Provably Efficient Reinforcement Learning for Large Aggregative Markov Games

Yuanchen Tang
Peking University
shtangyuanchen@stu.pku.edu.cn

Jingwu Tang
Carnegie Mellon University
jingwutang@cmu.edu

Zhiwei Steven Wu
Carnegie Mellon University
zstevenwu@cmu.edu

Abstract

Markov Games are widely studied in game theory and multi-agent reinforcement learning. It has been shown that, in general, the computational complexity of solving and learning the Nash Equilibrium (NE) is PPAD-complete. We study a general class of Markov games—Aggregative Markov Games (AMG)—where there is a vector of linear functions of players' joint actions, called an aggregator. Each player's transition is then a possibly non-linear function of the aggregator vector, and the dynamics are a function of the aggregator vector. We show that for any AMG with a large number of agents, it is possible to efficiently obtain an NE. We propose algorithms that can solve and learn an NE with sample and computational complexity polynomial in the number of agents n. Furthermore, we provide an algorithm that can select a near-optimal equilibrium under a linear loss function of agents' policies.

1 Introduction

An **aggregative game** is a general model for large-scale games in which players' actions collectively form an *aggregator* vector, consisting of linear functions of these actions. Each player's utility depends on this aggregator and their own action, potentially through a non-linear function. In congestion games, for example, the aggregator can represent congestion levels in specific areas or on particular roads. In large aggregative games, any single player's unilateral change in action has, at most, a limited effect on other players' utilities.

Aggregative games have been studied in the context of normal-form games (NFG). Cummings et al. [2015] presents an efficient algorithm for solving multi-dimensional aggregative NFG. We extend aggregative games to the Markov Games setting, with the additional assumption that the Markov Game transition function depends on the aggregator x.

Our contributions are three-fold:

1. We initiate the study of aggregative games within Markov Games and propose two algorithms to solve NE in large Aggregative Markov Games with both statistically and computational complexity polynomial in n.

2. Building on the proposed algorithms for solving NE in AMG, we present two learning algorithms to learn NE with polynomial computational and sample complexity.

3. We extend the solving and learning algorithms to select equilibria with respect to a loss function that is linear over the agents' policies.

Preprint. Under review.

2 Preliminaries

Markov Games. We use $MG(H, \mathcal{S}, \mathcal{A}, \mathcal{T}, \{r_i\}_{i=1}^n, \rho_0)$ to denote a *Markov Game* (MG) between m agents. Here, H is the horizon, $\mathcal{S}$ is the state space, and $\mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_m$ is the joint action space for all agents. We use $\mathcal{T}_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ to denote the transition function. Furthermore, the reward (utility) function for agent $i \in [n]$ is denoted by $r_{h,i} : \mathcal{S} \times \mathcal{A} \to [-1, 1]$. We use ρ_0 to denote the initial state distribution from which the initial state $s_0 \sim \rho_0$ is sampled. The policy of the i^{th} player is denoted as $\pi_i := \{\pi_{h,i} : S \to \Delta A_i\}_{h\in[H]}$. We denote the product policy of all the players as $\pi := \pi_1 \times \cdots \times \pi_M$, and denote the policy of all the players except the i^{th} player as π_{-i}. A trajectory $\xi \sim \pi = \{s_h, \vec{a}_h\}_{h=1,...,H}$ refers to a sequence of state-action pairs generated by starting from $s_0 \sim \rho_0$ and repeatedly sampling joint action $\vec{a}_h$ and next states s_{h+1} from π and $\mathcal{T}$ for $H-1$ time steps. We define $V_{h,i}^{\pi}(s)$ as the expected cumulative reward that will be received by the i^{th} player if starting at state s at step h and all players follow policy π, i.e. $V_{h,i}^{\pi}(s) = \mathbb{E}_{\xi\sim\pi}[\sum_{t=h}^{H} r_i(s_t, \vec{a}_t)|s_h = s]$. We define Q-value function of agent i as $Q_{h,i}^{\pi}(s, \vec{a}) = \mathbb{E}_{\xi\sim\pi}[\sum_{t=h}^{H} r_i(s_t, \vec{a}_t)|s_h = s, \vec{a}_h = \vec{a}]$. We define advantage of an agent i to be the difference between its Q-value on a selected action and the V-value on the state, i.e. $A_{h,i}^{\pi}(s, \vec{a}) = Q_{h,i}^{\pi}(s, \vec{a}) - V_{h,i}^{\pi}(s)$.

Nash Equilibrium. For any strategy π_{-i}, there also exists a best response of the i^{th} player, which is a policy $\mu^{\dagger}(\pi_{-i})$ satisfying

$$V_{h,i}^{\mu^{\dagger}(\pi_{-i}),\pi_{-i}}(s) = \sup_{\pi_i} V_{h,i}^{\pi_i,\pi_{-i}}(s) \quad \text{for any } (s, h) \in S \times [H].$$

We denote $V_{h,i}^{\dagger,\pi_{-i}} := V_{h,i}^{\mu^{\dagger}(\pi_{-i}),\pi_{-i}}$. The Q-functions of the best response can be defined similarly.

Our first objective is to find an approximate Nash equilibrium of Markov games.

Definition 2.1. *(ϵ-approximate Nash equilibrium in general-sum MGs). A product policy π is an ϵ-approximate Nash equilibrium if* $\max_{i\in[m]} \left(V_{1,i}^{\dagger,\pi_{-i}} - V_{1,i}^{\pi}\right)(s_1) \le \epsilon$.

Aggregative Markov Games. The aggregator vector $x^k(\vec{a})$ is defined as:

$$x_h^k(\vec{a}) = \delta \sum_{i=1}^{m} f_{h,i}^k(a_i)$$

where δ is a scaling factor, m is the number of agents, and $f_{h,i}^k(a_i)$ represents the contribution of agent i's action a_i to the k-th dimension of the aggregator. For Aggregative Markov Games, we assume that the reward function is a function of the state, the agent's own action and the aggregator. $r_{h,i} : \mathcal{S} \times \mathcal{A}_i \times X \to [-1, 1]$. We also assume that transition is a function of the state and the aggregator $\mathcal{T}_h : \mathcal{S} \times X \to \Delta(\mathcal{S})$. Normally, $x_{h,i} : \mathcal{S} \times \mathcal{A} \to X = [-W, W]^d$. The relation between reward, transition function with and without aggregator is: $r_{h,i}(s, a_i, x_h(s, \boldsymbol{a})) = r_{h,i}(s, \boldsymbol{a})$;$\mathbb{P}_h(s, x_h(s, \boldsymbol{a})) = \mathbb{P}_h(s, \boldsymbol{a})$

Assumption 2.1 (Bounded Influence)**.** *The influence of any single agent's action change on the aggregator vector is bounded. For any actions a_i and a_i',*

$$||x_h(a_i, a_{-i}) - x_h(a_i', a_{-i})||_\infty \le \delta$$

where δ is a constant. This assumption ensures that the impact of one agent's action change on the overall aggregator remains limited.

Assumption 2.2 (Lipschitz Continuity of Reward)**.** *The reward function $r_{h,i}(s, a_i, x_h)$ is Lipschitz continuous with respect to the aggregator x_h. Specifically, for any two aggregator values x_h' and x_h',*

$$|r_{h,i}(s, a_i, x_h) - r_{h,i}(s, a_i, x_h')| \le \gamma_r ||x_h - x_h'||_\infty$$

where γ_r is the Lipschitz constant for the reward function. This ensures that small changes in the aggregator lead to proportionally small changes in rewards.

Assumption 2.3 (Lipschitz Continuity of Transition Probability)**.** *The transition probability $\mathcal{T}_h(s, x_h)$ is Lipschitz continuous with respect to the aggregator x_h. For any two aggregator values x_h and x_h',*

$$|\mathcal{T}_h(s, x_h) - \mathcal{T}_h(s, x_h')|_{TVD} \le \gamma_{\mathcal{T}} ||x_h - x_h'||_\infty$$

where $\gamma_{\mathcal{T}}$ is the Lipschitz constant for the transition probability and $|\cdot|_{TVD}$ denotes the total variation distance. This assumption implies that changes in the aggregator vector only slightly affect the transition dynamics.

3 Warm-up: state-wise algorithm

In this section, we introduce the problem setting with a trivial algorithm and present its theoretical guarantee.

We consider Nash equilibrium (NE) being computed or learned in a dynamics-unknown environment. Our first algorithm is simply to compute Nash within each state with the reward functions, and combine all the result policy as the output. As it's highly local, we can still get an approximate pure Nash equilibrium, but no equilibrium selection is guaranteed.

Algorithm 1 State-wise

for s, h **do**
$\pi_h(\cdot|s) \leftarrow LP(\{r_{\cdot,h}(s,\cdot)(s,a_i,x_k)\}_{i,k,a_i})$

(Subroutine: solve a Linear Program 4.1.1 , scaled by γ_r, respectively for all x_k;
Pick the solution $\vec{p}$ with the largest objective value among all x_k;
Output a pure strategy sampled from the solution $\vec{p}$.)

Theorem 3.1. *For discretizing interval length α, algorithm 1 obtains an $(\zeta + 4\alpha + 2\delta + 2E)\gamma_r H + \delta\gamma_P \frac{H(H-1)}{2}$)-pure strategy Nash Equilibrium, where*

$$\zeta = \delta\sqrt{8n\log(2mn)}, E = O(\sqrt{n}\delta\,\mathrm{polylog}(d, 1/\beta))$$

4 Planning NE in AMG

We have already got the results for Sections 4 to 6 , and are still writing the draft for the paper.

In this section, we present our first main algorithm—Aggregative Backward Induction (Aggre-BI), and provide its theoretical guarantee.

4.1 Algorithm description

We describe our Nash-VI Algorithm 2:

- Performs backward induction, and computes a new (joint) policy π which is "greedily" computed from a Nash subroutine with respect to the value functions.

At a high-level, this strategy is standard in the majority of planning algorithms, and also underlies provably efficient planning algorithms such as CCE-Backward-Induction. However, CCE-Backward-Induction has two undesirable drawbacks: the computational complexity is dependent on $\Pi_i \mathcal{A}_i$, and it is limited to coarse correlated equilibrium (CCE), which is a weaker notion of NE that admits polynomial algorithms for normal-form games. However, our updating rule of backward induction releases the dependence on multi-agency, though doesn't avoid exponential dependence on aggregator dimension d.

4.1.1 Overview of techniques

Nash Subroutine. Our $Subroutine$ leverages the algorithm in Cummings et al. [2015] Appendix D on Q function, which efficiently computes an approximate pure Nash equilibrium for a normal-form aggregative game, with the input information of all the reward value on all (action, discretized aggregator) pairs, within time polynomial to $(W/\alpha)^d$

$$\max_{\vec{p}} \quad L(\vec{p}) + \sum_{s'} \mathbb{P}(s, \hat{x}_k, s') \cdot W_{h+1}(s')$$

$$\forall k \in [d], \quad \hat{x}_k - \alpha \leq \sum_{i=1}^{n} \sum_{j=1}^{m} \hat{f}_{ij}^k p_{ij} \leq \hat{x}_k + \alpha$$

$$\forall i \in [n], \quad \forall j \in \vec{BA}_i(\hat{x}_k), \quad 0 \leq p_{ij} \leq 1$$

$$\forall i \in [n], \quad \forall j \notin \vec{BA}_i(\hat{x}_k), \quad p_{ij} = 0$$

$$\forall i \in [n], \quad \sum_{j=1}^{m} p_{ij} = 1$$

Algorithm 2 Planning in AMG

for $h = H, \ldots, 1$ **do**
  for $s \in \mathcal{S}$ **do**
    for $i, a_i, x \in (\alpha\mathbb{Z} \cap [-W, W])^d$ **do**
      $Q_{i,h}(s, a_i, x), W_h(s) \leftarrow r_{i,h}(s, a_i, x) + \sum_{s'} P(s, x, s') V_{i,h+1}(s')$
    $\pi_h(\cdot|s) \leftarrow Subroutine(\{Q_{i,h}(s, a_i, x_k)\}_{i,k,a_i})$
    for i **do**
      $V_{i,h}(s) \leftarrow Q_{i,h}(s, \pi_h)$
(Subroutine: solve a Linear Program 4.1.1, scaled by the Lipschitz constant $\gamma_r + \gamma_P(H - h)$, respectively for all x_k;
Pick the solution $\vec{p}$ with the largest objective value among all x_k;
Output a pure strategy sampled from the solution $\vec{p}$ and corresponding backward welfare.)

4.2 Theoretical Guarantees

Now we are ready to present the theoretical guarantees for Algorithm 2.

Theorem 4.1. *For discretizing interval length α, with probability $1 - \beta$, algorithm 2 obtains an $(\zeta + 4\alpha + 2\delta + 2E)(\gamma_r H + \gamma_P \frac{H(H-1)}{2})$-pure strategy Nash Equilibrium, where*

$$\zeta = \delta\sqrt{8n\log(2mn)}, E = O(\sqrt{n}\delta \operatorname{polylog}(d, 1/\beta, H, S))$$

Theorem 4.2. *If social welfare is defined as $\mathbb{E}_{\{s_1, \vec{p}_1, s_2, \ldots\}}[\sum_{h,i,j} f_{ijs_h} p_{ijh}]$ and $OPT(\zeta)$ is the optimal welfare of an approximate NE(that achieves an $\zeta(\gamma_r + \gamma_P(H - h))$-approximate PNE in every step Q-game), our algorithm achieves social welfare of $OPT(\zeta) - \alpha\gamma_P \frac{H(H-1)}{2} l_{max} - HE$*

5 Learning NE in AMG

In this section, we present our second main algorithm—Aggregative Nash Value Iteration (Aggre-VI), and provide its theoretical guarantee.

5.1 Problem formulation

We consider Nash equilibrium (NE) being learned in a fully unknown environment. One goal of reinforcement learning is to design algorithms for Markov games that can find an ϵ-approximate Nash equilibrium using a number of episodes that is small in its dependency on S, A, B, H as well as $1/\epsilon$ (PAC sample complexity bound). An alternative goal is to design algorithms for Markov games that achieves regret that is sublinear in K, and polynomial in S, A, B, H (regret bound). Here we aim to find approximate pure-strategy Nash equilibrium in an efficient way and let ϵ approach the planning result ϵ_{solve} as K goes up.

5.2 Algorithm description

We describe our Nash-VI Algorithm 3. In each episode, the algorithm can be decomposed into two parts.

- Line 3-15 (Optimistic planning from the estimated model): Performs value iteration with bonus using the empirical estimate of the transition $\hat{\mathbb{P}}$, and computes a new (joint) policy π which computed from a Nash subroutine with respect to the estimated value functions;
- Line 19-22 (Play the policy and update the model estimate): Executes the policy π, collects samples, and updates the estimate of the transition $\hat{\mathbb{P}}$.

At a high-level, this two-phase strategy is standard in the majority of model-based RL algorithms, and also underlies provably efficient model-based algorithms such as UCBVI for single-agent (MDP) setting and VI-ULCB for the two-player Markov game setting . However, VI-ULCB has two undesirable drawbacks: the sample complexity is not tight in any of H, S, and $\Pi_i A_i$ dependency, and its computational complexity is PPAD-complete (a complexity class conjectured to be computationally hard .

As we elaborate in the following, our Nash-VI algorithm differs from VI-ULCB in a few important technical aspects, which allows it to significantly improve the sample complexity over VI-ULCB, and ensures that our algorithm terminates in polynomial time.

Before digging into explanations of techniques, we remark that line 16-17 is only used for computing the output policies. It chooses policy π^{out} to be the policy in the episode with minimum gap $(\overline{V}_1 - \underline{V}_1)(s_1)$.

5.2.1 Overview of techniques

Nash Subroutine. Our $Subroutine$ also leverages the algorithm in Cummings et al. [2015] Appendix D, but on the estimation $\overline{Q}$. To compute this, we need $\overline{Q}$ to be Lipschitz.

Extraction of Lipschitzness As Q are lipschitz, their upper confidence bound should be lipschitz as well. To extract lipschitzness of Q, we *solve a joint constraints consisting of confidence interval and lipschitzness*, which in fact can be modeled as a linear programming. Furthermore, Our updating rule catches specialty of lipschitzness and simplifies this approaches, computing upper confidence bounds of Q function more efficiently.

Discretized Estimation In large games, estimating dynamics on the whole action space is usually not practical. Thus we shifted the estimation space of dynamics to the discretized aggregator space. This shift substantially reduced the space complexity of estimation, time complexity of extraction, and the sample complexity or convergence rate.

5.3 Theoretical guarantees

Now we are ready to present the theoretical guarantees for Algorithm 3.

Theorem 5.1. *For discretizing interval length α, with probability $1-p$, after K episodes, under known rewards $\{r_{h,i}\}$, our Algorithm 3 obtains:*

- *an $(\epsilon_{solve} + \epsilon_{dscr} + \sqrt{H^4 S^2 (2W/\alpha)^d \iota / K})$-pure strategy Nash Equilibrium,*
- *regret bounded by $O\left(S\sqrt{H^3 T \iota (2W/\alpha)^d}\right) + K\epsilon_{solve} + K\epsilon_{dscr}$,*

where

$$\epsilon_{solve} = (\zeta + 4\alpha + 2\delta + 2E)(\gamma_r H + \gamma_P \frac{H(H-1)}{2}), \iota = O(\log(SAKH/p)), \epsilon_{dscr} = \gamma_P \alpha H^2$$

Theorem 5.2. *(Same as thm2 but limited on $\overline{Q}$), notice that $\overline{Q}$ is "close" to Q at the optimal episode.*

6 Equilibrium Selection in AMG

Definition 6.1 (Linear separable ?)**.** *Our linear welfare is a sum of agents' actions' respective impacts on the outside society, and is defined as:*

$$Welfare^{\pi} = \mathbb{E}_{\xi \sim \pi}[\sum_{t=1}^{H} \sum_{i=1}^{m} w_{i,h}(s_t, a_{i,t}) | s_1]$$

Algorithm 3 Sample-based Backward Induction

Initialize: for any $(s,h,i,a_i,x_h), \overline{Q}_{h,i}(s,a_i,x) \leftarrow H, \underline{Q}_{h,i}(s,a_i,x) \leftarrow 0, \Delta \leftarrow H, N_h(s,a_i,x) \leftarrow 0, L_h \leftarrow \gamma_r + \gamma_P(H-h)$
for $k = 1, \ldots, K$ **do**
  for $h = H, \ldots, 1$ **do**
    for $s \in \mathcal{S}$ **do**
      for $i \in [m], a_i \in \mathcal{A}_i, x \in X_\alpha$ **do**

$$\overline{Q}^k_{h,i}(s,a_i,x) \leftarrow \min_{x' \in X_\alpha} \left\{ (r_h + \widehat{\mathbb{P}}_h \overline{V}^k_{h+1})(s,a_i,x') + \beta_t(x') + L_h ||x - x'||_\infty + \gamma_P H \alpha \right\}$$

$$\underline{Q}^k_{h,i}(s,a_i,x) \leftarrow \max_{x' \in X_\alpha} \left\{ (r_h + \widehat{\mathbb{P}}_h \underline{V}^k_{h+1})(s,a_i,x') - \beta_t(x') - L_h ||x - x'||_\infty - \gamma_P H \alpha \right\}$$

      $\pi^k_h(\cdot|s), W^k_h(s) \leftarrow Subroutine(\{\overline{Q}^k_{h,i}(s,a_i,x_k)\}_{i,k,a_i})$
(Subroutine: solve a Linear Program 4.1.1, scaled by the Lipschitz constant $\gamma_r + \gamma_P(H-h)$, respectively for all x_k;
Pick the solution $\vec{p}$ with the largest objective value among all x_k;
Output a pure strategy sampled from the solution $\vec{p}$.)
      for $i \in [m]$ **do**
        $\overline{V}_{h,i}(s) \leftarrow \mathbb{D}_{\pi^k} \overline{Q}^k_{h,i}(s)$
        $\underline{V}_{h,i}(s) \leftarrow \mathbb{D}_{\pi^k} \underline{Q}^k_{h,i}(s)$
  if $\max_{i \in [m]} (\overline{V}_{i,1}(s_1) - \underline{V}_{i,1}(s_1)) < \Delta$ **then**
    $\Delta \leftarrow \max_i (\overline{V}_{i,1}(s_1) - \underline{V}_{i,1}(s_1)), \pi_{out} \leftarrow \pi$
  Sample a trajectory$\{(s_h, a_h, r_h)\}$ with $\{\pi_h\}$
  for $h = 1, \ldots, H$ **do**
    d
    (Denote $s = s_h, \boldsymbol{a} = \boldsymbol{a_h}, s' = s_{h+1}$, unique x s.t. $||x - S_h(s,\boldsymbol{a})||_\infty \leq \alpha$)
    $N_h(s,x) + +; N_h(s,x,s') + +; \widehat{\mathbb{P}}_h(s,x,\cdot) \leftarrow \frac{N_h(s,x,\cdot)}{N_h(s,x)}$

7 To do

Remaining questions: 1. Carefully go through details of the Value Iteration Learning algorithm. Can we avoid exponential computational complexity over n?

2. Probability budget. We can only solve NE with high prob? How to avoid certain issues brought by this?

3.

References

R. Cummings, M. Kearns, A. Roth, and Z. S. Wu. Privacy and truthful equilibrium selection for aggregative games. In *Web and Internet Economics: 11th International Conference, WINE 2015, Amsterdam, The Netherlands, December 9-12, 2015, Proceedings 11*, pages 286–299. Springer, 2015.

A Proof of section 3

A.1 Proof of Theorem 3.1

Within state, the lipschitz constant is γ_r. By Theorem C.1 $Subroutine$ leads to $(\zeta + 4\alpha + 2\delta + 2E)\gamma_r$-PNE.

We prove backward:

$$V_1^\dagger(s) - V_1^\pi(s) \leq t_h = \sum_{h'=h}^{H} ((\zeta + 4\alpha + 2\delta + 2E)\gamma_r + \gamma_p \delta(H-h))$$

Assuming this holds in $h+1$, for any unilateral best response deviation at $h, s, i, a_i, a_i', a_{-i}$, we have

$$\begin{aligned}
&V_h^\dagger(s) - V_h^\pi(s) \\
&= Q_{i,h}^\dagger(s,(a_i',a_{-i})) - Q_{i,h}^\pi(s,(a_i,a_{-i})) \\
&= r' - r + \mathbb{P}'V^\dagger - \mathbb{P}V^\pi \\
&= r' - r + (\mathbb{P}' - \mathbb{P})V^\dagger + \mathbb{P}(V^\dagger - V^\pi) \\
&\leq (\zeta + 4\alpha + 2\delta + 2E)\gamma_r + \gamma_p\delta(H-h) + t_{h+1} = t_h
\end{aligned}$$

Summing over h we get

$$V_1^\dagger(s) - V_1^\pi(s) \leq (\zeta + 4\alpha + 2\delta + 2E)\gamma_r H + \delta\gamma_P \frac{H(H-1)}{2}$$

B Planning and Equilibrium Selection in Normal Form Aggregative Games

We adopt the algorithm in [Cummings et al., 2015], which can solve a NE for large aggregative normal form games (NFG). They solve the NE by first discretizing the domain of the aggregator value, then

C Proof of Section 4

Define $Q_{h,i}(s, a_i, x)$ for continuous points $x \in [-W, W]^d$ the same way as in the algorithm, though they're not actually computed.

$$\begin{aligned}
&\min_{\vec{p}} \quad L(\vec{p}) \\
&\forall k \in [d], \quad \hat{x}_k - \alpha \leq \sum_{i=1}^{n}\sum_{j=1}^{m} \hat{f}_{ij}^k p_{ij} \leq \hat{x}_k + \alpha \\
&\forall i \in [n], \quad \forall j \in \vec{BA}_i(\hat{x}_k), \quad 0 \leq p_{ij} \leq 1 \\
&\forall i \in [n], \quad \forall j \notin \vec{BA}_i(\hat{x}_k), \quad p_{ij} = 0 \\
&\forall i \in [n], \quad \sum_{j=1}^{m} p_{ij} = 1
\end{aligned}$$

C.1 Proof of Theorem 4.1

Theorem C.1 (Cummings et al. [2015] Appendix D)**.** *With probability $1-p$, the $Subroutine$ from a γ-lipschitz normal-form aggregative game outputs a $(\zeta + 4\alpha + 2\delta + 2E)\gamma$-PNE.*

The Q and V value are accurate w.r.t. Q^π, V^π, i.e. $Q = Q^\pi, V = V^\pi$ for every (h, i, s, a_i, x) argument pair.
Since

$$\begin{aligned}
|Q_{h,i}(s,a_i,x) - Q_{h,i}(s,a_i,x')| &= |r_{h,i}(s,a_i,x) - r_{h,i}(s,a_i,x') + < \mathbb{P}(s,x) - \mathbb{P}(s,x'), V_{h+1} > | \\
&\leq (\gamma_r + \gamma_P(H-h))||x - x'||_\infty
\end{aligned}$$

, this lipschitz constant leads to an approximate PNE policy π_h, such that

$$\max_{a_i} Q_h^\pi(s,a_i,a_{-i}) - Q_h^\pi(s,\pi_{h,i},a_{-i}) \leq (\zeta + 4\alpha + 2\delta + 2E)(\gamma_r + \gamma_P(H-h)) := \epsilon_h$$

Iteratively, we can prove below backward

$$\text{for all } s, Q_{h,i}^\pi(s,a_i,x) \geq Q_{h,i}^{\dagger,\pi_{-i}}(s,a_i,x) - \sum_{h'=h+1}^{H} \epsilon_{h'} \text{ and } V_{h,i}^\pi(s) \geq Q_{h,i}^{\dagger,\pi_{-i}}(s) - \sum_{h'=h}^{H} \epsilon_{h'}$$

We can get overall regret is bounded by $\sum_{h=1}^{H} \epsilon_h$

C.2 Proof of Theorem 4.2

Proof. As before, we follow the routine of backward induction. Denote the expected future welfare of our policy for state:$W_h^\pi(s)$, for state-action: $W_h^\pi(s, \boldsymbol{a})$, welfare within this timestep and state $w_h(s, \boldsymbol{a})$.

The notation $< \mathbb{P}(s, \boldsymbol{a}), W >$ is a abbreviation of inner product $\sum_{s'} \mathbb{P}(s, \boldsymbol{a}, s') \cdot W(s')$. Then by definition, the optimal welfare should be like

$$W_h^{OPT}(s, \boldsymbol{a}) = w_h(s, \boldsymbol{a}) + < \mathbb{P}(s, \boldsymbol{a}), W_{h+1}^{OPT} >, W_h^{OPT}(s) = \max_{\boldsymbol{a} \in Q-game} W_h^{OPT}(s, \boldsymbol{a})$$

In timestep h, consider the objective: for an aggregator x, state s,

$$\sum_{i,j} w_{ijs} p_{ijs} + < \mathbb{P}(s, x, \cdot), W_{h+1}(\cdot) >$$

This is linear on $\vec{p}$, hence can be minimized by linear programming. As a result, we prove backward:

$$W_h^\pi(s) - W_h^{OPT}(s) \geq -\theta_h = -\sum_{t=h}^{H} (\alpha \gamma_P (H - t) w_{max} + E)$$

Assuming this holds in $h + 1$, denote $x, \boldsymbol{a}$ as the optimal discretized aggregator and outputed action in $Subroutine$, x^{OPT} as the corresponding discretized aggregator of $\boldsymbol{a}^{OPT}$, we have

$$\begin{aligned}
W_h^\pi(s) &= w_h(s, \boldsymbol{a}) + < \mathbb{P}(s, \boldsymbol{a}), W_{h+1}^\pi > \\
&= (w_h(s, \boldsymbol{a}) + < \mathbb{P}(s, x), W_{h+1}^\pi >) + < \mathbb{P}(s, \boldsymbol{a}) - P(s, x), W_{h+1}^\pi > \\
&\text{(w.h.p. LP maximized the bracket with gap } E) \\
&\geq (w_h(s, \boldsymbol{a}^{OPT}) - E + < \mathbb{P}(s, x^{OPT}), W_{h+1}^\pi >) + < \mathbb{P}(s, \boldsymbol{a}) - \mathbb{P}(s, x), W_{h+1}^\pi > \\
&= (w_h(s, \boldsymbol{a}^{OPT}) + < \mathbb{P}(s, \boldsymbol{a}^{OPT}), W_{h+1}^{OPT} >) \\
&\quad + < \mathbb{P}(s, \boldsymbol{a}^{OPT}), W_{h+1}^\pi - W_{h+1}^{OPT} > \\
&\quad + < \mathbb{P}(s, x^{OPT}) - \mathbb{P}(s, \boldsymbol{a}^{OPT}), W_{h+1}^\pi > \\
&\quad + < \mathbb{P}(s, \boldsymbol{a}) - \mathbb{P}(s, x), W_{h+1}^\pi > \\
&\quad - E \\
&\text{(Induction, } W_{h+1} \leq (H - h) w_{max} \text{ and Lipschitz of } \mathbb{P}) \\
&\geq W_h^{OPT}(s, \boldsymbol{a}) - \theta_{h+1} - 2\alpha \gamma_P (H - h) w_{max} - E \\
&= W_h^{OPT}(s) - \theta_h
\end{aligned}$$

Summing over h, we get the overall gap bound $\theta_1 = \alpha \gamma_P H(H - 1) l_{max} + HE$

□

D Proof of Section 5

Define a superscript k of $\overline{Q}$ and $\underline{Q}$ to refer to their values at the end of episode k.

Define $\overline{Q}_{h,i}^k(s, a_i, x)$ and $\underline{Q}_{h,i}^k(s, a_i, x)$ for continuous points $x \in [-W, W]^d$ the same way as in the algorithm, though they're not actually computed.

D.1 Proof of Theorem 5.1

We begin with proving the "upper bound" property of $\overline{Q}_{h,i}$

Denote $\epsilon_h = (\zeta + 4\alpha + 2\delta + 2E)(\gamma_r + \gamma_P(H - h))$, then $\epsilon_{solve} = \sum_{h=1}^{H} \epsilon_h$ To solve the NE, we will need that $\overline{Q}_{h,i}$ is lipschitz.

Lemma D.1 (Lipschitzness of $\overline{Q}$)**.** *For all k, h, i, s, a_i, $\overline{Q}_{h,i}^k(s, a_i, \cdot)$ is L_h-lipschitz·*

Proof. For any x_1, x_2, let

$$x_1' = \arg\min_{x' \in X_\alpha} \left\{ (r_h + \widehat{\mathbb{P}}_h \overline{V}^k_{h+1})(s, a_i, x') + \beta_t(x') + L_h ||x_1 - x'||_\infty + \gamma_P H \alpha \right\}$$

$$x_2' = \arg\min_{x' \in X_\alpha} \left\{ (r_h + \widehat{\mathbb{P}}_h \overline{V}^k_{h+1})(s, a_i, x') + \beta_t(x') + L_h ||x_2 - x'||_\infty + \gamma_P H \alpha \right\}$$

$$\begin{aligned}
\overline{Q'}^k_{h,i}(s, a_i, x_1) - \overline{Q'}^k_{h,i}(s, a_i, x_2) =& (r_h + \widehat{\mathbb{P}}_h \overline{V}^k_{h+1})(s, a_i, x_1') + \beta_t(x_1') + L_h ||x_1 - x_1'||_\infty \\
& - (r_h + \widehat{\mathbb{P}}_h \overline{V}^k_{h+1})(s, a_i, x_3') + \beta_t(x_2') + L_h ||x_2 - x_2'||_\infty \\
\leq & (r_h + \widehat{\mathbb{P}}_h \overline{V}^k_{h+1})(s, a_i, x_2') + \beta_t(x_2') + L_h ||x_1 - x_2'||_\infty \\
& - (r_h + \widehat{\mathbb{P}}_h \overline{V}^k_{h+1})(s, a_i, x_3') + \beta_t(x_2') + L_h ||x_2 - x_2'||_\infty \\
=& L_h ||x_1 - x_2'||_\infty - L_h ||x_2 - x_2'||_\infty \\
\leq & L_h ||x_1 - x_2||_\infty
\end{aligned} \tag{1}$$

Similarly, we can prove

$$\overline{Q'}^k_{h,i}(s, a_i, x_1) - \overline{Q'}^k_{h,i}(s, a_i, x_2) \leq L_h ||x_1 - x_2||_\infty \tag{2}$$

Therefore,

$$|\overline{Q'}^k_{h,i}(s, a_i, x_1) - \overline{Q'}^k_{h,i}(s, a_i, x_2)| \leq L_h ||x_1 - x_2||_\infty \tag{3}$$

Since the clipping doesn't harm lipschitzness, we get

$$|\overline{Q}^k_{h,i}(s, a_i, x_1) - \overline{Q}^k_{h,i}(s, a_i, x_2)| \leq L_h ||x_1 - x_2||_\infty \tag{4}$$

□

Lemma D.2. *(Discretized Uniform Convergence) Consider function class*

$$\mathcal{V}_{h+1} = \{V : \mathcal{S}_{h+1} \to \mathbb{R} \mid V(s) \in [0, H-h] \; for \; all \; s \in \mathcal{S}_{h+1}\}$$

There exists an absolute constant c, with probability at least $1 - p$*, we have:*

$$|(\widehat{\mathbb{P}} - \mathbb{P})V(s, x_k)| \leq c\sqrt{SH^2\iota / N(s, x_k)} + \gamma_p \alpha (H - h)$$

for all (s, h) *and all* $x_k \in (\alpha\mathbb{Z} \cap [-W, W])^d, V \in \mathcal{V}_{h+1}$.

Proof. By a similar statement of uniform convergence from [Bai yu2020], we have for some absolute constant c, with probability at least $1 - p$,$(\widehat{\mathbb{P}} - \mathbb{P})V(s, x_k) \leq c\sqrt{SH^2\iota / N(s, x_k)} := \beta(N(s, x_k))$

Define $N(s, \boldsymbol{a}), N(s, \boldsymbol{a}, \cdot)$ as their natural meaning and fix $N(s, \boldsymbol{a})$ to let randomness be only over transition realization rather than joint action, then by lipschitzness of transition function, $TVD(\mathbb{E}[\frac{N(s,\boldsymbol{a},\cdot)}{N(s,\boldsymbol{a})}], \mathbb{P}(s, x_k, \cdot)) \leq \gamma_p \alpha$. As

$$\frac{\sum_{\boldsymbol{a}: x(\boldsymbol{a}) \approx x_k} N(s, \boldsymbol{a}, \cdot)}{\sum_{\boldsymbol{a}: x(\boldsymbol{a}) \approx x_k} N(s, \boldsymbol{a})} = \sum_{\boldsymbol{a}: x(\boldsymbol{a}) \approx x_k} \frac{N(s, \boldsymbol{a})}{\sum_{\boldsymbol{a}: x(\boldsymbol{a}) \approx x_k} N(s, \boldsymbol{a})} \cdot \frac{N(s, \boldsymbol{a}, \cdot)}{N(s, \boldsymbol{a})}$$

is a linear composition of them, we also have $TVD(\mathbb{E}[\frac{N(s,x_k,\cdot)}{N(s,x_k)}], \mathbb{P}(s, x_k, \cdot)) \leq \gamma_p \alpha$. Futhermore, for any $V \in \mathcal{V}_{h+1}$, $(\mathbb{E}\widehat{\mathbb{P}} - \mathbb{P})V \leq \gamma_p \alpha (H - h)$

$$\begin{aligned}
(\widehat{\mathbb{P}} - \mathbb{P})V &= (\widehat{\mathbb{P}} - \mathbb{E}\widehat{\mathbb{P}})V + (\mathbb{E}\widehat{\mathbb{P}} - \mathbb{P})V \\
&\leq \beta(N(s, x_k)) + \gamma_p \alpha H
\end{aligned}$$

□

We continue the second part of the proof with proving the optimistic estimations are indeed upper bounds of the corresponding V-value and Q-value functions.

Lemma D.3. *With probability $1-p$, for any (s,h,i) and $k\in[K], a_i\in\mathcal{A}_i, x\in[-W,W]^d$:*

$$\overline{Q}^k_{h,i}(s,a_i,x)\geq Q^{\dagger,\pi^k_{-i}}_{h,i}(s,a_i,x)-\sum_{h'=h+1}^{H}\epsilon_{h'},\quad \overline{Q}^k_{h,i}(s,a_i,x)\geq Q^{\pi^k}_{h,i}(s,a_i,x),\quad \underline{Q}^k_{h,i}(s,a_i,x)\leq Q^{\pi^k}_{h,i}(s,a_i,x), \tag{5}$$

$$\overline{V}^k_{h,i}(s)\geq V^{\dagger,\pi^k_{-i}}_{h,i}(s)-\sum_{h'=h}^{H}\epsilon_{h'},\quad \overline{V}^k_{h,i}(s)\geq V^{\pi^k}_{h,i}(s),\quad \underline{V}^k_{h,i}(s)\leq V^{\pi^k}_{h,i}(s). \tag{6}$$

Proof. For each fixed k, we prove this by induction from $h=H+1$ to $h=1$. For the base case, we know at the $(H+1)$-th step,$\overline{V}^k_{H+1,i}(s)=V^{\dagger,\pi^k_{-i}}_{H+1,i}(s)=0$. Now, assume the inequality (6) holds for the $(h+1)$-th step, for the h-th step, by the definition of Q-functions and lemma D.1

$$\begin{aligned}
&\overline{Q}^k_{h,i}(s,a_i,x)-Q^{\dagger,\pi^k_{-i}}_{h,i}(s,a_i,x)\\
=&\left[\widehat{\mathbb{P}}^k_h\overline{V}^k_{h+1,i}\right](s,x')-\left[\mathbb{P}_hV^{\dagger,\pi^k_{-i}}_{h+1,i}\right](s,x)\\
&+r_{h,i}(s,a_i,x')-r_{h,i}(s,a_i,x)+L||x'-x||+\beta_t(s,x')+\gamma_PH\alpha\\
=&\underbrace{\widehat{\mathbb{P}}^k_h\left(\overline{V}^k_{h+1,i}-V^{\dagger,\pi^k_{-i}}_{h+1,i}\right)(s,x)}_{(A)}+\underbrace{\left(\widehat{\mathbb{P}}^k_h-\mathbb{P}_h\right)V^{\dagger,\pi^k_{-i}}_{h+1,i}(s,x')}_{(B)}+\beta_t(s,x')+\gamma_PH\alpha\\
&+\underbrace{(r_{h,i}+\mathbb{P}_hV^{\dagger,\pi^k_{-i}}_{h+1,i})[(s,a_i,x')-(s,a_i,x)]}_{(C)}+L||x'-x||
\end{aligned}$$

By induction hypothesis, for any s', $\left(\overline{V}^k_{h+1,i}-V^{\dagger,\pi^k_{-i}}_{h+1,i}\right)(s')\geq-\sum_{h'=h+1}^{H}\epsilon_{h'}$, and thus $(A)\geq-\sum_{h'=h+1}^{H}\epsilon_{h'}$. By Lemma D.2(discretized uniform concentration), w.p. $1-\frac{p}{3HKSA}$, $|(B)|\leq C\sqrt{SH^2\iota/N^k_h(s,x')}=\beta_t(s,x')+\gamma_PH\alpha$.By Lipschitz assumption, $|(C)|\leq(\gamma_r+\gamma_P(H-h))||x'-x||$. Putting everything together we have $\overline{Q}^k_{h,i}(s,a_i,x)-Q^{\dagger,\pi^k_{-i}}_{h,i}(s,a_i,x)\geq-\sum_{h'=h+1}^{H}\epsilon_{h'}$. The second and third inequality can be proved similarly.

Now assume inequality (5) holds for the h-th step, by the definition of V-functions and ϵ_h-Nash equilibrium(w.p. $1-p/3$ obtained in all execution due to Lipschitz $\overline{Q}$),

$$\overline{V}^k_{h,i}(s)=\mathbb{D}_{\pi^k}\overline{Q}^k_{h,i}(s)\geq\max_{\mu}\mathbb{D}_{\mu\times\pi^k_{-i}}\overline{Q}^k_{h,i}(s)-\epsilon_h.$$

By Bellman equation,

$$V^{\dagger,\pi^k_{-i}}_{h,i}(s)=\max_{\mu}\mathbb{D}_{\mu\times\pi^k_{-i}}Q^{\dagger,\pi^k_{-i}}_{h,i}(s).$$

Since by induction hypothesis, for any (s,a_i,x), $\overline{Q}^k_{h,i}(s,a_i,x)\geq Q^{\dagger,\pi^k_{-i}}_{h,i}(s,a_i,x)-\sum_{h'=h+1}^{H}\epsilon_{h'}$, we further have $\overline{Q}^k_{h,i}(s,\boldsymbol{a})\geq Q^{\dagger,\pi^k_{-i}}_{h,i}(s,\boldsymbol{a})-\sum_{h'=h+1}^{H}\epsilon_{h'}$ for any action profile $\boldsymbol{a}$. As a result, we also have $\overline{V}^k_{h,i}(s)\geq V^{\dagger,\pi^k_{-i}}_{h,i}(s)-\sum_{h'=h}^{H}\epsilon_{h'}$, which is exactly inequality (6) for the h-th step. The second and third inequality can be proved similarly. □

Proof of Theorem 5.1. Let us focus on the i-th player and ignore the subscript when there is no confusion. To bound

$$\max_i\left(V^{\dagger,\pi^k_{-i}}_{1,i}-V^{\pi^k}_{1,i}\right)(s^k_h)\leq\max_i\left(\overline{V}^k_{1,i}-\underline{V}^k_{1,i}\right)(s^k_h)+\sum_{h=1}^{H}\epsilon_h,$$

we notice the following propagation:

$$\begin{cases}(\overline{Q}_{h,i}^k - \underline{Q}_{h,i}^k)(s,a_i,x) \le \widehat{\mathbb{P}}_h^k(\overline{V}_{h+1,i}^k - \underline{V}_{h+1,i}^k)(s,x) + 2\beta_h^k(s,x) + 2\gamma_P H\alpha, \\ (\overline{V}_{h,i} - \underline{V}_{h,i})(s) = [\mathbb{D}_{\pi_h}(\overline{Q}_{h,i}^k - \underline{Q}_{h,i}^k)](s).\end{cases} \tag{7}$$

We can define $\widetilde{Q}_h^k$ and $\widetilde{V}_h^k$ recursively by $\widetilde{V}_{H+1}^k = 0$ and

$$\begin{cases}\widetilde{Q}_h^k(s,x) = \widehat{\mathbb{P}}_h^k\widetilde{V}_{h+1}^k(s,x) + 2\beta_h^k(s,x) + 2\gamma_P H\alpha, \\ \widetilde{V}_h^k(s) = [\mathbb{D}_{\pi_h}\widetilde{Q}_h^k](s).\end{cases} \tag{8}$$

Then we can prove inductively that for any k, h, s and x we have

$$\begin{cases}\max_i(\overline{Q}_{h,i}^k - \underline{Q}_{h,i}^k)(s,a_i,x) \le \widetilde{Q}_h^k(s,x), \\ \max_i(\overline{V}_{h,i} - \underline{V}_{h,i})(s) \le \widetilde{V}_h^k(s).\end{cases} \tag{9}$$

Thus we only need to bound $\sum_{k=1}^K \widetilde{V}_1^k(s)$. Define the shorthand notation

$$\begin{cases}\beta_h^k := \beta_h^k(s_h^k, x_h^k), \\ \Delta_h^k := \widetilde{V}_h^k(s_h^k), \\ \zeta_h^k := [\mathbb{D}_{\pi^k}\widetilde{Q}_h^k]\left(s_h^k\right) - \widetilde{Q}_h^k(s_h^k, x_h^k), \\ \xi_h^k := [\mathbb{P}_h\widetilde{V}_{h+1}^k](s_h^k, x_h^k) - \Delta_{h+1}^k.\end{cases} \tag{10}$$

We can check ζ_h^k and ξ_h^k are martingale difference sequences. As a result,

$$\begin{aligned}\Delta_h^k =& \mathbb{D}_{\pi^k}\widetilde{Q}_h^k\left(s_h^k\right) \\ =& \zeta_h^k + \widetilde{Q}_h^k\left(s_h^k, x_h^k\right) \\ =& \zeta_h^k + 2\beta_h^k + [\widehat{\mathbb{P}}_h^k\widetilde{V}_{h+1}^k]\left(s_h^k, x_h^k\right) + 2\gamma_P H\alpha \\ \le& \zeta_h^k + 3\beta_h^k + [\mathbb{P}_h\widetilde{V}_{h+1}^k]\left(s_h^k, x_h^k\right) + 2\gamma_P H\alpha \\ =& \zeta_h^k + 3\beta_h^k + \xi_h^k + \Delta_{h+1}^k + 2\gamma_P H\alpha.\end{aligned}$$

By Azuma inequality, with probability $1 - p/3$, all terms in Martingale difference sequences are small. Recursing this argument for $h \in [H]$ and taking the sum,

$$\sum_{k=1}^K \Delta_1^k \le \sum_{k=1}^K\sum_{h=1}^H \left(\zeta_h^k + 3\beta_h^k + \xi_h^k + 2\gamma_P H\alpha\right) \le O\left(S\sqrt{H^3T\iota(\frac{2W}{\alpha})^d}\right) + 2K\gamma_P H^2\alpha.$$

There regret bound will be RHS plus $K\sum_{h=1}^H \epsilon_h$.

Thus the output policy with the smallest $\Delta_1^k = \Delta_1^{k^*}$ satisfies that

$$\Delta_1^{k^*} \le \sum_{k=1}^K \Delta_1^k/K \le O(\frac{\cdots}{\sqrt{K}}) + 2\gamma_P H^2\alpha$$

, getting a $(\Delta_1^{k^*} + \epsilon_{solve})$-NE □

D.2 Proof of theorem 5.2

The proof is exactly the same as Proof of theorem 4.2, except that the result policy is selected from the scope of (estimated) $\overline{Q}^{k^*}$.

Algorithm 4 Sample-based Backward Induction under Unknown Reward

Initialize: for any $(s,h,i,a_i,x_h), \overline{Q}_{h,i}(s,a_i,x) \leftarrow H, \underline{Q}_{h,i}(s,a_i,x) \leftarrow 0, \Delta \leftarrow H, N_h(s,a_i,x) \leftarrow 0, L_h \leftarrow \gamma_r + \gamma_P(H-h)$
for $k = 1,\ldots,K$ **do**
  for $h = H,\ldots,1$ **do**
    for $s \in \mathcal{S}$ **do**
      for $i \in [m], a_i \in \mathcal{A}_i, x \in X_\alpha$ **do**

$$\overline{Q}^k_{h,i}(s,a_i,x) \leftarrow \min_{x' \in X_\alpha} \left\{ (r_h + \widehat{\mathbb{P}}_h \overline{V}^k_{h+1})(s,a_i,x') + \beta_t(x') + L_h ||x - x'||_\infty + \gamma_P H \alpha \right\}$$

$$\underline{Q}^k_{h,i}(s,a_i,x) \leftarrow \max_{x' \in X_\alpha} \left\{ (r_h + \widehat{\mathbb{P}}_h \underline{V}^k_{h+1})(s,a_i,x') - \beta_t(x') - L_h ||x - x'||_\infty - \gamma_P H \alpha \right\}$$

      $\pi^k_h(\cdot|s) \leftarrow Subroutine(\{\overline{Q}^k_{h,i}(s,a_i,x_k)\}_{i,k,a_i})$
      (Subroutine4.1.1: solve a linear programming scaled by the Lipschitz constant $\gamma_r + \gamma_P(H-h)$ respectively for all x_k;
      Output a pure strategy sampled from the solution $\vec{p}$.)
      for $i \in [m]$ **do**
        $\overline{V}_{h,i}(s) \leftarrow \mathbb{D}_{\pi^k} \overline{Q}^k_{h,i}(s)$
        $\underline{V}_{h,i}(s) \leftarrow \mathbb{D}_{\pi^k} \underline{Q}^k_{h,i}(s)$
  if $\max_{i \in [m]} (\overline{V}_{i,1}(s_1) - \underline{V}_{i,1}(s_1)) < \Delta$ **then**
    $\Delta \leftarrow \max_i (\overline{V}_{i,1}(s_1) - \underline{V}_{i,1}(s_1)), \pi_{out} \leftarrow \pi$
  Sample a trajectory$\{(s_h,a_h,r_h)\}$ with $\{\pi_h\}$
  for $h = 1,\ldots,H$ **do**
    d
    (Denote $s = s_h, \boldsymbol{a} = \boldsymbol{a_h}, s' = s_{h+1}$, unique x s.t. $||x - S_h(s,\boldsymbol{a})||_\infty \le \alpha$)
    $N_h(s,x) + +; N_h(s,x,s') + +; \widehat{\mathbb{P}}_h(s,x,\cdot) \leftarrow \frac{N_h(s,x,\cdot)}{N_h(s,x)}$

D.3 (Undone)Unknown Reward

Theorem D.1. *For discretizing interval length α, with probability $1-p$, after K episodes, under known rewards $\{r_{h,i}\}$, our Algorithm 3 obtains:*

- *an $(\epsilon_{solve} + \epsilon_{dscr} + \sqrt{H^4 S^2 (2W/\alpha)^d \iota / K})$-pure strategy Nash Equilibrium,*
- *regret bounded by $O\left(S\sqrt{H^3 T \iota (2W/\alpha)^d}\right) + K\epsilon_{solve} + K\epsilon_{dscr}$,*

where

$$\epsilon_{solve} = (\zeta + 4\alpha + 2\delta + 2E)(\gamma_r H + \gamma_P \frac{H(H-1)}{2}), \iota = O(\log(SAKH/p)), \epsilon_{dscr} = \gamma_P \alpha H^2$$