arXiv:2304.04346v1 [cs.AI] 10 Apr 2023

一种采用公共信息方法进行多智能体控制的新型基于点的算法

Dengwang Tang、Ashutosh Nayyar、Rahul Jain

摘要—公共信息（CI）方法提供了一种系统化的方法，可将多智能体随机控制问题转化为单智能体部分可观测马尔可夫决策问题（POMDP），称为协调器的部分可观测马尔可夫决策过程。然而，由于其动作空间极其庞大，这类POMDP可能难以求解。我们提出一种用于多智能体随机控制问题的新算法，称为协调者启发式搜索值迭代（CHSVI）。该算法将CI方法与适用于大动作空间的基于点的POMDP算法相结合。我们通过最优求解若干基准问题来展示该算法。

I. 引言

多智能体控制问题出现在许多应用场景中，包括自主智能体或机器人团队执行任务、拥有多个用户的通信网络，以及智能电网等分布式系统。此类问题通常涉及多个智能体，它们可能掌握不同的信息，并采取动作与动态系统交互。在合作型多智能体问题中，智能体的目标是优化一个共同的性能指标。

在本研究中，我们假设可以在与系统交互之前，离线共同确定控制策略（或如分散式部分可观测马尔可夫决策过程文献所述，采用集中式规划和分散式执行）。即便采用集中式规划，多智能体控制问题也可能比相应的单智能体问题难得多，原因有多种：（1）由于所有智能体的信息、动作和系统演化动态彼此相互依存，我们需要同时确定所有智能体的策略；（2）智能体的决策过程不仅涉及对底层状态的估计，还涉及对其他智能体信息的估计，因为预测其他智能体的动作至关重要。同理，智能体还需要理解其他智能体可能如何看待自身的信息和动作，等等，因此需要构建复杂的信念层级。

有若干种结构化方法，可以将多智能体控制问题转化为或分解为单智能体控制问题。其中一种方法是公共信息（CI）方法 [1]。在这种方法中，通过假设存在一个虚构的参与者，称为协调者，将多智能体控制问题转化为单智能体部分可观测马尔可夫决策问题（POMDP）。在每个时间点，每个智能体可用的信息被划分为两部分：公共信息和私有信息。在每个时间点，不是让每个智能体分别决定其动作，而是由协调者为每个智能体选择一个处方，即将该智能体的私有信息映射到其动作的映射函数。处方的选择仅基于公共信息。原则上，协调者的问题可以使用单智能体POMDP算法求解。

然而，单智能体部分可观测马尔可夫决策过程并不容易求解。 [2], [3]提出了通过更新价值函数的一组支撑向量（即 α-向量）来求解部分可观测马尔可夫决策过程的精确值迭代方法。不过，这些方法仅适用于状态数量极少的问题 [4]。一类称为基于点的 方法 [5], [6], [7] 的近似求解方法，在近似求解具有数百或数千个状态的部分可观测马尔可夫决策过程问题方面取得了相当大的成功（例如，基于点的值迭代 [5] 是首个在拥有 870 个状态的*Tag* 问题上展现出优异性能的算法。HSVI2 [8] 能够处理拥有 12,545 个状态的 *RockSample*[7,8]问题）。这些方法基于以下前提：无需计算所有 α-向量，而是可以通过基于点的备份 过程高效地近似贝尔曼更新，即计算一小部分 α-向量，用以表示更新后的价值函数在特定信念点处的梯度（关于这类方法的全面综述和教程，见 [4] ）。

然而，与典型的单智能体部分可观测马尔可夫决策过程相比，协调器的部分可观测马尔可夫决策过程具有天文数字般 庞大的动作空间，因为其动作是映射。（即使是像 `DecTiger` (2,1,β)这样看似规模很小的问题（见附录A.1）也包含数百万个动作。）庞大的动作空间给将最先进的基于点的方法用于协调器的部分可观测马尔可夫决策过程带来了重大挑战，原因在于：(1)备份过程需要在动作空间上求解离散优化问题。(2) 某些基于点的方法的信念探索过程 [6], [7]也需要在动作空间上求解离散优化问题。因此，尽管CI方法将多智能体问题转化为了单智能体问题，但仍然需要一种专用算法来求解这些特殊的单智能体问题。

在本文中，我们提出了一种用于多智能体控制的新算法，该算法将CI方法与适用于大动作空间的基于点的部分可观测马尔可夫决策过程求解方法相结合。我们在若干具有巨大动作空间的基准问题上展示了该新算法的性能。

相关工作： CI方法 [1] 已被应用

提交至第62届IEEE决策与控制会议（CDC2023）的较短版本。本研究得到美国海军研究办公室项目N00014-20-1-2258以及美国国家科学基金会项目ECCS-2025732和ECCS-1750041的资助。D. Tang、A. Nayyar和R. Jain任职于南加州大学明·谢赫电气与计算机工程系，地址：加利福尼亚州洛杉矶 90089 2560 {dengwang，ashutosh.nayyar，-。电子邮箱: rahul.jain}@usc.edu

在许多场景中。在 [9], [10], 中，作者应用该方法，为某些类别的多智能体通信问题建立了结构性结论。在 [11], 中，作者将该方法扩展至涵盖私有信息压缩。在 [12], [13], [14], 中，作者提出了将 CI方法与近似规划和学习方法相结合的构想。尽管利用 CI方法已经取得了许多理论成果 [1], [10], [11], [14],，但基于该方法的高效规划算法仍然缺乏。因此，经验结果要么是在私有信息空间非常小的情况下建立的[13], ，要么是通过机器学习技术获得的 [15]。

分散式部分可观测马尔可夫决策过程（Dec-POMDP）是部分可观测马尔可夫决策过程（POMDP）的多智能体扩展，已得到广泛研究（综述见 [16] ），其中也应用了基于点的方法（例如 [17], [18], [19]）。在许多分散式部分可观测马尔可夫决策过程算法中，策略表示为策略树 [19], ，其规模会随时间范围呈指数级增长。因此，这类算法需要大量内存，只能求解有限时间范围较小的问题。也有一些针对无限时间范围问题设计的分散式部分可观测马尔可夫决策过程算法 [20], [21]，但这些算法无法提供最优性保证。

基于点的方法已应用于具有部分历史共享的多智能体问题。例如，在 [22],中，作者将基于点的方法应用于智能体之间能够通信的有限时间范围问题。然而，据我们所知，我们的工作首次将CI方法与基于点的部分可观测马尔可夫决策过程方法相结合，用于一般的无限时间范围多智能体问题。

贡献： (1) 在与CI方法相关的研究背景下，我们提出了首个能够求解具有大动作空间的一般协调器的部分可观测马尔可夫决策过程的实用算法。(2)在分散式部分可观测马尔可夫决策过程相关文献的背景下，我们为无限时域问题提供了一种内存高效的随时可用算法，并给出了输出策略价值的上界和下界。

II. 预备知识

A. 问题表述

1）部分历史共享的多智能体控制：我们考虑一个由元组 $\mathcal{E} = (\mathcal{I},\ \mathcal{S},\ \mathcal{A},\ \mathcal{M},\ \mathcal{O},\ b_0,\ \mathbb{P},\ r,\ \beta)$ 表征的无限时域多智能体控制模型，其中 $\mathcal{I}$ 是一个有限的智能体集合；

- $\mathcal{S}$ 是表示状态空间的有限集合；
- $\mathcal{A} = \prod_{i\in\mathcal{I}} \mathcal{A}^i$，其中 $\mathcal{A}^i$ 是一个有限集合，表示智能体 i的动作空间；
- $\mathcal{M} = \prod_{i\in\mathcal{I}} \mathcal{M}^i$，其中 $\mathcal{M}^i$ 是一个有限集合，表示智能体 i的私有信息域；
- $\mathcal{O}$ 是一个有限集合，表示公共观测域（在时间 t 时的公共观测，允许在时间 t之前已被某些智能体私下观测到）；
- $b_0 \in \Delta(\mathcal{S}\times\mathcal{M})$ 是所有智能体的状态和私有信息的初始联合分布；
- $\mathbb{P}$： $\mathcal{S}\times\mathcal{M}\times\mathcal{A} \mapsto \Delta(\mathcal{S}\times\mathcal{M}\times\mathcal{O})$ 是状态—信息联合转移核；
- r： $\mathcal{S}\times\mathcal{A}\mapsto\mathbb{R}$ 是即时奖励；
- $\beta \in (0,1)$ 是折扣因子。

参与者 i 在时间 t 的信息空间为 $\mathcal{H}_t^i = (\mathcal{O})^t \times \mathcal{M}^i$，它表示从时间 1 到 t 的公共观测，以及时间 t的私有信息。动作可能会被所有智能体共同观测到，也可能不会。如果任何a_t^i 被所有智能体共同观测到，那么它就是公共观测 o_{t+1}的一部分。该问题的目标是选择一个联合策略 $\pi = (\pi^i)_{i\in\mathcal{I}}$，$\pi^i = (\pi_t^i)_{t=0}^{\infty}$，$\pi_t^i : \mathcal{H}_t^i \mapsto \mathcal{A}^i$，以最大化总折扣奖励 $J(\pi)$，即

$$\max_{\pi} \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \beta^t r(s_t, a_t)\right].$$

如 [1], 所示，该模型可用于建模许多具有部分历史共享的分散式决策与控制问题，即某些智能体子集的动作与观测历史为各方共同知晓的问题。

注释 1。在 [11], 作者引入了充分私有信息的概念，它是对私有信息的一种压缩，足以满足决策目的。可以将该模型中的私有信息 $\mathcal{M}$空间替换为充分私有信息空间，本文提出的算法仍然适用。

*2)*转换为协调器的部分可观测马尔可夫决策过程：遵循 [1], 中引入的CI方法，我们将该问题 $\mathcal{E}$ 转换为一个等价的部分可观测马尔可夫决策过程问题，称为协调器的部分可观测马尔可夫决策过程，该过程可由一个元组 $\bar{\mathcal{E}} = (\mathcal{S},\ \bar{\mathcal{A}},\ \mathcal{O},\ b_0,\ \bar{\mathbb{P}},\ \bar{r},\ \beta)$表示，其中 $\mathcal{O}$，b_0，β 与原始模型中的相同； $\mathcal{S} = \mathcal{S}\times\mathcal{M}$ 是（增广的）状态空间； $\bar{\mathcal{A}} = \prod_{i\in\mathcal{I}} \bar{\mathcal{A}}^i$ 是（新的）动作空间，其中 $\bar{\mathcal{A}}^i$ 是从 $\mathcal{M}^i$ 到 $\mathcal{A}^i$的映射集合（称为处方）； $\bar{\mathbb{P}}$： $\mathcal{S}\times\mathcal{A}\mapsto\Delta(\mathcal{S}\times\mathcal{O})$是组合的状态转移和观测核； $\bar{r}$： $\mathcal{S}\times\mathcal{A}\mapsto\mathbb{R}$ 是瞬时奖励函数。 $\bar{\mathbb{P}}$ 和 $\bar{r}$ 分别定义为

$$\bar{\mathbb{P}}(\bar{s}', o|\bar{s}, \gamma) = \mathbb{P}(\bar{s}', o|\bar{s}, a) \quad \forall \bar{s}, \bar{s}' \in \bar{\mathcal{S}}, o \in \mathcal{O}, \gamma \in \bar{\mathcal{A}}$$
$$\bar{r}(\bar{s}, \gamma) = r(s, a) \quad \forall \bar{s} \in \bar{\mathcal{S}}, \gamma \in \bar{\mathcal{A}}$$

其中 $\bar{s} = (s, m)$ $a = (a^i)_{i\in\mathcal{I}}$ 和 $a^i = \gamma^i(m^i)$。

本文聚焦于协调器的部分可观测马尔可夫决策过程及其变体。不失一般性，我们移除上的上划线，并假设私有信息是状态的固定函数。对于 $s \in \mathcal{S}$，我们用 $m_s = (m_s^i)_{i\in\mathcal{I}} \in \mathcal{M}$表示其对应的私有信息。为简化表达，我们也去掉 $\mathbb{P}$ 和 $\bar{r}$ 上的上划线。请注意，我们仍使用 $\mathcal{A}$ 表示处方空间，以便将其与单个智能体的动作空间区分开来。备注 2。即使是看似简单的问题，例如具有 $N \le 3$ 扇门和 1 步延迟信息共享的 DecTiger问题（见附录 A. 1 $N = 2$ $|\mathcal{A}| =$），$\prod_{i=1}^{2} |\mathcal{A}^i|^{|\mathcal{M}^i|} = 3^{14} \approx 4.78\times10^6$ $N = 3$ 协调器的部分可观测马尔可夫决策过程也可能具有大到难以承受的动作空间。例如，对于 ，我们有 ；对于 ，我们有 $|\bar{\mathcal{A}}| = \prod_{i=1}^{2} |\mathcal{A}^i|^{|\mathcal{M}^i|} = 4^{26} \approx 4.50\times10^{15}$。对于动作空间如此庞大的部分可观测马尔可夫决策过程，大多数计算机甚至没有足够的内存来初始化现成的部分可观测马尔可夫决策过程求解器，更不用说运行它们了。

B. 用于部分可观测马尔可夫决策过程的基于点的算法

在本节中，我们概述部分可观测马尔可夫决策过程的基于点的算法及其共同组成部分：基于点的备份操作。随后，我们讨论启发式搜索价值迭代（HSVI）算法 [6], ，该算法将作为我们所提出算法的基础。为描述本节中的算法，我们考虑一个标准的单智能体无限时域折扣奖励部分可观测马尔可夫决策过程，其通过一个元组$\mathcal{E} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, b_0, \mathbb{P}, r, \beta)$定义。

1) 基于点的算法和备份操作：对于一个部分可观测马尔可夫决策过程，价值函数被定义为信念状态的函数，贝尔曼算子定义如下：对于任意函数 V： $\Delta(\mathcal{S}) \mapsto \mathbb{R}$,

$$TV(b) := \max_{a \in \mathcal{A}} \left[r(b,a) + \beta \sum_{o \in \mathcal{O}} \mathbb{P}(o|b,a) V(\tau(b,a,o)) \right]$$

其中 $b \in \Delta(\mathcal{S})$和 τ 是信念更新函数。由于已知价值函数 V 是分段线性且凸的，因此可以将其写为 $V(b) = \max_{\alpha \in \mathcal{V}} \alpha^T b$，其中 $\mathcal{V}$是有限个 $|\mathcal{S}|$–维向量的集合。然后我们有

$$\begin{aligned}
& TV(b) \\
&= \max_{a \in \mathcal{A}} r(b,a) + \sum_{o \in \mathcal{O}} \mathbb{P}(o|b,a) \max_{\alpha \in \mathcal{V}} [\tau(b,a,o)]^T \alpha \\
&= \max_{a \in \mathcal{A}} r(b,a) + \sum_{o \in \mathcal{O}} \max_{\alpha \in \mathcal{V}} [\mathbb{P}(o|b,a) \tau(b,a,o)]^T \alpha \\
&= \max_{a \in \mathcal{A}} r(b,a) + \sum_{o \in \mathcal{O}} \max_{\alpha \in \mathcal{V}} \sum_{s' \in \mathcal{S}} \mathbb{P}(s', o|b,a) \alpha(s') \\
&= \max_{a \in \mathcal{A}} \max_{\mu \in \mathcal{V}^{\mathcal{O}}} r(b,a) + \sum_{o \in \mathcal{O}} \sum_{s' \in \mathcal{S}} \mathbb{P}(s', o|b,a) \mu_o(s') \\
&= \max_{a \in \mathcal{A}} \max_{\mu \in \mathcal{V}^{\mathcal{O}}} (\alpha^{a,\mu})^T b
\end{aligned}$$

其中， $\mathcal{V}^{\mathcal{O}}$ 是从 $\mathcal{O}$ 到 $\mathcal{V}$ 的所有映射的集合，且

$$\alpha^{a,\mu}(s) := r(s,a) + \beta \sum_{o \in \mathcal{O}} \sum_{s' \in \mathcal{S}} \mathbb{P}(s', o|s,a) \mu_o(s').$$

因此，可以通过更新 α-向量集合 [2], [3], [4] 来求得价值函数。

$$\mathcal{V}^{(k+1)} := \{\alpha^{a,\mu} : a \in \mathcal{A}, \mu \in (\mathcal{V}^{(k)})^{\mathcal{O}}\}. \tag{1}$$

然而，计算 (1) 的效率很低：我们可以看到，$|\mathcal{V}^{(k+1)}| = |\mathcal{A}| \cdot |\mathcal{V}^{(k)}|^{|\mathcal{O}|}$，也就是说， α-向量数量的增长是双指数级的。即使有用于剪除被支配的 α-向量的过程 [3], , 精确值迭代仍然不适合处理大型状态空间和动作空间 [4]。

虽然在贝尔曼更新中计算所有 α向量的代价很高，但计算一个能在特定信念点 $b \in \Delta(\mathcal{S})$ 处支撑更新后函数的 α 向量（即对于所有 ( ), $TV(b)\ \alpha^T b \quad TV\ \tilde{b}\ \geq \alpha^T \tilde{b} \quad \tilde{b} \in \Delta\ \mathcal{S}$ and ( )) 并不昂贵：可以通过 α^b 计算该向量。

$$\alpha^{b,a,o} := \arg\max_{\alpha \in \mathcal{V}} \sum_{s \in \mathcal{S}} b(s) \sum_{s' \in \mathcal{S}} \mathbb{P}(s', o|s,a) \alpha(s') \tag{2}$$
$$\forall a \in \mathcal{A}, o \in \mathcal{O}$$

$$\alpha^{b,a}(s) := r(s,a) + \beta \sum_{o \in \mathcal{O}} \sum_{s' \in \mathcal{S}} \mathbb{P}(s', o|s,a) \alpha^{b,a,o}(s') \tag{3}$$
$$\forall s \in \mathcal{S}, a \in \mathcal{A}$$

$$\alpha^b := \arg\max_{\alpha^{b,a} : a \in \mathcal{A}} \left(\alpha^{b,a}\right)^T b \tag{4}$$

上述过程被称为信念点 b 处的（基于点的）备份过程。部分可观测马尔可夫决策过程的基于点的算法基于这样一种思想：通过在经过精心选择的信念点子集上使用备份过程，可以获得一组 α-向量，从而对最优价值函数进行良好的近似。

函数 `GenericPointBased`**：输入：** 单智能体部分可观测马尔可夫决策过程模型（$\mathcal{S}$、 $\mathcal{A}$、 $\mathcal{O}$、 b_0、 $\mathbb{P}$、 r、 β）**输出：** 基于信念的策略 π 和 $V^{\pi}(b_0)$的下界 初始化 $\mathcal{V}$； **当** 停止准则未满足时 **执行**选择信念点集 $\mathcal{B}$； $\alpha^b = \texttt{Backup}(\mathcal{V}, b)$ 对所有 $b \in \mathcal{B}$； 将 $\{\alpha^b\}_{b \in \mathcal{B}}$ 添加到 $\mathcal{V}$； 偶尔修剪 $\mathcal{V}$ 中的被支配向量； π: $=$ `DirectControlPolicy`$(\mathcal{V})$； v : $=$ 最大值$_{\alpha \in \mathcal{V}} \alpha^T b_0$； **返回** π， v

算法1： 用于单智能体部分可观测马尔可夫决策过程的通用基于点的算法

此外，如果初始集合 $\mathcal{V}$ 表示最优价值函数的一个均匀可改进下界（定义见 [23]），那么在通过基于点的更新插入新的 α-向量后， $\mathcal{V}$ 仍将是一个均匀可改进下界。还可以证明，在这种情况下，直接控制策略[1] 的值（由 $\mathcal{V}$ 得到）下界为 V，即由 α-向量 $\mathcal{V}$构成的分段线性凸函数。因此，我们对所得策略具有理论保证。许多基于点的算法（例如 [25], [6], [7], [26]）的通用框架如算法1所示。这些算法都采用相同的备份过程。它们的主要区别在于如何确定要执行备份的信念点集合。

2) 启发式搜索价值迭代： [6] 中的HSVI算法总结于算法2中。除了表示 V^*最优价值函数下界的一组α-向量外，该算法还维护 V^*的上界。在每个时间点，该算法执行深度优先搜索以选择少量信念点。该算法在这些信念点处同时更新上界和下界。在这些更新中，该算法在这些信念点处计算上界和下界的精确贝尔曼更新，

[1]直接控制策略 [24], [8] 可描述如下：在信念b下，找到使 $\alpha^T b$最大化的 α-向量 $\alpha^* \in \mathcal{V}$，并采取与 α^*相关联的动作 a^*，即在最初计算 α^* 时式（4）中的最大化动作。

并将新值纳入界中。因此，随着时间推移，下界不断提高，上界不断降低。HSVI 是一种随时可用的算法，始终致力于缩小上界与下界之间的差距。当算法在满足某个停止条件后终止时，它会返回与用于构造下界函数的 α-向量对应的直接控制策略。

HSVI算法通过采用基于上界的启发式搜索过程区别于早期算法，该过程定义于函数ChooseNext中。为确定下一个要探索的信念，算法首先乐观地选择一个使与上界相关的动作价值函数（或Q函数）最大化的动作。随后，算法选择一个使超额间隙最大化的观测。这一过程使算法能够聚焦于当前上界与下界之间的间隙尚不理想的信念。

下界函数 L 在 HSVI 中表示为一组 α-向量。其更新函数 $L.\texttt{Update}(b)$通过在 b执行备份操作，将一个新的 α-向量添加到该集合中。算法还会定期剪枝集合中某些被支配的 α-向量。

与下界不同，HSVI 中的上界函数通过一组孤立点 $(b, \bar{v}_b) \in \Delta(\mathcal{S}) \times \mathbb{R}$ 的下凸包来表示，即，设 $\mathbf{B}$ 为一个矩阵，其 n 列向量表示 n 信念，且 $\bar{v} \in \mathbb{R}^n$ 为一个表示这些信念的上界值的向量，那么对于任意 $b \in \Delta(\mathcal{S})$,

$$U(b) = \min\{\bar{v}^T \eta : \eta \in \mathbb{R}^n_+, \mathbf{B}\eta = b.\} \tag{5}$$

该表示基于 V^* 是凸的这一事实。令 T 为第二-B.1节中定义的贝尔曼算子。算法2中的 $U.\texttt{Update}(b)$ 过程计算

$$\bar{v}_b := TU(b) = \max_{a \in \mathcal{A}} r(b,a) + \beta \sum_{o \in \mathcal{O}} \mathbb{P}(o|b,a) U(\tau(b,a,o)) \tag{6}$$

然后将 $(b, \bar{v}_b)$ 添加到用于计算凸包的点集。该算法还会定期移除集合中的冗余点。该过程返回 $a^* \in \mathcal{A}$，即 (6) 的一个优化器，以供未来在搜索启发式方法中使用。

函数 HSVI:
  输入: 单智能体部分可观测马尔可夫决策过程模型 $(\mathcal{S}, \mathcal{A}, \mathcal{O}, b_0, \mathbb{P}, r, \beta)$
  输出: 一个基于信念的策略 π；一个上界用于 $V^*(b_0)$，以及 $V^\pi(b_0)$ 的下界。
  // $\zeta \in (0, 1)$ is a hyperparameter.
  初始化 U 和 L;
  当 停止准则未满足 **执行**
    $\epsilon = \zeta[U(b_0) - L(b_0)]$;
    Explore(U, L, b_0, ϵ);
  $\pi =$ DirectControlPolicy(L);
  return $\pi, U(b_0)\ L(b_0)$
函数 Explore(U, L, b, ϵ):
  // Modifies U, L
  $a^* = U.\texttt{Update}(b)$;
  $L.\texttt{Update}(b)$;
  如果 $U(b) - L(b) \leq \epsilon$ **则返回**;
  $(b', \epsilon') =$ ChooseNext(U, L, b, a^*, ϵ);
  Explore(U, L, b', ϵ');
  $U.\texttt{Update}(b)$;
  $L.\texttt{Update}(b)$;
函数 ChooseNext(U, L, b, a^*, ϵ):
  $o^* = \arg\max_{o \in \mathcal{O}} \mathbb{P}(o|b,a^*)[U(\tau(b,a^*,o)) - L(\tau(b, a^*, o)) - \frac{\epsilon}{\beta})]$;
  返回 $\tau(b, a^*, o^*)\ \frac{\epsilon}{\beta}$;

算法2: 启发式搜索价值迭代

III. 协调者的 HSVI 算法

如前所述，与常规的单智能体部分可观测马尔可夫决策过程相比，协调器的部分可观测马尔可夫决策过程拥有数量呈指数级增长的动作。这使得 HSVI 的下界更新和上界更新都无法实现。具体来说，在备份操作 (2)–(4) 中，我们需要先求解 $\lrcorner|\mathcal{A}|\times|\mathcal{O}|$ 个离散优化问题 (2)，然后在 $\mathcal{A}$ 上求解 $^{-}$ 个优化问题 (4)。此外，要在信念点 b 处执行上界的贝尔曼更新，HSVI 算法需要针对 $(\gamma, o) \in \mathcal{A} \times \mathcal{O}$ 的所有对 $^{-}$ 计算 $U(\tau(b, \gamma, o))$。由于协调器的部分可观测马尔可夫决策过程中的动作数量庞大，这同样无法实现。

在本节中，我们介绍协调器启发式搜索值迭代（CHSVI）算法，该算法结合了CI方法与HSVI算法。我们不直接在协调器的部分可观测马尔可夫决策过程上应用HSVI，而是在协调器的部分可观测马尔可夫决策过程的多步扩展形式上应用HSVI。借助扩展形式和处方空间的结构，我们能够简化上界和下界更新操作，从而构造出更实用的算法。

我们首先描述扩展形式的协调器的部分可观测马尔可夫决策过程。为便于说明，考虑 $\mathcal{I} = \{1, 2\}$，尽管这一思想自然可以扩展到两个以上的参与者。我们从协调器的$^{-}$ $^{-}$部分可观测马尔可夫决策过程 $\mathcal{E} = \{\mathcal{S}, \mathcal{A}, \mathcal{O}, b_0, \mathbb{P}, r, \beta\}$ 出发，通过将每个时间t 扩展为三个阶段，得到扩展的部分可观测马尔可夫决策过程 $\mathcal{E}$：$(t, 0)$、$(t, 1)$、$(t, 2)$。状态空间为$\mathcal{S}^0$：在 $(t, 0)$ 时为 $= \mathcal{S}$，在 $(t, 1)$ 时为 $\mathcal{S}^1$，在 $(t, 2)$ 时为 $\mathcal{S}^2$。在 $(t, 0)$ 和 $(t, 1)$ 时，公共观测空间均为 $\{\emptyset\}$；在 $(t, 2)$ 时为 $\mathcal{O}$。只有阶段 $(t, 2)$ 具有即时奖励，该奖励对于 $(s, a) \in \mathcal{S}^2$ 为 $r(s, a)$。系统的演化如下:

1)在$(t,0)$处，协调者仅为智能体1选择处方 $\gamma^1 \in \bar{\mathcal{A}}^1$。状态从 $s \in \mathcal{S}$ 确定性地转移至$(s, \gamma^1(m_s^1)) \in \mathcal{S} \times \mathcal{A}^1$。
2)在$(t,1)$处，协调者仅为智能体2选择处方 $\gamma^2 \in \bar{\mathcal{A}}^2$。状态从$(s, a^1) \in \mathcal{S} \times \mathcal{A}^1$ 确定性地转移至$(s, a^1, \gamma^2(m_s^2)) \in \mathcal{S} \times \mathcal{A}^1 \times \mathcal{A}^2$。3)在$(t,2)$处，协调者无需采取任何行动。状态—信息联合转移核只是与初始模型 $\mathcal{E}$中定义的相同 $\mathbb{P}(s', o|s, a)$。

新部分可观测马尔可夫决策过程的总奖励为

$\sum_{t=0}^{\infty}\beta^t r(s_{(t,2)}, a_{(t,2)})$。略微滥用一下符号，对于 $\gamma^i \in \bar{\mathcal{A}}^i$，我们定义 $\gamma^i(a^i|m^i) := \mathbf{1}_{\{\gamma^i(m^i)=a^i\}}$。三个阶段的信念更新函数为

$$[\tau^0(b,\gamma^1)](s,a^1) = b(s)\gamma^1(a^1|m_s^1)$$
$$[\tau^1(b,\gamma^2)](s,a^1,a^2) = b(s,a^1)\gamma^2(a^2|m_s^2)$$
$$[\tau^2(b,o)](s') = \frac{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathbb{P}(s',o|s,a)b(s,a)}{\sum_{\tilde{s}\in\mathcal{S}}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathbb{P}(\tilde{s},o|s,a)b(s,a)}$$

ˆ扩展协调者的部分可观测马尔可夫决策过程 $\mathcal{E}$ 与¯协调者的部分可观测马尔可夫决策过程 $\mathcal{E}$ 的区别仅在于一个时间点内事件的顺序 t，而不在于时间 t 的总奖励以及从 t 到 $t+1$ 的ˆ¯动态。因此， $\mathcal{E}$ 与 $\mathcal{E}$等价：ˆ¯ $\mathcal{E}$ 中的任何策略在 $\mathcal{E}$ 中都有一个对应策略，且二者的总奖励相同，反之亦然。

在CHSVI算法（算法3）中，我们将HSVI的框架应用于扩展协调者的部分可观测马尔可夫决策过程，同时利用处方空间的特殊结构，更高效地实现更新过程。在以下各节中，我们将详细介绍CHSVI算法中使用的上界 U 和下界 L。

注 3：在本节中，我们采用一种不可知的方法，将任意协调者的部分可观测马尔可夫决策过程转换为扩展协调者的部分可观测马尔可夫决策过程，其中中间状态空间为 $\mathcal{S}\times\mathcal{A}^1$ 和 $\mathcal{S}\times\mathcal{A}^1\times\mathcal{A}^2$，并且在每个阶段只有一个智能体（包括自然）采取行动。实际上，CHSVI算法的思想可以应用于协调者的部分可观测马尔可夫决策过程的任意转换，只要在每个阶段仅有一个智能体采取行动即可。

A. 下界更新

在 CHSVI 中，我们为 t 的三个阶段分别维护三个下界函数（ α-向量集合）。令 L^ℓ 表示下界函数， $\mathcal{V}^\ell$ 表示阶段 ℓ （某个时间 t）对应的 α- 向量集合。令 T^ℓ 表示阶段 ℓ 的贝尔曼算子。下面详细说明更新步骤。

阶段 $\ell = 0, 1$：在此阶段，协调者为智能体 $i=\ell+1$ 选择一个处方。对于 $b\in\Delta(\mathcal{S}^\ell)$，在 b 处的下界贝尔曼更新为

$$[T^\ell L^{\ell+1}](b) = \max_{\gamma^i\in\bar{\mathcal{A}}^i} L^{\ell+1}(\tau^\ell(b,\gamma^i))$$
$$= \max_{\gamma^i\in\bar{\mathcal{A}}^i}\max_{\alpha\in\mathcal{V}^i}\sum_{(s^\ell,a^i)\in\mathcal{S}^\ell\times\mathcal{A}^i} b(s^\ell)\gamma^i(a^i|m^i_{s^\ell})\alpha(s^\ell,a^i) \quad (7)$$

优化问题（7）可通过以下步骤求解：

$$\gamma^{i,\alpha}(m^i) := \arg\max_{a^i\in\mathcal{A}^i}\sum_{s^\ell\in\mathcal{S}^\ell:m^i_{s^\ell}=m^i} b(s^\ell)\alpha(s^\ell,a^i) \quad (8)$$
$$\forall m^i\in\mathcal{M}^i, \alpha\in\mathcal{V}^i$$
$$J(\alpha) := \sum_{s^\ell\in\mathcal{S}^\ell} b(s^\ell)\alpha(s^\ell,\gamma^{i,\alpha}(m^i_{s^\ell})) \quad \forall\alpha\in\mathcal{V}^i \quad (9)$$
$$\alpha^* := \arg\max_{\alpha\in\mathcal{V}^i} J(\alpha) \quad (10)$$
$$\gamma^{i,*} := \gamma^{i,\alpha^*}, \quad (11)$$

Function `CHSVI`:
- **Input:** A coordinator's POMDP model $(\mathcal{S},\bar{\mathcal{A}},\mathcal{O},b_0,\mathbb{P},r,\beta)$
- **Output:** A belief based coordination policy π; an upper bound for $V^*(b_0)$ and lower bound for $V^\pi(b_0)$.
- `//` $\zeta\in(0,1)$ `is a hyperparameter.`
- Initialize U and L;
- **while** *Stopping criterion not satisfied* **do**
  - $\epsilon = \zeta[U(b_0)-L(b_0)]$;
  - `Explore`(U,L,b_0,ϵ);
- $\pi =$ `DirectControlPolicy`(L);
- **return** $\pi, U(b_0), L(b_0)$

Function `Explore`(U,L,b,ϵ):
- $\gamma^* = U$.`Update`(b);
- L.`Update`(b);
- **if** $U(b)-L(b)\le\epsilon$ **then return**;
- $(b',\epsilon') =$ `ChooseNext`$(U,L,b,\gamma^*,\epsilon)$;
- `Explore`(U,L,b',ϵ');
- U.`Update`(b);
- L.`Update`(b);

Function `ChooseNext`$(U,L,b,\gamma^*,\epsilon)$:
- Set $\ell\in\{0,1,2\}$ to be such that $b\in\Delta(\mathcal{S}^\ell)$;
- **if** $\ell<2$ **then**
  - **return** $\tau^\ell(b,\gamma^*),\epsilon$;
- **else**
  - $o^* = \arg\max_{o\in\mathcal{O}}\mathbb{P}(o|b)[U(\tau^2(b,o)) - L(\tau^2(b,o)) - \epsilon\beta^{-1}]$;
  - **return** $\tau^2(b,o^*),\epsilon\beta^{-1}$;

Al追踪Im：3协调员追踪 t'mHSVI Al i h

这里我们利用了这样一个事实：对于每个固定的 $\alpha\in\mathcal{V}^i$，关于 $\gamma^i\in\bar{\mathcal{A}}^i$ 的优化问题可以分解为关于 $\mathcal{A}^i$ 的 $|\mathcal{M}^i|$ 个优化问题。然后，我们将以下新的 alpha 向量 $\alpha^b\in\mathbb{R}^{\mathcal{S}^\ell}$ 添加到 $\mathcal{V}^\ell$：

$$\alpha^b(s^\ell) = \sum_{a^i\in\mathcal{A}^i}\gamma^{i,*}(a^i|m^i_{s^\ell})\alpha^*(s^\ell,a^i) \quad \forall s^\ell\in\mathcal{S}^\ell. \quad (12)$$

注 4：如果此时直接应用基于点的备份过程（2）–（4），则运算次数将为 $\Theta(|\mathcal{S}||\mathcal{V}||\bar{\mathcal{A}}^i|)$，由于 $|\bar{\mathcal{A}}^i| = |\mathcal{A}^i|^{|\mathcal{M}^i|}$，其将随 $|\mathcal{M}^i|$ 呈指数增长。相比之下，（8）–（12）中所列过程的运算次数为 $\Theta(|\mathcal{S}||\mathcal{V}^i||\mathcal{A}^i||\mathcal{M}^i|)$，关于所有涉及的参数均为多项式级。

注 5：对于固定的 α-向量，我们利用处方空间 $\bar{\mathcal{A}}^i$ 上优化问题的可分性来简化优化问题（7）。只有在每个阶段仅有一个智能体时才能实现这一点，这也是我们采用协调者的部分可观测马尔可夫决策过程扩展形式的原因之一。

阶段2：对于 $b\in\Delta(\mathcal{S}^2)$， b处下界的贝尔曼更新为

$$[T^2L^0](b) = r(b) + \beta\sum_{o\in\mathcal{O}}\mathbb{P}(o|b)\max_{\alpha\in\mathcal{V}^0}[\tau^2(b,o)]^T\alpha$$

$$= r(b) + \beta \sum_{o \in \mathcal{O}} \max_{\alpha \in \mathcal{V}^0} \sum_{s^2 \in \mathcal{S}^2} b(s^2) \sum_{s' \in \mathcal{S}} \mathbb{P}(s', o|s^2)\alpha(s')$$

其中 $r(b) := \sum_{(s,\ a) \in \mathcal{S} \times \mathcal{A}} b(s,\ a) r(s,\ a)$。

由于这一阶段没有动作，我们可以像在 (2)-(4) 中那样计算新的 alpha 向量 $\alpha^b \in \mathbb{R}^{\mathcal{S}^2}$（只是这里没有动作），即我们计算

$$\alpha^{b,o} := \arg\max_{\alpha \in \mathcal{V}^0} \sum_{s^2 \in \mathcal{S}^2} b(s^2) \sum_{s' \in \mathcal{S}} \mathbb{P}(s', o|s^2)\alpha(s') \quad \forall o \in \mathcal{O} \tag{13}$$

$$\alpha^b(s^2) := r(s^2) + \beta \sum_{o \in \mathcal{O}} \sum_{s' \in \mathcal{S}} \mathbb{P}(s', o|s^2)\alpha^{b,o}(s') \quad \forall s^2 \in \mathcal{S}^2 \tag{14}$$

下界更新算法见算法 4。

函数 L。Update(b):
  // L stores three sets of α-vectors: $\mathcal{V}^\ell, \ell = 0, 1, 2$
  设置 $\ell \in \{0, 1, 2\}$，使得 $b \in \Delta(\mathcal{S}^\ell)$;
  如果 $\ell < 2$，**则**
    使用 (8) – (12) 计算 α^b；
  else
    使用 (13)(14) 计算 α^b；
  将 α^b 添加到 $\mathcal{V}^\ell$;
  偶尔剪枝 $\mathcal{V}^\ell$ 中的被支配向量;

算法 4: 下界更新

B. 新的上界表示

在本节中，我们提出了一种新的部分可观测马尔可夫决策过程上界表示，称为基于 α约束的上界。我们寻找新的上界表示的动机，源于扩展协调器POMDP中的基于凸包的上界存在的一个缺陷。我们相信，我们的新上界可以缓解这一缺陷。无论如何，新的上界从不比原始上界更差（即更大）。

在 HSVI 中，上界函数在搜索启发式中发挥着非常重要的作用。它由一组孤立点 $(b,\ v^b) \in \Delta(\mathcal{S}) \times \mathbb{R}$ 表示。其他信念点处的上界则通过基于凸包的插值获得。尽管基于凸包的上界及其近似在许多单智能体部分可观测马尔可夫决策过程问题上取得了实证成功 [8], [7]，但由于以下原因，此类上界可能不足以用于扩展协调者部分可观测马尔可夫决策过程：对于定义在同一空间上的两个概率分布 b 和$\tilde{b}$，如果在任何表示 b 的凸组合中都不包含 $\tilde{b}$（即其系数为 0），我们就称 b 与 $\tilde{b}$ 不可表示。将动作纳入状态空间，会产生数量呈指数级增长、两两不可表示的信念。因此，除非算法探索这些信念，否则这些信念处的上界不会从其初始值改善（即变小）。由于这些算法采用基于乐观原则的探索启发式 [8], [7]，算法很可能会最终会对这些信念进行穷举采样，导致行为类似于暴力搜索。

为了更清楚地说明这一点，考虑一个具有全支撑的信念 $b_0 \in \Delta(\mathcal{S})$。接着，考虑两个处方$\gamma^1, \tilde{\gamma}^i \in \bar{\mathcal{A}}^1$。假设 $\gamma^1(m^1) = a^1 \neq \tilde{\gamma}^1(m^1) = \tilde{a}^1$对某个 $m^1 \in \mathcal{M}^1$成立。令 $b_1 = \tau^0(b_0, \gamma^1)$ 和 $\tilde{b}_1 = \tau^0(b_1, \tilde{\gamma}^1)$。令 $s \in \mathcal{S}$ 为某个满足 $m_s^1 = m^1$的状态。我们有 $b_1(s, a^1) > 0, \tilde{b}_1(s, a^1) = 0$。这意味着 $\tilde{b}_1$无法用 b_1 表示，反之亦然。因此，即使 b_1 非常接近 b_1 （当γ^1 仅在一个 $m^1 \in \mathcal{M}^1$上与 $\tilde{\gamma}^1$ 不同时，情况便是如此），在 b_1 处更新上界也不会影响 b_1处的上界$\tilde{}$。

为了解决这个问题，我们希望上界表示不仅紧致，还能促进更多交叉学习，也就是说，已知信念 $\tilde{b}$的上界 b有助于降低另一个信念的上界，即使 $\tilde{b}$ 无法用 b表示。此外，如果 b 和 $\tilde{b}$ 彼此接近，我们也希望它们的上界同样接近。

为设计一种新的上界表示，我们从凸包表示（5）的对偶形式出发：

$$U(b) = \max\{b^T y : y \in \mathbb{R}^{\mathcal{S}}, \mathbf{B}^T y \leq \bar{v}\}. \tag{15}$$

对（15）的直接解释如下：最优价值函数具有一个 α-向量表示 $V^*(b) = \max_{\alpha \in \mathcal{V}^*} \alpha^T b$，其中 $\mathcal{V}^*$ 表示（1）中定义的 $\mathcal{V}^{(k)}$ 的极限。最大化问题（15）为 $V^*(b)$提供了一个上界，因为 $\mathcal{V}^*$ 中的所有 α-向量都满足约束 $\mathbf{B}^T\alpha \leq \bar{v}$ （这是因为 $\bar{v}$ 是信念取值于$\mathbf{B}$时 V^* 的一个上界）。换言之，（15）可以视为一种基于 α-约束的上界，其中上界函数由一组线性约束构造而成，而集合 $\mathcal{V}^*$ 应满足这些约束。

基于这一思路，我们使用价值函数上界以外的其他类型的线性约束，例如，$\alpha(s) \geq v_{\min}$ where $v_{\min} := \min_{s \in \mathcal{S}, a \in \mathcal{A}} r(s, a)/(1 - \beta)$。将其添加到 (15) 中，我们得到

$$U(b) = \max\{b^T y : y \in \mathbb{R}^{\mathcal{S}}, \mathbf{B}^T y \leq \bar{v}, y \geq v_{\min}\mathbf{1}\}. \tag{16}$$

其上界不劣于 HSVI 中原先使用的上界。与（15）不同，上界（16）能够促进更多交叉学习：通过界定约束集的 L_∞直径，可以证明（16）是L_1-Lipschitz 连续的，常数为 $v_{\max} - v_{\min}$ （其中$v_{\max} := \max_{s \in \mathcal{S},\ a \in \mathcal{A}} r$（$s$，$a$）/（$1 - \beta$）），从而确保$\tilde{}$若 b 接近 b，则它们的上界也相对接近$\tilde{}$，无论 b 是否能够用 b 表示。有关更多类型的 α-约束，请参见附录B。

总之，任何被 $\mathcal{V}^*$ 中所有向量满足的有效线性不等式，都可以添加到 α-约束集合中。与基于凸包的界相比，这为我们提供了很大的灵活性。接下来，我们将描述扩展协调器的部分可观测马尔可夫决策过程中的 α-约束上界的更新。下一节介绍的更新方法独立于 α-约束的具体选择。

C. 上界更新

我们为三个阶段分别维护一个上界函数。按照上一节的描述，每个上界函数都由一组 α-约束表示。令 $\mathbf{M}^\ell y \le w^\ell$ 表示与阶段 ℓ 的 α-约束（关于 $y \in \mathcal{S}^\ell$）相对应的集合。令 T^ℓ 表示阶段 ℓ 的贝尔曼算子。现在，我们详细介绍更新步骤：

阶段 $\ell = 0, 1$：在此阶段，协调者为智能体 $i = \ell + 1$选择一个处方。对于 $b \in \Delta(\mathcal{S}^\ell)$，在 b处经过贝尔曼更新的上界为

$$\begin{aligned}\bar{v}^b &:= [T^\ell U^{\ell+1}](b) = \max_{\gamma^i \in \bar{\mathcal{A}}^i} U^{\ell+1}(\tau^\ell(b, \gamma^i)) \\ &= \max_{\gamma^i \in \bar{\mathcal{A}}^i} \max_{\substack{y \in \mathbb{R}^{\mathcal{S}^i} \\ \mathbf{M}^i y \le w^i}} \sum_{(s^\ell, a^i) \in \mathcal{S}^\ell \times \mathcal{A}^i} b(s^\ell)\gamma^i(a^i | m^i_{s^\ell}) y(s^\ell, a^i) \\ &=: \max_{\gamma^i \in \bar{\mathcal{A}}^i} \max_{\substack{y \in \mathbb{R}^{\mathcal{S}^i} \\ \mathbf{M}^i y \le w^i}} J^i(b, \gamma^i, y) \qquad (17)\end{aligned}$$

现在请注意，如果我们将每个 γ^i 视为 $[0,1]^{\mathcal{A}^i \times \mathcal{M}^i}$中的一个 0-1 指示向量，那么 $J^i(b, \gamma^i, y)$ 关于 γ^i 和y 是双线性的：对于固定的 y，它关于 γ^i 是线性的；对于固定的 γ^i，它关于 y 是线性的。因此，(17) 是一个双线性规划（BP）问题，优化领域文献 [27], [28], [29] 以及分散式控制领域文献都对此进行了广泛研究。 [30], [31]。值得注意的是，在 [30], 中，作者提供了一种将双线性规划程序转换为混合整数线性规划的方法[2]。Gurobi Optimization™ 还开发了用于双线性规划程序的专用求解器 [32]。

注释 6. 尽管双线性规划程序通常是 NP难的 [33]，但通过双线性规划程序对上界更新进行建模，相较于 HSVI 中的原始更新方法（即针对每个 $\gamma^i \in \bar{\mathcal{A}}^i$分别求解式 (17) 中的内层线性规划），仍有以下几项优势：(i) 它使我们能够应用双线性规划程序的系统化方法，避免穷举处方。(ii) 并非所有双线性规划程序都属于 NP难问题。(iii) 它为近似上界方法（例如对双线性规划程序的混合整数线性规划重构 [30] 进行某种松弛）打开了大门。

阶段2：对于 $b \in \Delta(\mathcal{S}^2)$，在信念 b 处的上界的贝尔曼更新为

$$\begin{aligned}\bar{v}^b &= [T^2 U^0](b) \\ &= r(b) + \beta \sum_{o \in \mathcal{O}} \mathbb{P}(o|b) \max_{\substack{y \in \mathbb{R}^{\mathcal{S}} \\ \mathbf{M}^0 y \le w^0}} [\tau^2(b, o)]^T y \qquad (18)\end{aligned}$$

这可以通过求解 $|\mathcal{O}|$线性规划来计算。

上界更新算法如算法5所示。

函数 U.Update(b)**:**
  // U stores three sets of α-constraints: $\mathcal{C}^\ell, \ell = 0,1,2$
  将 $\ell \in \{0,1,2\}$ 设为使 $b \in \Delta(\mathcal{S}^\ell)$ 成立；
  如果 $\ell < 2$ **那么**
    计算 $\bar{v}^b, \gamma^{i,*}$，最优值和双线性规划程序 (17) 的优化器；
  else
    用 (18) 计算 $\bar{v}^b$；
    将 $\gamma^{i,\ *}$ 设为表示空处方；
  将 $b^T y \le \bar{v}^b$ 添加到 $\mathcal{C}^\ell$；
  偶尔剪除冗余的 α-约束；
  返回 $\gamma^{i,\ *}$；

算法 5： 上界更新

IV. 实验结果

我们使用 Python（`https://github.com/dwtang/chsvi`）实现了 CHSVI算法。算法中涉及的所有 BP 和 LP 均使用 Gurobi Optimization Studio™ 求解。超参数 ζ 设为 0。85。当当上界与下界之间的差距小于 0.01，或运行时间超过 24 小时时，算法终止。下界采用与 [8] 相同的方式，通过固定动作界进行初始化。上界中使用的 α-约束初始化为边际信念约束（见附录B.2），这些约束由一个松弛的部分可观测马尔可夫决策过程问题得出，其中假设所有私有信息都是共有的。我们采用与 HSVI 相同的剪枝策略：对于下界，只剪除被其他向量逐点支配的 α-向量。对于上界，通过求解线性规划移除冗余的 α-约束。

我们在附录A.1和A.2中分别定义的 `DecTiger` 和 `MultiCast`实例上运行该算法。实验使用一台配备 Intel Xeon® E3-1231 v3 CPU（4 核，3.4Ghz）和 16GB RAM 的计算机进行。结果如表 III 和图 1 所示，附录 C 中还提供了其他图表。除 `DecTiger`(3,1,0.99)外，在所有实例中，该算法都能消除上界与下界之间的差距，并找到近乎最优的策略。据我们所知，除存在解析解的情况外，这些是针对无限时域多智能体控制问题的首批经证明的最优解。

	$L(b_0)$	$U(b_0)$	时间（秒）
`DecTiger` (2,1,0.9)	32.7704	32.7792	56
`DecTiger` (2,1,0.99)	388.4035	388.4134	2081
`DecTiger` (3,1,0.9)	6.7139	6.7236	63781
`DecTiger` (3,1,0.99)	76.2553	222.2532	86483
`MultiCast` (8,8,0.1,0.2,0.9)	-4.8550	-4.8450	346
`MultiCast` (16,16,0.2,0.4,0.9)	-10.0653	-10.0400	86423

表 I
实验结果。

尽管算法花费了数小时才将 `MultiCast`个实例的上界与下界之间的差距缩小到0.01，但我们想指出，事后看来，如图2所示，下界很快就达到了近似最优值。算法的其余部分主要是降低上界，以提供最优性保证。请记住，CHSVI是一种随时可用算法。这意味着，即使

[2]该方法仅适用于BP中两组变量的约束彼此分离的情况。(17)中的优化问题确实属于此类。

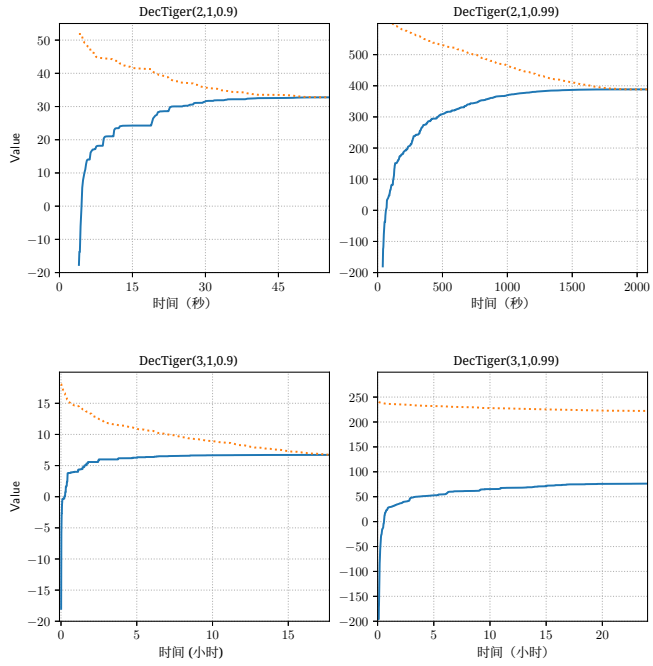


图 1. DecTiger (N, d, β) 上的实验结果。虚线表示上界 $U(b_0)$，实线表示下界 $L(b_0)$。

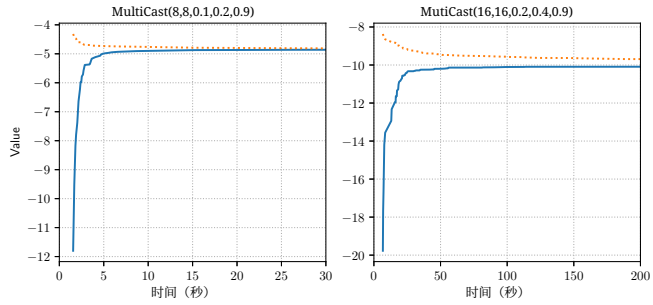


图 2. MultiCast (C^1, C^2, p^1, p^2, β) 上的实验结果。虚线表示上界 $U(b_0)$，实线表示下界 $L(b_0)$。这里只展示了算法开始阶段。下界很快达到接近最优的值。在其余运行时间内，上界和下界的变化都非常缓慢。

如果我们提前终止算法，仍然可以获得性能接近最优的策略。

该算法耗时超过 18 小时才缩小了 DecTiger (3,1,0.9) 的差距。不过，事后看来，可以观察到下界相对较快地达到接近最优的值（运行 3 小时时，下界已达到 5.9919，与最终值 6.7236 相差不远）。对于 DecTiger(3,1,0.99)，运行 24 小时时差距仍然很大。不过，我们推测，24 小时报告的解已经接近最优，且随着运行时间增加，差距会继续缩小。

我们想指出的是，在我们的计算机上，直接将现成的部分可观测马尔可夫决策过程求解器应用于我们所使用的任何实例都是不可能的：我们曾尝试应用最先进的 HSVI2 求解器 [8]，但 MultiCast(5,5,0.1,0.2,0.9)，该问题有 36个状态、4个观测以及 2^{12} 个动作/处方。程序仅初始化求解器就花费了 1个多小时（在此阶段，如果终止算法，算法不会输出任何解）。相比之下，我们求解的实例规模要大得多：我们实例的协调器的部分可观测马尔可夫决策过程的状态数、观测数和动作/处方数列于表 II 中。鉴于 HSVI2 算法初始化阶段的复杂度（该阶段使用FastInformed Bound 方法 [8]计算上界）与动作/处方数呈线性关系，我们预计，对于 MultiCast (8,8,0.1,0.2,0.9)，仅初始化 HSVI2 求解器就至少需要 2.7 天，而对于其他任何实例则需要 194 天。

	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{O}\rvert$	$\lvert\bar{\mathcal{A}}\rvert$
DecTiger (2,1,β)	74	37	3^{14}
DecTiger (3,1,β)	435	145	4^{26}
MultiCast (8,8,0.1,0.2,0.9)	64	4	2^{18}
MultiCast (16,16,0.2,0.4,0.9)	256	4	2^{34}

表 II
协调器的部分可观测马尔可夫决策过程的维度。

V. 结论

由于信息不对称，分散式随机控制问题可能比其集中式对应问题困难得多。尽管某些分散式控制问题可以通过CI方法转换为等价的集中式问题，但由于动作数量极其庞大，这些集中式问题仍然难以求解。在本工作中，我们提出了协调者启发式搜索值迭代（CHSVI）算法，该算法结合CI方法与基于点的部分可观测马尔可夫决策过程方法，以求解多智能体随机控制问题。与直接对协调器的部分可观测马尔可夫决策过程使用基于点的算法相比，我们的算法能够更加高效地求解多智能体控制问题（见备注1和2）。此外，我们的方法还提出了若干可立即开展的未来研究方向，以进一步提高可扩展性，例如：（1）用于CHSVI的近似上界更新方法；（2）将CI方法与其他基于点的算法（如SARSOP[7]）相结合；（3）为上界表示初始化 α-约束的高效方法。

参考文献

[1] A. Nayyar、A. Mahajan 和 D. Teneketzis，《具有部分历史共享的去中心化随机控制：一种公共信息方法》，*IEEE* 自动控制汇刊，第 58 卷，第 7 期，第 1644–1658 页，2013 年。[2] E. J. Sondik，《无限时域部分可观测马尔可夫过程的最优控制：折扣成本》，运筹学，第 26 卷，第 2 期，第 282–304 页，1978 年。[3] A. Cassandra、M. L. Littman 和 N. L. Zhang，《增量剪枝：一种简单、快速且精确的部分可观测马尔可夫决策过程方法》，载于第 *13* 届 *UAI* 会议论文集，1997 年，第 54–61 页。[4] G. Shani、J. Pineau 和 R. Kaplow，《基于点的部分可观测马尔可夫决策过程求解器综述》，自主智能体与多智能体系统，第 27 卷，第 1–51 页，2013 年。[5] J. Pineau、G. Gordon 和 S. Thrun，《适用于大型部分可观测马尔可夫决策过程的任意时长基于点的近似方法》，人工智能研究期刊，第 27 卷，380，2006 年，第 335 页。

[6] T. Smith 和 R. Simmons，《部分可观测马尔可夫决策过程的启发式搜索价值迭代》，载于第 20 届 UAI 会议论文集，2004 年。[7] H. Kurniawati、D. Hsu 和 W. S. Lee，《SARSOP：通过近似最优可达信念空间实现高效的基于点的部分可观测马尔可夫决策过程规划》，载于机器人科学与系统会议论文集，2008 年。[8] T. Smith 和 R. Simmons，《基于点的部分可观测马尔可夫决策过程算法：改进的分析与实现》，载于第 21 届 UAI 会议论文集，2005 年。[9] A. Nayyar、T. Bas¸ar、D. Teneketzis 和 V. V. Veeravalli，《具有能量采集传感器的通信与远程估计的最优策略》，IEEE 自动控制汇刊，第 58 卷，第 9 期，第 2246–2260 页，2013 年。[10] A. Mahajan，《具有控制共享的耦合子系统的最优去中心化控制》，IEEE 自动控制汇刊，第 58 卷，第 9 期，第 2377–2382 页，2013 年。[11] H. Tavafoghi、Y. Ouyang 和 D. Teneketzis，《非对称信息动态决策问题的统一方法：非策略性智能体》，IEEE 自动控制汇刊，第 67 卷，1119，2022 年，第 3 期，第 1105 页。[12] J. Subramanian、A. Sinha、R. Seraj 和 A. Mahajan，《部分可观测系统中近似规划与强化学习的近似信息状态》，机器学习研究期刊，第 23 卷，1，2022 年，12 页。[13] K. Zhang、E. Miehling 和 T. Bas¸ar，《具有部分历史共享的去中心化随机控制的在线规划》，载于 ACC 会议论文集，2019 年，第 3544–3550 页。[14] H. Kao 和 V. Subramanian，《多智能体强化学习中基于公共信息的近似状态表示》，载于 AISTATS 会议论文集。PMLR，2022 年，第 6947–6967 页。[15] J. Foerster、F. Song、E. Hughes、N. Burch、I. Dunning、S. Whiteson、M. Botvinick 和 M. Bowling，《深度多智能体强化学习的贝叶斯动作解码器》，载于 ICML 会议论文集，2019 年。[16] C. Amato、G. Chowdhary、A. Geramifard、N. K. ¨Ure 和 M. J. Kochenderfer，《部分可观测马尔可夫决策过程的去中心化控制》，载于 IEEE CDC 会议论文集，2013 年，第 2398–2405 页。[17] A. Kumar 和 S. Zilberstein，《去中心化部分可观测马尔可夫决策过程的基于点的备份：复杂度与新算法》，载于第 9 届国际自主智能体与多智能体系统会议（AAMAS），2010 年。[18] L. C. MacDermed 和 C. L. Isbell，《具有最优信念压缩的去中心化部分可观测马尔可夫决策过程基于点的值迭代》，神经信息处理系统进展，第 26 卷，2013 年。[19] J. S. Dibangoye、C. Amato、O. Buffet 和 F. Charpillet，《将去中心化部分可观测马尔可夫决策过程作为连续状态马尔可夫决策过程进行最优求解》，人工智能研究期刊，第 55 卷，第 443–497 页，2016 年。[20] J. Pajarinen 和 J. Peltonen，《用于高效部分可观测马尔可夫决策过程及去中心化部分可观测马尔可夫决策过程规划的周期性有限状态控制器》，NIPS 会议论文集，2011 年。[21] J. S. Dibangoye、O. Buffet 和 F. Charpillet，《无限时域折扣型去中心化部分可观测马尔可夫决策过程的误差有界近似》，载于 ECML PKDD 会议论文集，2014 年，第 338–353 页。[22] S. Adhikari 和 P. Gmytrasiewicz，《通信型交互式部分可观测马尔可夫决策过程的基于点的求解方法》，载于欧洲多智能体系统会议（EUMS）论文集，2021 年，第 245–263 页。[23] N. L. Zhang 和 W. Zhang，《加速部分可观测马尔可夫决策过程值迭代的收敛》，人工智能研究期刊，第 14 卷，第 29–51 页，2001 年。[24] M. Hauskrecht，《部分可观测马尔可夫决策过程的价值函数近似》，人工智能研究期刊，第 13 卷，第 33–94 页，2000 年。[25] M. T. Spaan 和 N. Vlassis，《Perseus：部分可观测马尔可夫决策过程的随机化基于点的值迭代》，人工智能研究期刊，第 24 卷，220，2005 年，第 195 页。[26] P. Poupart、K.-E. Kim 和 D. Kim，《缩小差距：改进部分可观测马尔可夫决策过程最优解的界》，载于 ICAPS 会议论文集，第 21 卷，2011 年，第 194–201 页。[27] G. Gallo 和 A. ¨Ulk¨uc¨u，《双线性规划：一种精确算法》，数学规划，第 12 卷，第 1 期，第 173–194 页，1977 年。[28] H. Konno，《求解双线性规划程序的割平面算法》，数学规划，第 11 卷，第 1 期，第 14–27 页，1976 年。[29] R. Cogill 和 S. Lall，《离散团队决策问题的近似算法》，SIAM 控制与优化期刊，第 45 卷，第 4 期，第 1359–1368 页，2006 年。[30] M. Petrik 和 S. Zilberstein，《平均回报去中心化马尔可夫决策过程》，载于 IJCAI 会议论文集，2007 年，第 1997–2002 页。[31] ——，《多智能体规划的双线性规划方法》，人工智能研究期刊，第 35 卷，第 235–274 页，2009 年。

[32] “非凸二次优化”，https://www.gurobi.com/events/non-convex-quadratic-optimization-2/，访问日期：03/21/2023。[33] J. Tsitsiklis 和 M. Athans，“去中心化决策与检测问题的复杂性”，*IEEE* 自动控制汇刊，第 30 卷，第 5 期，第 440–446 页，1985 年。[34] R. Nair、M. Tambe、M. Yokoo、D. Pynadath 和 S. Marsella，“驯服去中心化部分可观测马尔可夫决策过程：面向多智能体环境的高效策略计算”，载于*IJCAI* 会议论文集，2003 年。

附录

A. 实验环境

在本节中，我们介绍用于第四节实验结果的环境。

1)DecTiger：我们考虑带有三个参数（N、d、β）的 DecTiger [34], [19] 问题的参数化扩展：两个智能体面对着 N关闭的门。其中一扇门后面是一只凶猛的老虎，而所有其他门后面都是装有老虎形状的饼干的盒子。在每个时间步，每个智能体都有 $N+1$ 种选择：打开其中一扇门，或聆听老虎的吼声。然而，聆听需要付出代价，并且容易出错：智能体以概率 $p = 0.85/(0.7 + 0.15N)$听到来自正确门的吼声，以概率 $q = 0.15/(0.7 + 0.15N)$听到来自其他任意一扇门的吼声。智能体获得的即时奖励列于表 III中。折扣因子为 β。初始时，老虎被放置在均匀随机选择的一扇门后面。如果两个智能体都选择聆听，世界状态不会发生变化。如果任一智能体打开了任意一扇门，那么老虎会将自己重新置于均匀随机的位置。如果两个智能体都选择聆听，那么在下一个时间步，它们将分别获得条件独立的观测，且每个观测都遵循上述分布。如果任一智能体打开了任意一扇门，那么无论另一个智能体采取何种行动，所有观测都将彼此独立地服从均匀随机分布，并且与老虎的实际位置无关。

$a^1 \backslash a^2$	倾听	打开老虎所在的门	打开另一扇门
倾听	-2	-101	$\frac{20}{N} - 1$
打开老虎所在的门	-101	-50	-100
打开另一扇门	$\frac{20}{N} - 1$	-100	$\frac{40}{N}$

表 III

DecTiger(N, d, β) 的即时奖励

在原始设定中 [34], [19], ，智能体的动作和观测都是私有信息，这符合分散式部分可观测马尔可夫决策过程模型的传统做法。在本研究中，我们考虑该问题的一种延迟信息共享模型：智能体彼此共享动作和观测，但会延迟 $d > 0$。更具体地说，在时间 t，每个智能体都可以获取其他智能体在 $t-d$ 之前（不包括该时刻）采取的动作，以及这些动作产生的观测。对于智能体 i，在时间 t，其在时间 $t-d$ 及之后采取的动作及其产生的观测均为私有信息。

DecTiger(N, d, β) 可使用我们在第 II-A.1 节中介绍的多智能体控制模型建模如下：$\mathcal{S} = [N]$ 表示老虎可能所在的位置。$\mathcal{A}^i = [N+1]$ 表示智能体 i 在任意给定时间的选择。令 $\mathcal{Z}^i = [N]$ 表示智能体 i 和 $\mathcal{Z} = \mathcal{Z}^1 \times \mathcal{Z}^2$ 的观测空间。那么 $\mathcal{M}^i = \{\emptyset\} \cup \bigcup_{k=1}^{d} (\mathcal{Z}^i \times \mathcal{A}^i)^k$ 是

智能体 i 尚未被其他智能体知晓的动作和观测空间。（符号 $\emptyset$ 表示在开始时不存在此类动作和观测。）公共观测空间 $\mathcal{O} = \{\emptyset\} \cup (\mathcal{Z} \times \mathcal{A})$ 是新共享的动作和观测的空间，即 d步之前的动作和观测。（符号 $\emptyset$表示智能体共享的第一条信息尚未到达。）构成模型 $\mathcal{E}$其余的元组（即 b_0，$\mathbb{P}$，r）可以自然地定义。

2)MultiCast：我们考虑一种受 [10] 和 [20]启发的双智能体多址广播系统模型。该模型由（C^1、C^2、p^1、p^2、β）参数化。每个智能体 i 都有一个容量为 C^i的缓冲区。初始时，缓冲区为空。在每个时间 t，数据包以概率p^i到达智能体 i的缓冲区，且该事件独立于所有历史以及另一个智能体的数据包到达事件。如果缓冲区已满，则丢弃该数据包，智能体需支付 $c_D = 2$的惩罚。否则，将数据包放入缓冲区。两个用户都可以尝试通过共享广播信道传输数据包。如果只有一个智能体尝试传输，则数据包成功传输并从缓冲区中移除。如果两个智能体同时尝试传输，则由于碰撞，两个智能体的传输均失败。尝试传输无论成功与否，均需支付 $c_T = 0.5$的成本。除丢弃惩罚和传输成本外，每个数据包在缓冲区中停留一个时间单位，智能体还需支付1的惩罚。惩罚和成本的折扣因子为 β。智能体知道自己的数据包到达历史，但不知道其他智能体的历史。一个智能体还可以在另一个智能体采取行动后观察其行动。

可以证明，在该模型中，每个智能体缓冲区中的数据包数量是用于决策的充分私有信息。因此，根据备注 1，`MultiCast` (C^1，C^2，p^1，p^2，β)可以按如下方式，使用第 II-A.1 节中的多智能体控制模型进行建模：$\mathcal{S}^i = \{0, 1, \cdots, C^i\}$ 是智能体 i 的私有状态集合，表示其缓冲区中的数据包数量。$\mathcal{S} = \mathcal{S}^1 \times \mathcal{S}^2$ 是状态空间。$\mathcal{A}^i = \{\text{NT}, \text{T}\}$ 表示智能体 i 在任意给定时间的两种选择。$\mathcal{M}^i = \mathcal{S}^i$ 是充分私有信息空间。$\mathcal{O} = \mathcal{A}$ 是公共观测空间。即时奖励 r定义为

$$r(s, a) = \sum_{i=1}^{2} r^i(s^i, a^i)$$
$$r^i(s^i, a^i) = -s^i - c_D \cdot p^i \mathbf{1}_{\{s^i = C^i\}} - c_T \cdot \mathbf{1}_{\{a^i = \text{T}\}}$$

构成模型的其余元组 $\mathcal{E}$（即 b_0，$\mathbb{P}$）可以自然地定义。

B. 关于新的上界表示的更多内容

在本节中，我们将进一步介绍第 III-B 节所述上界表示的更多类型的 α-约束。

1) 基于下界的约束：对于任意信念 b，我们可以通过考虑一个奖励最小化的部分可观测马尔可夫决策过程，推导出 $\min_{\alpha \in \mathcal{V}^*} \alpha^T b$ 的下界 v^b。然后，我们可以将不等式 $b^T y \geq v^b$ 添加到 α-约束集合中。A

这类粗略下界可以通过快速信息下界方法 [24] 高效计算。

2) 基于边缘信念的约束：通常，我们可以根据 b 的某个边缘分布推导出 $V^*(b)$ 的一个上界；也就是说，假设 $\mathcal{S} = \mathcal{K}^1 \times \mathcal{K}^2$，则对于所有 $b \in \Delta(\mathcal{S})$，都有 $V^*(b) \leq \bar{v}(\varphi(b))$，其中$\varphi(b) \in \Delta(\mathcal{K}^1)$ $[\varphi(b)](k^1) = \sum_{k^2 \in \mathcal{K}^2} b(k^1, k^2)$。一个例子是通过求解一个松弛问题，推导出协调器的部分可观测马尔可夫决策过程的上界；在该问题中，假设部分或全部私有信息是共同已知的。（由于私有信息空间和增广状态空间更小，这个松弛问题可能更容易求解。）这类上界可以很容易地转化为 α-约束：假设在固定的 $b^1 \in \Delta(\mathcal{K}^1)$ 处有一个上界 $\bar{v}^{b^1}$，那么我们知道，$\mathcal{V}^*$ 中所有的 α-向量都满足该约束

$$\max_{b:\varphi(b)=b^1} b^T y \leq \bar{v}^{b^1}. \tag{19}$$

我们可以通过以下步骤将(19)转换为一组有限个线性约束：首先注意到

$$\max_{b:\varphi(b)=b^1} b^T y = \sum_{k^1 \in \mathcal{K}^1} b^1(k^1) \max_{k^2 \in \mathcal{K}^2} y(k^1, k^2).$$

因此，通过添加辅助向量 $\bar{y} \in \mathbb{R}^{\mathcal{K}^1}$，我们得到 α-约束：

$$\begin{aligned} &y(k^1, k^2) \leq \bar{y}(k^1) \quad \forall k^1 \in \mathcal{K}^1, k^2 \in \mathcal{K}^2, \\ &(b^1)^T \bar{y} \leq \bar{v}^{b^1}. \end{aligned} \tag{20}$$

如果奖励最小化问题的下界可以表示为边缘信念的函数，则也可以用类似的方法求得相应的α-约束。假设对于所有满足$\varphi(b) = b^1$的 b，v^{b^1} 都是信念 b 下最小总奖励的下界，那么我们可以使用辅助向量$y \in \mathbb{R}^{\mathcal{K}^1}$ 来构造 α-约束：

$$\begin{aligned} &y(k^1, k^2) \geq \underline{y}(k^1) \quad \forall k^1 \in \mathcal{K}^1, k^2 \in \mathcal{K}^2, \\ &(b^1)^T \underline{y} \geq \underline{v}^{b^1}. \end{aligned} \tag{21}$$

C. 实验结果的补充图

在 CHSVI算法中，内存使用主要来自两个方面：(1) 用于表示下界的 α-向量，以及 (2) 用于表示上界的 α-约束。我们记录了算法每次运行期间 α-向量和 α-约束[3]的数量。结果如图 3 和图 4 所示。锯齿状模式是由于定期剪枝 α-向量和 α-约束所致。

与单智能体部分可观测马尔可夫决策过程中的 HSVI 一样，剪枝过程对于降低算法的计算和内存复杂度至关重要。它对于降低时间复杂度也发挥着重要作用，因为阶段

[3] α-关联辅助变量与主变量的约束，即 (20)(21)，未包括在内。

含有较少 α 向量或较少 α 约束的优化问题更容易求解。从图中可以看出，在所有情况下， α 向量和 α 约束中的大多数都属于阶段 2。这是意料之中的，因为阶段 2 的状态空间在三个阶段中最大，因此信念空间的维度也更高。状态空间越大，每个 α 向量或 α 约束所占用的内存也越多，因此阶段 2 α 向量和 α 约束占用了大部分内存。在 DecTiger(2,1,0.99)和 DecTiger(3,1,0.9)中，我们观察到， α 向量和 α 约束的数量增长速率逐渐降低，并在算法终止前很久就趋于稳定。我们推测，如果给予足够的运行时间， DecTiger(3,1,0.99)也会呈现相同的行为。

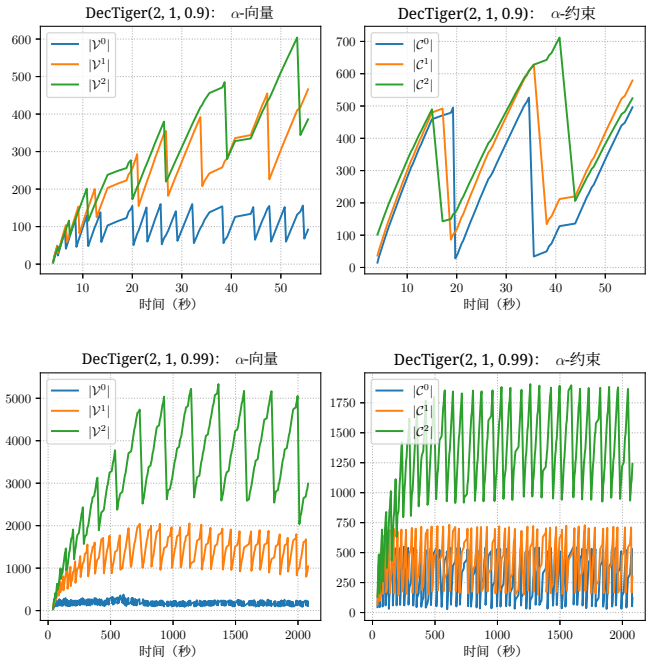


图 3. DecTiger $(2,1,\beta)$ 的 α-向量和 α-约束的数量。$\mathcal{V}^\ell$ 是算法 4 中定义的阶段 ℓ 的 α-向量集合。 $\mathcal{C}^\ell$ 是算法 5 中定义的阶段 ℓ 的 α-约束集合。

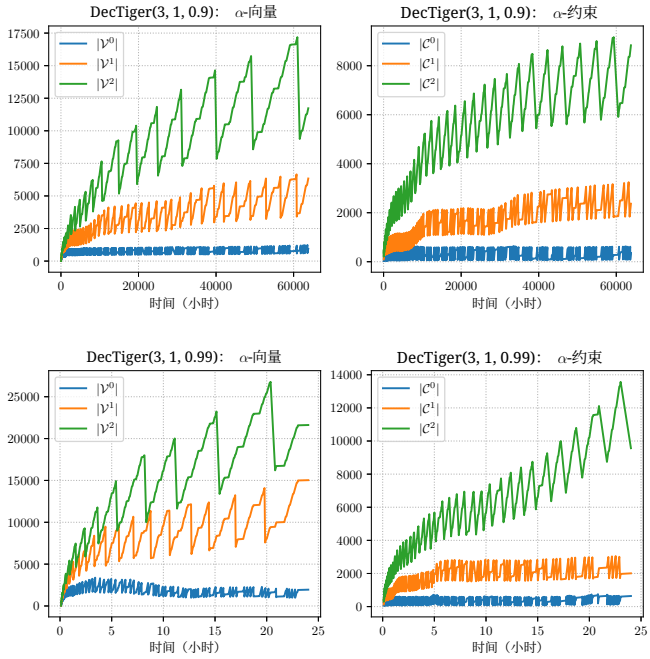


图 4. DecTiger $(3,1,\beta)$ 的 α-向量和 α-约束的数量。$\mathcal{V}^\ell$ 是算法 4 中定义的阶段 ℓ 的 α-向量集合。 $\mathcal{C}^\ell$ 是算法 5 中定义的阶段 ℓ 的 α-约束集合。