$\widetilde{\mathcal{O}}\left(\frac{S^2AH^5}{\varepsilon^2}+\frac{S^4AH^7}{\varepsilon}\right)$．K の下界：$\widetilde{\Omega}\left(\frac{S^2AH^2}{\varepsilon^2}\right)$．この下界はアルゴリズムが non-markov な方策を吐き出す場合でも、報酬と遷移確率が time-homogeneous な場合でも成立する．ただし, 報酬関数が無限個ある時の下界の話．報酬関数の数が限られている場合は不明．

- [8]: Jin よりも H だけまし。設定は time-inhomogeneous: Jin のものと同じ． K の上界 (time-inhomogeneous)：$\widetilde{\mathcal{O}}\left(\frac{S^2AH^4}{\varepsilon^2}+\frac{S^4AH^7}{\varepsilon}\right)$． UCRL の reward free バージョンっぽい． K の上界 (time-homogeneous)：$\widetilde{\mathcal{O}}\left(\frac{S^2AH^3}{\varepsilon^2}+\frac{S^4AH^7}{\varepsilon}\right)$．time-inhomogeneous よりも H まし．

- [10]: 設定は time-inhomogeneous: Jin のものと同じ．下界を $\widetilde{\Omega}\left(\frac{S^2AH^3}{\varepsilon^2}\right)$ と Jin のものよりも H 増やしている． non-stationary のせい，と言っているが，Jin の導出も time-inhomogeneous なので本当かな？上界：$\widetilde{\mathcal{O}}\left(\frac{S^2AH^3}{\varepsilon^2}+\frac{S^4AH^7}{\varepsilon}\right)$．UCRL のボーナス項を $1/\sqrt{n}$ から $1/n$ にしたら効率が上がったっぽい．

- [14]: 設定は time-inhomogeneous: ほぼ Jin のものと同じだが，確率的な報酬関数を考えている． 上界：$\widetilde{\mathcal{O}}\left(\frac{SAH^5\log(N)}{\varepsilon^2}\right)$．任意の報酬関数の代わりに，$N$ 個に限定された報酬関数を考えている．このとき，S に対する依存性が取れて $\log(N)$ に変わるっぽい．これは Jin の下界の導出で $N \geq \exp(S)$ の数の報酬関数を使うからみたい．実際，$N \propto \exp(S)$ なら Zhang の上界はほぼ Jin のものと一緒になる．Reward Free のときの N に依存する下界は不明．ただし、Task-agnostic の下界は $\widetilde{\mathcal{O}}\left(\frac{SAH^2\log(N)}{\varepsilon^2}\right)$．

- [15]: 遷移確率は time-homogeneous だが、報酬関数は inhomogeneous (time-inhomogeneous な遷移確率も提案されているので，そっちを使おう)．また，上の全ての設定では報酬関数が一様にバウンドされていたが、この論文では totally bounded な設定を考える．アルゴリズムを頑張って改善して Nearly Minimax を達成．上界は報酬の範囲が変わってるので丁寧に比較しないといけないよ．これはホライズンフリーになってるが、比較のために Totally bounded を外したよ．

- [13]: 遷移確率は time-homogeneous だが, 報酬関数は inhomogeneous．最適な行動とそれ以外の行動を見分けるための gap を利用することで効率よく解ける: $\mathrm{gap}_h(s,a) := V_h^*(x) - Q_h^*(s,a) \geq 0$ について、$0 < \rho \geq \mathrm{gap}_{\min}(r) \forall r \in \mathbb{R}^{SA}$．上界: $\widetilde{\mathcal{O}}\left(\frac{S^2AH^3}{\rho\varepsilon}+\frac{S^3AH^4}{\varepsilon}\right)$．gap-dependent にすることで、分母の ε^2 が ε になってる．直感的には gap に依存した十分な数の訪問回数に至ったら UCB bonus を打ち切ることで余計な探索をなくしてる．

4.4 Constrained MDP

- [5] ではエージェントに violation をしていいと仮定している (sublinear)．遷移カーネル，コスト 関数，制約は未知として行った．そして 4 つのアルゴリズムを提案．下のリグレットは全て上界.$\widetilde{\mathcal{O}_\mathcal{C}}$ はコスト関数について,$\widetilde{\mathcal{O}_\mathcal{V}}$ は violation に関して．$\widetilde{\mathcal{O}_\mathcal{C}}\left(H^3\sqrt{|S|^3|A|K}\right)$,$\widetilde{\mathcal{O}_\mathcal{V}}\left(H^3\sqrt{|S|^3|A|K}\right)$ non-sta

  1. OptCMDP:
  2. OptCMDP-bonus:
  3. OptDual-CMDP:
  4. OptPrimalDualCMDP:

表 2: 無報酬強化学習における事前知識と性能保証の比較．$\widetilde{\mathcal{O}}$ と $\widetilde{\Omega}$ の記法では対数係数を省略した．

	事前知識	エピソード効率	文献
下界	-	$\widetilde{\Omega}\left(\frac{S^2AH^2}{\varepsilon^2}\right)$	[6]
上界	-	$\widetilde{\mathcal{O}}\left(\frac{S^2AH^2}{\varepsilon^2}\right)$	[15]
	$\mathbf{r} \in \{\mathbf{r}_n\}_{n\in[N]}$	$\widetilde{\mathcal{O}}\left(\frac{SAH^5}{\varepsilon^2}\right)$	[14]
	$\rho \leq \mathrm{gap}_{\min}(\mathbf{r})$	$\widetilde{\mathcal{O}}\left(\frac{S^2AH^3}{\rho\varepsilon}+\frac{S^3AH^4}{\varepsilon}\right)$	[13]

- [12] ベースラインの方策仮定してないけど制約についての仮定がある．エージェントは制約と報酬は既知と仮定している．unichainMDP 考えて，報酬と制約の平均をだしている．報酬と制約の期待リグレットの上界は，$\widetilde{\mathcal{O}}\left(T^{\frac{2}{3}}\right)$．T は time-Horizon．sta

- [2] は baseline 方策考えている．そして，目的コスト関数を減らすことを目標で，そのリグレットは [9] より $\sqrt{|S|}$ 減っている．violation は 0．遷移確率とコストと制約に信頼区間を設けている．non-sta

- [9] はコスト関数，制約，遷移カーネル未知で，LP 問題の時は vioration は baseline 方策考えて 0 になっている．primal dual 問題の時は制約についての知識なしで violation の regret は $\widetilde{\mathcal{O}}(1)$, 報酬のリグレットはどちらも $\widetilde{\mathcal{O}}\left(\frac{H^3}{\tau-c^0}\sqrt{|S|^3|A|K}\right)$, non-sta

  baseline の方策を考えていないときはほとんどの論文で primal dual 問題を考えている．方策と制約のラグランジュ考えて同時に最適化．報酬のリグレットは他の論文と比べてあんまりよくなっていないけど，violation は小さくなっているから，リスク避けたい時には貢献していると言及．

- [7] は [2] と比較している論文．[2] よりも弱い仮定を使って，解いている．手法は primal-dual．遷移確率は未知．Bayesian reward objective regret **[Yuki: baysian reward regret がよくわかってない．．．]** は $\widetilde{\mathcal{O}}\left(|H|^{2.5}\sqrt{|S|^2|A|K}\right)$,vaiolation は $\widetilde{\mathcal{O}}(1)$.

- [3] は baseline 方策考えている．最適化に関する関数のリグレットは $\widetilde{\mathcal{O}}\left(S^4H^7AK\right)$, 制約に関するリグレットは 0.non-sta

割引率を含めた累積報酬の期待値を次のように書く．

$$V^\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)|s_0 = s, a_t \sim \pi(\cdot|s_t)].$$

上の $V^\pi(s)$ を初期分布 μ から期待値とったものを次のように定義する．

$$\nu^\pi = \mathbb{E}_{s\sim\mu}[V^\pi(s)].$$

定義 5. *CMDP* では次の最適化問題を解くことを目標とする．

$$\max \quad \nu^\pi \quad s.t. \quad \Gamma^\pi \geq c$$

$\Gamma^\pi = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)|s_0 = s, a_t \sim \pi(\cdot|s_t)]$ ，c は制約に関する閾値で $c(s, a)$ は何らかの制約に関する関数．

[**Toshinori:** TODO: 参考文献のフォーマットは終わり際にキレイにしても良さそう]

5 メモ書き

[**Toshinori:** 書く内容：下限と上限は書こう。今後の研究者が Limitation に気づきやすいように書きたい。事前知識が入ってる場合に Minimax になるのか、]

問題設定

- adversarial MDP（小津野）
- non-stationary MDP（小津野）
- constrained MDP（市原）
- robust MDP（小津野）
- robust MDP (distribution version)（小津野）
- zero-sum extensive form game（小津野）
- markov game（小津野）
- Markov potential game（小津野）
- Horizon-free（小津野）
- reward-free（北村）
- safe RL (あんまり詳しくない。いろいろと細かい何かがあったはず)
- 北村：Multi task RL (ややありそう)

[**Tadashi:** タイトルにある通り、逐次意思決定における諸問題設定と問題に関する事前知識が性能保証に及ぼす影響についての survey ですので、representation learning、regularized MDP、average reward とかはあんまり関係なさそうです。][**Yuki:** constrained mdp のベースラインとなる方策なども事前知識などにあたるのでしょうか][**Tadashi:** そうですね。それも一種の事前知識だと思います][**Tadashi:** とはいえ，出来るだけ環境に関する事前知識に制限しましょうか... ベースラインの方策も含めるなら，そもそも最適方策も事前知識やん？ってなっちゃうので...Constrained MDP の場合はベースライン方策がなければ strict な no violation 学習は不可能そう？なので．そのあたりも含め調べていきましょう][**Tadashi:** POMDP はそもそも解けない問題が多い（むしろ大半？）っぽいので除外しますか...]

性能保証

- problem-dependent (i.e., first order) and worst-case regret
- problem-dependent and worst-case sample complexity
- uniform-pac（これは problem-dependent はあるのか…？）

参考文献

[1] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax Regret Bounds for Reinforcement Learning. In *International Conference on Machine Learning*, 2017.

[2] Archana Bura, Aria HasanzadeZonuzy, Dileep Kalathil, Srinivas Shakkottai, and Jean-Francois Chamberland. Dope: Doubly optimistic and pessimistic exploration for safe reinforcement learning. In *Advances in Neural Information Processing Systems*.

[3] Archana Bura, Aria HasanzadeZonuzy, Dileep Kalathil, Srinivas Shakkottai, and Jean-Francois Chamberland. Safe exploration for constrained reinforcement learning with provable guarantees. *arXiv preprint arXiv:2112.00885*, 2021.

[4] Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

[5] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*, 2020.

[6] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020.

[7] Krishna C Kalagarla, Rahul Jain, and Pierluigi Nuzzo. Safe posterior sampling for constrained mdps with bounded constraint violation. *arXiv preprint arXiv:2301.11547*, 2023.

[8] Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In *Algorithmic Learning Theory*, pages 865–891. PMLR, 2021.

[9] Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained mdps. *Advances in Neural Information Processing Systems*, 34:17183–17193, 2021.

[10] Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. *arXiv preprint arXiv:2007.13442*, 2020.

[11] Ian Osband and Benjamin Van Roy. On Lower Bounds for Regret in Reinforcement Learning. *ArXiv*, abs/1608.02732, 2016.

[12] Rahul Singh, Abhishek Gupta, and Ness Shroff. Learning in constrained markov decision processes. *IEEE transactions on control of network systems*, 2022.

表 3: さまざまなゲームの問題設定．[**Tadashi:** これはどうまとめるかなぁ...]

設定名称	報酬関数 r	状態遷移確立 P	目的関数	文献
Zero-Sum IIG	エピソード依存		リグレット	?
Markov Game			制約付きリグレット or 制約付き準最適性	?
Markov Potential Game			リグレット	

[13] Jingfeng Wu, Vladimir Braverman, and Lin Yang. Gap-dependent unsupervised exploration for reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 4109–4131. PMLR, 2022.

[14] Xuezhou Zhang, Yuzhe Ma, and Adish Singla. Task-agnostic exploration in reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 11734–11743, 2020.

[15] Zihan Zhang, Simon S Du, and Xiangyang Ji. Nearly minimax optimal reward-free reinforcement learning. *arXiv preprint arXiv:2010.05901*, 2020.