arXiv:2601.02505v1 [cs.RO] 5 Jan 2026

时间与资源约束下时空任务分配效能的学习与优化

期刊标题 XX(X)：1–17©作者 2025年 重印与许可：
sagepub.co.uk/journalsPermissions.nav DOI：
10.1177/ToBeAssigned
www.sagepub.com/SAGE

家珍 Liu[1], Glen Neville[1,2],Jinwoo Park[1], Sonia Chernova[1], Harish Ravichandar[1]

摘要
复杂的多机器人任务通常需要异构团队共同优化任务分配、调度和路径规划，以在严格约束下提升团队性能。我们将这些复杂性形式化为一类新问题，称为多机器人时空效能优化分配（*STEAM*）。STEAM 基于基于特质的框架，利用机器人的能力（例如载荷和速度）对其进行建模；但它超越了典型的二元成功—失败模型，明确将分配效能建模为特质效能映射。这些映射编码了分配给某项任务的综合能力如何决定其性能。此外，STEAM 还支持时空约束，包括用户指定的时间预算（即最大完工时间）。为解决 STEAM 问题，我们提出了一种名为效能优化增量式任务分配图搜索（*E-ITAGS*）的新算法，该算法同时优化任务性能并遵守时间预算，方法是交错执行任务分配、调度和路径规划。鉴于特质效能映射难以、甚至不可能预先指定，E-ITAGS 通过可实现性感知主动学习模块高效地学习这些映射。我们的方法具有可实现性感知能力，因为它明确考虑到在学习过程中，并非所有特征组合都能由现有机器人实现。此外，我们还推导出经实验验证的 E-ITAGS 效能次优性界。基于应急响应领域的详细数值仿真和实验表明，与基线方法相比，E-ITAGS 能够生成效能更高的分配，同时满足资源和时空约束。我们还表明，我们的主动学习方法具有较高的样本效率，并在数据效率与计算效率之间建立了有原则的权衡。

关键词
多机器人系统，时空任务分配，主动学习

1 引言

异构多机器人系统（MRS）汇集互补能力，以应对农业（Rold´an 等，2016年）、国防（McCook 和 Esposito，2007年）、装配（Stroupe 等，2005年）和仓库自动化（Baras 和 Dasygenis，2019年等不同领域的挑战。要在这些复杂领域实现有效协作，通常需要对任务分配（由谁执行）（Ravichandar 等人，2020年）、调度（何时执行）（Matos 等，2021年）、运动规划（如何执行）（Baras 和 Dasygenis，2019年）以及此类时空问题的交叉领域（Neville 等，2021；Messing 等，2022）进行推理。

在本研究中，我们首先对一类新颖的异构多机器人时空协调问题进行形式化定义，并将其称为 时空效能优化的多机器人系统分配（*STEAM*）。STEAM 问题与成熟的多机器人任务分配（MRTA）分类体系（Gerkey 和 Matari´2004年；Nunes 等人 2017年）所定义的 ST-MR-TA 问题密切相关（详见第 2.1 节）。为有效解决 STEAM 问题，我们进一步开发了效能优化增量式任务分配图搜索（*E-ITAGS*）*，这是一种由启发式驱动的交错式搜索算法（见图 1），专为在异构团队时间预算约束下运行而设计。

我们研究并改进了 STEAM 类问题的三个关键方面：i）联盟级任务的可分解性，ii）机器人能力对任务性能的集体影响，以及 iii）任务规范。下面，我们从这三个角度简要介绍我们的方法，并结合现有方法进行阐述。有关相关工作的详细讨论，以及不同方法所采用的具体假设子集，请参阅第2 节。

首先，有几种方法假设可以将多机器人任务分解（Giordani 等人，2013年；Krizmancic 等人，2020年）为单个机器人子任务，或枚举所有可能的联盟（Gombolay 等人，2016年；Schillinger 等人，2018年）。然而，对于涉及复杂协作的任务，明确规定每个机器人的角色往往很困难，甚至是不可能的；而枚举所有具备能力的联盟也可能难以实现。相比之下，基于我们此前在基于特质的任务分配方面的工作（Ravichandar 等人，2020年；Neville 等，2021年；Messing 等人，2022年；Srikanthan 和 Ravichandar，2022年；Neville 等人，2023年），我们从能力的角度对机器人进行建模，并以集体能力来描述

[1]佐治亚理工学院，美国；[2]Zoox，美国

通讯作者：刘家珍，佐治亚理工学院，美国。电子邮件：jliu3103@gatech.edu

*开源地址：https://github.com/GT-STAR-Lab/Q-ITAGS

任务需求，从而形成一个灵活的且与机器人无关的框架。因此，E-ITAGS既不需要分解任务，也不需要枚举所有可能的联盟。

其次，现有方法往往假设任务结果遵循二元成功或失败模型（Fu 等人，2022b；Messing 等人，2022；Srikanthan 和 Ravichandar，2022）。这种有限的模型不适用于若干现实问题，因为这些问题要求最大化性能指标，而非满足任意阈值（例如，分布式感知、供应链优化和灾害响应）。相比之下，我们提出了特质效能映射，这是一种新颖且富有表现力的连续模型，可将集体能力（特征）映射到任务性能（效能），有助于对分配的效能进行编码和优化。

第三，现有方法通常要求用户明确指定机器人能力与任务成功完成之间的关系（Prorok 等人，2017；Ravichandar 等人，2020；Mayya 等人，2021；Neville 等人，2023）。然而，众所周知，人类虽然擅长做出复杂决策，却往往无法明确阐述自己如何做出决策；事实上，我们往往会编造理由来为自己的决定辩护（Rieskamp 等人，2003；Rieskamp 和 Otto，2006）。相比之下，E-ITAGS 不要求用户指定集体能力与任务性能之间的关系。它采用数据驱动的主动学习模块，以高效学习这些影响关系，并将其编码在特质效能映射中。

E-ITAGS 通过交错搜索任务分配与调度方案来求解 STEAM 问题。这种交错范式已被证明能够同时优化任务分配、调度和路径规划，而不会造成难以承受的计算开销（Messing 等人，2022）。为引导 E-ITAGS 的搜索，我们引入了两种新的启发式方法：i）时间预算超限（*TBO*），用于衡量给定调度方案相对于用户指定的完工时间阈值的次优程度；ii）归一化分配成本（*NAC*），用于提供当前分配效能的归一化估计（由特质-效能映射预测）。E-ITAGS 将这两种启发式方法进行凸组合，并提供一个超参数，使用户能够权衡完工时间与任务性能之间固有的取舍。

在现实世界中学习特质效能映射时，看似细微的因素可能会严重限制学习。请注意，可分配给特定任务的集体能力，直接取决于学习期间可用的单个机器人的能力，并受其限制。例如，若一个团队由三台载荷分别为 5 、 10 和 15 磅的机器人组成，就不可能组建并评估一个集体载荷恰好为 8 磅或 18 磅的联盟。一般而言，给定团队能够实现的所有集体能力，恰好构成由各个体能力按二元权重（0 或 1）加权组合而成的集合。我们将这一重要限制称为可实现性，并在研究中明确予以考虑。我们提出了多种考虑可实现性的学习策略，以不同程度的近似方式纳入可实现性，并证明这些策略在数据效率与计算效率之间建立了有原则的权衡。

除了上述经验性优势外，我们还对 E-ITAGS 的运行机制提供了理论见解。我们推导出一个在温和假设下，以分配效能为依据的 E-ITAGS 次优性界限。值得注意的是，我们的分析揭示了分配效能与调度总时长之间固有的权衡关系，而这一关系取决于单个超参数。这一见解和分析可为用户选择超参数提供直观指导。我们还通过实验表明，该界限在实践中始终成立。

总而言之，我们的核心贡献包括

- 一种时空多机器人任务分配问题的形式化描述，强调在遵守分配效能 优化的同时满足时间预算；
- 一种由两种新颖启发式方法引导的交错式 图搜索算法，用于求解此类问题；
- 一个可实现性感知主动学习 模块，能够有效学习未知的特质-效能映射，并平衡数据效率与计算效率；
- 针对分配效能，为我们的方法提供了一个次优界 。

我们从以下方面对 E-ITAGS 进行了全面评估：i）优化任务分配效能，ii）遵守时间预算，以及 iii）学习集体能力与任务性能之间的未知映射。我们既在数值模拟中，也在 RoboCup Rescue（Kitano and Tadokoro 2001；Kitano 等 1999；Federation 2025年）这一自主搜索与救援模拟器中进行了评估。结果明确表明，在生成完工时间相近的调度方案时，E-ITAGS 在分配效能方面优于现有方法。我们还详细分析了 E-ITAGS 为寻找有效任务分配而采用的增量搜索行为。此外，E-ITAGS 的可实现性感知主动采样方法在保持数据效率的同时，显著降低了计算成本。

本文是对 Neville 等 (2024年发表的会议论文的扩展和修订版本。主要新增内容和改进包括：i）可实现性感知主动学习方法，该方法考虑给定团队的资源约束；ii）识别并处理在将主动学习应用于受约束的多机器人协调时出现的特有问题；iii）新增两组实验，在一个开源机器人学模拟器中开展搜索与救援任务，以验证我们整体方法的有效性。为便于查阅，我们在第 5节介绍 i）和 ii）的算法细节，并在第 6.4 节和第 6.5节介绍 iii）的实验结果。

下文将回顾相关的既有工作，见第 2节。第3 节给出了 STEAM 问题的形式化定义，并介绍主动学习在其中的作用。我们在第 4节介绍 E-ITAGS，并在第 4.3节推导次优界。我们在第 5节介绍可实现性感知主动学习方法。第6 节讨论所有实验及其结果。最后，我们在第7节总结结论并讨论未来工作。

2 相关工作

本节综述相关工作，主要围绕两大支柱展开：多机器人任务分配和多机器人系统主动学习。

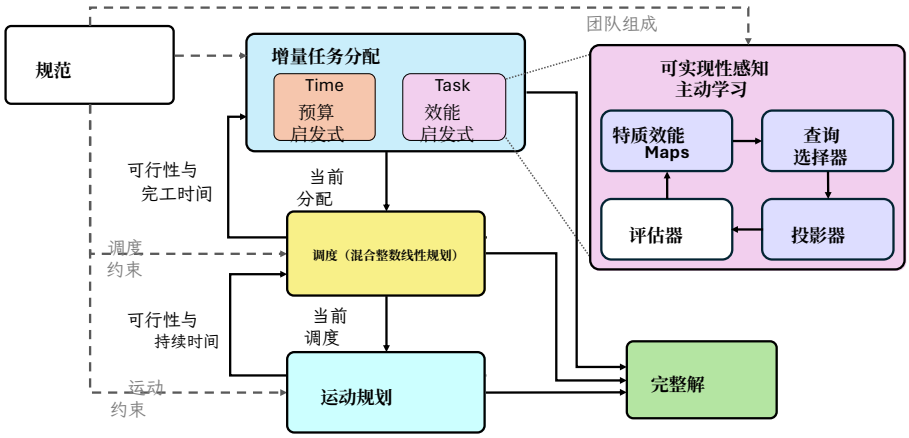


图1. E-ITAGS通过优化集体性能，在遵守时空约束和资源约束的同时，为异构多机器人团队执行时空任务分配。可实现性感知主动学习模块对特质效能映射进行显式建模、主动学习和优化，以近似集体能力对任务性能的影响。更多详情见第 4 节和第 5节，以及图 2 和 3。

2.1 多机器人任务分配

尽管多机器人任务分配问题有许多变体（Gerkey 和 Matari´c 2004年；Nunes 等人 2017年），但我们仅讨论一种与 STEAM 问题密切相关的变体：单任务（ST）机器人、多机器人（MR）任务以及时间延展型（TA）分配。ST-MR-TA 问题要求在满足时间约束（例如优先关系和排序）、时空约束（例如行进时间）和截止时间的前提下，将联盟分配给任务。除了这些问题，我们的问题表述还涉及空间约束（例如任务位置和障碍物）。

用于解决单任务多机器人任务分配问题的基于拍卖的方法 通过竞价过程将任务拍卖给机器人，竞价依据的是一个效用函数，该函数综合考虑机器人（或联盟）执行任务的能力以及任何时间约束（Giordani等人 2013年；Krizmancic等人 2020年；Quinton等人 2023年）。拍卖非常有效，但通常要求每个机器人都了解自己对某项任务的个体效用。因此，这些方法要求满足以下条件之一：i）多机器人任务可分解为子任务，且每个子任务都能由单个机器人完成；或ii）联盟中的机器人明确了解自己对所有协作任务的个体贡献。

端到端优化方法将单任务多机器人任务分配问题表述为一个单一的混合整数线性规划，纳入复杂的相互依赖约束，并以整体完工时间或效用函数作为目标进行优化（Korsah 等，2012年；Choudhury 等人，2022年；Guerrero等人，2017年；Chakraa等人，2023年；Schillinger等人，2018年）。然而，这些方法通常要求所有任务都能够分解为单智能体任务，例如Schillinger等人（2018年）所采用的方法。与基于拍卖和基于优化的方法不同，我们的方法不要求进行任务分解。

学习驱动方法作为多机器人任务分配的一种实用替代范式应运而生（Dai等人 2025年；Wang等人 2022年；Jose和Zhang 2024年；Marino等人 2025年）。基于学习的方法训练一种能够输出分配决策的策略，并且通常在推理时具有更高的可扩展性。Jose和Zhang （2024年）致力于使机器人能够主动等待队友变得可用。该策略通过在一个大规模数据集上进行模仿学习来训练，但样本效率较低。Dai等人（2025年）；Marino等人（2025年）则采用强化学习，以避免对大量示范数据的需求。然而，两者都依赖于为任务定义最低必要资源集合，而在现实场景中，这些资源可能很难任意准确地指定。

我们采用的任务分配方法与基于特征的方法最为接近（Koes，2006年；Prorok 等人，2017年；Ravichandar 等人，2020年；Neville 等，2020年、2021年；Srikanthan 和 Ravichandar，2022年；Fu 等，2022a、b；Singh 等，2023年；Neville 等人，2023年；Park 等人，2023年）。该方法采用灵活的建模框架，以特征的形式对任务需求进行编码（例如，任务 1 要求在运输 50 磅载荷的同时以 10 米/秒的速度行进）。每项任务不受特定智能体集合或数量的限制。相反，其重点是寻找一个由智能体组成的联盟，使这些智能体整体具备所需的能力。因此，这些方法能够灵活且自动地确定联盟规模和组成。然而，现有的大多数基于特征的方法仅限于单任务机器人、多机器人任务、即时分配（ST-MR-IA）问题，此类问题不需要进行调度（Koes，2006年；Prorok 等人，2017年；Ravichandar 等人，2020年；Neville 等人，2020年；Srikanthan 和 Ravichandar，2022年；Singh 等，2023年）；少数值得注意的例外包括 Neville 等人（2021年、2023年）以及 Fu 等（2022b、a），它们能够处理单任务多机器人任务分配问题。

现有基于特征的算法的一个关键局限，包括 Neville 等(2021年，2023年）；Fu 等(2022b）；Parketal. (2023年），在于它们假设用户会明确量化执行每项任务所需的集体特征最低要求。这会带来两个问题。首先，即使是专家，在处理复杂任务和多维能力时，也可能难以明确量化最低要求（Rieskamp et al.2003）。机器人学的其他子领域也意识到了这一问题，并开发了防止和处理奖励或效用

误设问题的方法（Hadfield-Menell 等，2017；Mallozzi 等，2018）。其次，所有方法，包括那些能够学习要求的方法（Fu 等，2022a），都假设分配超过最低要求的任何额外特征都不会给团队带来益处，而且未达到指定阈值就会导致彻底失败。这实际上忽视了任务性能随分配能力变化而自然波动的情况。与此截然不同，E-ITAGS 无需用户指定特征要求，并采用表达能力更强的模型来刻画分配能力与任务性能之间的关系。

2.2 多机器人系统的主动学习

我们综合概述了以多机器人或多智能体系统为背景应用主动学习的既有研究，并根据机器人之间的协调发生在运动层面还是任务层面进行分类。

在运动层面考虑协调时，主动学习是解决多机器人主动感知和信息性路径规划问题的一种常见范式（Igoe 等，2022；Kailas 等，2025；Schlotfeldt 等，2018；Tzes 等，2022；Stadler 等，2023；Cannelli 等，2023；Jang 等，2020）。这些问题需要通过序贯决策，利用机器人的机载传感器收集有关未知环境过程的关键信息。随着收集到的信息不断增加，机器人能够减少其对环境认知中的不确定性。关于何时、何地进行采样的决策，构成了学习过程的“主动性”。

将主动学习应用于多机器人问题、并在任务层面进行协调推理的研究较少受到关注。在为数不多的相关工作中，Le Ny 等人（2006年）将多智能体任务分配建模为一种带有切换成本和奖励的 restless 强盗问题。Singh 等人（2023年）研究了多机器人瞬时任务分配问题，其中奖励函数未知，这与我们的问题设定密切相关。该研究提出了一种基于高斯过程置信上界（Srinivas 等，2009年a）的算法，并采用一种称为 zooming 算法（Slivkins 等，2019）的自适应离散化机制。尽管该方法已被证明有效，Singh 等人（2023年）仅处理所有任务均假定同时发生的情形。

3 问题表述

我们首先介绍先前工作（Neville 等，2021年；Messing 等人，2022；Neville 等人，2023年）中的预备知识以供参考，然后正式定义一类新的时空任务分配问题。

考虑一支由 N 个 i类机器人组成的团队，其中第 i 个机器人的能力由一组特征描述 $q^{(i)} = \left[q_1^{(i)}, q_2^{(i)}, \cdots, q_U^{(i)}\right]^\intercal \in \mathbb{R}^U_{\geq 0}$，其中 $q_u^{(i)}$ 对应于该 i^{th} 机器人的u^{th} 特征。当i^{th} 机器人不具备 u^{th} 特征时，我们将其赋值为 $q_u^{(i)} = 0$ （例如，消防车具有载水量，但其他机器人可能没有）。因此，团队的能力可由一个**团队特征矩阵** $\boldsymbol{Q} = \left[q^{(1)}, \cdots, q^{(N)}\right]^\intercal \in \mathbb{R}^{N\times U}_{\geq 0}$表示，其第 iu 个元素（即第 i 行、第 u 列）表示第 i^{th} 个机器人的 u^{th} 特征。

我们将待完成的 M 个任务建模为一个任务网络 T：一个有向图 $G = (\mathcal{V}, \mathcal{E})$，其中顶点$\mathcal{V}$ 表示一组任务 $\{a_m\}_{m=1}^M$，边 $\mathcal{E}$表示任务之间的约束。对于一对任务 a_i 和 a_j，若它们满足 $a_i \neq a_j$，我们考虑两类约束：(i) 优先约束 $a_i \prec a_j$ 要求任务 a_i 必须在任务 a_j 开始之前完成（例如，必须先扑灭火灾，才能开始维修）（Weld 1994）；(ii) 互斥约束确保 a_i 和 a_j 不能同时进行（例如，机械臂不能同时拿起两个物体）（Bhargava and Williams 2019）。此外，我们将 C_{max} 定义为总时间预算，用于表示用户可接受的最长完工时间。

令$\mathbf{A} \in \boldsymbol{\mathcal{A}} \subseteq \{0,1\}^{M\times N}$ 表示二元**分配矩阵**，其中$\mathbf{A}_{mn} = 1$ 当且仅当第 n 个机器人被分配到第 m 个任务时成立，否则$\mathbf{A}_{mn} = 0$ 。机器人可以单独完成任务，也可以组成联盟协作完成；只要任务安排在互不重叠的时间段内执行，任何机器人都可以被分配到多个任务。我们用$\mathbf{A}_m$ 表示$\mathbf{A}$ 的第 m 行，该行指定了分配到第 m 个任务的机器人联盟。我们进一步定义$\mathbf{Y} = \mathbf{A}\boldsymbol{Q} \in \mathbb{R}^{M\times U}_{\geq 0}$ 为根据分配方案$\mathbf{A}$汇总的所有任务的特征，其中 y_m 表示其第 m 行，包含第 m 个任务可用的集体特征。请注意，为简洁起见，后续章节中我们使用 $\mathcal{Y} = \mathbb{R}^{M\times U}_{\geq 0}$。

3.1 效能优化的时空任务分配

在本研究中，我们扩展了先前工作的问题表述，以考虑任务分配的效能。我们提出了一类新的问题：面向多机器人系统的时空效能优化分配（*STEAM*），旨在优化机器人对任务的分配效能，同时确保相应的完工时间低于用户指定的阈值。随后，我们将 STEAM 扩展为包含对未知特质效能映射的主动学习，此类映射将给定联盟的集体能力映射到任务性能。

令 $\xi_m : \mathbb{R}^U_{\geq 0} \to [0,1]$ 为归一化的**特质-效能映射**，它根据分配给 m^{th} 任务的集体特征，计算与该任务相关的非负效能分数（数值越接近 1，表示联盟越有效）。本质上，ξ_m 量化了给定联盟的能力如何转化为其在 m^{th}任务上的性能。我们定义**总分配效能**来概括所有任务的性能：

$$\Xi(\mathbf{A}) = \sum_{m=1}^{M} \xi_m(\boldsymbol{Q}^T \mathbf{A}_m^T) = \sum_{m=1}^{M} \xi_m(y_m) \qquad (1)$$

请注意，这些映射 ξ_m，$\forall m$ 是从特征（而非机器人）到性能的映射，因此能够得出更具泛化性的关系。我们还将在第3.2 节讨论未知情况下特质-效能映射的主动学习。

我们使用元组 **问题域** $\boldsymbol{\mathcal{D}} = \langle$任务网络 $\boldsymbol{Q}$, Ξ, I_c, L_T, W, $C_{max}\rangle$来定义，其中任务网络 是任务网络，$\boldsymbol{Q}$ 是团队特征矩阵，Ξ是汇总性能函数，I_c 和 L_T 分别是与任务相关的所有初始配置集和终止配置集，

W 是对世界的描述（例如地图），C_{max} 是总时间预算。

最后，我们使用元组 $\boldsymbol{S} = \left\langle \hat{\mathbf{A}}, \hat{\sigma}, \hat{X} \right\rangle$，定义**解**，以解决由$\mathcal{D}$定义的问题。其中，$\hat{\mathbf{A}}$ 表示一个分配，$\hat{\sigma}$表示一个满足所有时间约束的调度，而 X 是一个有限的无碰撞运动规划方案集合。STEAM 的目标是在遵守总时间预算 C_{max} 的同时，找到使总分配效能 $\Xi(\mathbf{A})$最大化的解。

3.2 可实现性感知主动学习

如上所述，$\xi_m(\cdot)$是一个函数，它将分配给第 m个任务的集体特征映射为任务性能的度量。准确了解这些映射，有助于在将有限资源（即机器人）分配给竞争目标（即任务）时做出有效权衡，但这些映射通常难以明确指定。因此，我们不假设已知 $\xi_m(\cdot) \ \ \forall m$，而是假设可以访问一个模拟器，并从中学习。我们特别希望限制查询次数，以便开展学习，因为此类查询可能成本高昂或耗时。因此，我们将学习 $\xi_m(\cdot) \ \ \forall m$ 的问题表述为一个主动学习问题：必须有效采样有前景的联盟，从而高效学习特征—效能映射。

为了主动学习 $\{\xi_m(\cdot)\}_{m=1}^{M}$，我们使用查询选择器，根据采集函数 $V: \mathcal{Y} \to \mathbb{R}$ 的指示，采样最有希望的数据点。采集函数用于量化样本的效用，常见定义包括最大化置信上界（Srinivas 等，2009a）或最大化方差。有效采样面临的一大障碍是，尽管 V 认为点 $\mathrm{Y} \in \mathcal{Y}$ 具有较高效用，但对于给定的一组具有团队特征矩阵 $\boldsymbol{Q}$ 的机器人而言，该点未必可实现。这是因为每个机器人都是不可分割单元，只能被分配或不被分配给某项任务。事实上，真正能够直接实现的样本是满足 $\mathrm{Y} = \mathrm{A}\boldsymbol{Q}$ 的样本，其中 A 属于分配空间 $\mathrm{A} \subseteq \{0,1\}^{M \times N}$。形式上，我们将 $\mathcal{Y}_{\boldsymbol{Q}} = \{\mathrm{Y} \in \mathcal{Y} | \mathrm{Y} = \mathrm{A}\boldsymbol{Q}, \forall \mathrm{A} \in \mathrm{A}\}$ 定义为所有直接可实现样本的集合。我们考虑到采集函数的常见设计实践与样本可实现性之间存在这种脱节。为弥合这一差距，我们旨在利用样本的可实现性来指导采集函数。

4 时间预算下的效能优化

为了解决第 3 节所定义的 STEAM 问题，我们提出了**效能优化迭代任务分配图搜索（E-ITAGS）**算法。本节重点讨论在线解决 STEAM 问题，并假设特质效能映射 $\{\xi_m(\cdot)\}_{m=1}^{M}$ 已经学习完成。我们将在第 5 节讨论特质效能映射的学习。

我们首先简要介绍 E-ITAGS，之后再详细说明。E-ITAGS 采用交错式架构，同时解决任务分配、调度和运动规划问题（见图 1）。这种交错式方法继承自我们之前的工作 Neville 等 (2021年)； Messing 等(2022年)； Neville 等 (2023年)，在这些工作中，我们充分证明了与文献中常见的顺序式方法相比，该方法具有显著优势。例如，交错式架构的计算效率明显更高，因为它们避免了回溯（Messing 等 2022年）。然而，E-ITAGS 相较于我们之前的工作进行了三项关键改进：i）优化分配效能而非完工时间；ii）采用更具表现力的任务性能模型；iii）遵守时间预算（见第 2 节）。

E-ITAGS 执行增量式基于图的搜索，并由一种新颖的启发式方法引导。该启发式方法在优化任务分配效能与满足时间预算要求之间平衡搜索。我们构建并求解混合整数线性规划（MILP），以解决增量搜索中的调度与运动规划问题。为克服显式指定任务需求所带来的挑战，E-ITAGS 使用主动学习模块来学习联盟的集体特征与任务性能之间的关系。当施加非常严格的时间预算 C_{max} 时，无论如何分配机器人，都可能不存在可行解。对于这种情况，E-ITAGS 可以在搜索完所有组合且未找到解后，报告不可行性。

4.1 任务分配

任务分配层通过增量任务分配图 $\mathcal{N}$ 进行贪心最佳优先搜索。在该图中，每个节点表示将机器人分配给任务的一种分配方案。节点与其他节点相连，且这些节点之间仅有一个机器人的分配不同。请注意，根节点表示将所有机器人分配给每个任务。实际上，这种分配通常会带来最佳性能（因为所有可用机器人都参与每项任务），但也会导致最长的调度（按顺序执行任务）。E-ITAGS 的搜索从这个初始节点开始，并逐步移除分配，以寻找一种能够满足时间预算且不会显著牺牲任务性能的分配方案。

为引导搜索，我们设计了两种启发式方法：归一化分配成本 ***(NAC)***，根据分配的效能引导搜索；以及 时间预算超限 ***(TBO)***，根据与该分配相关的调度完工时间引导搜索。

归一化分配成本 ***(NAC)***是衡量给定分配如何改善任务性能的归一化指标：

$$f_{NAC}(\overline{\mathbf{A}}) = \frac{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\overline{\mathbf{A}})}{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_{\text{null}})} \tag{2}$$

其中，$\overline{\mathrm{A}}$ 表示正在评估的分配，$\Xi(\mathrm{A})$ 表示式 (1) 中定义的总分配效能，A_{root} 表示根节点处的分配，而 A_{null} 表示没有机器人被分配给任何任务。由于 $\Xi(\mathrm{A}_{\text{root}})$ 和 $\Xi(\mathrm{A}_{\text{null}})$ 分别定义了分配效能的上界和下界，因此 NAC 被限制在 $[0,1]$ 范围内。请注意，计算 $\Xi(\cdot)$所需的效能函数 $\{\xi_m(\cdot)\}_{m=1}^{M}$将按照第5节所述进行学习。我们将这一启发式方法称为“成本”，因为式（2）本质上计算的是任务性能的归一化下降，即当我们从根节点中减去分配时所需付出的成本。

NAC 不考虑调度层，并会在分配图中进行更广泛的搜索。这是因为

较浅的节点已分配更多机器人，因此往往能带来更高的分配效能。因此，NAC 倾向于选择能最大化分配效能和任务性能的分配，即使这可能导致完工时间更长。

时间预算超限 ***(TBO)***衡量给定分配对应的调度超出时间预算的程度，其计算方式为

$$f_{TBO}(C_{\overline{\boldsymbol{\sigma}}}) = \max\left(\frac{C_{\overline{\boldsymbol{\sigma}}} - C_{max}}{|C_{\boldsymbol{\sigma}_{worst}} - C_{max}|},\ 0\right) \tag{3}$$

其中，$C_{\boldsymbol{\sigma}}$ 表示调度 $\boldsymbol{\sigma}$的完工时间，$\boldsymbol{\sigma}$ 表示与正在评估的 $\mathbf{A}$ 对应的调度，C_{max} 表示用户指定的完工时间时间预算，$\boldsymbol{\sigma}_{worst}$ 表示将所有机器人分配给每项任务的最长调度。

由于 TBO 只考虑调度而不考虑分配，因此它倾向于偏好图中更深层的节点；这些节点拥有更少的机器人和约束，因而具有更短的完工时间。因此，TBO 偏好满足时间预算的分配，但代价是更深的搜索和更多的节点扩展。时间扩展任务分配指标 ***(TETAM)*** 是 NAC 和 TBO 的凸组合，用于平衡分配效能与时间预算：

$$f_{TETAM}(\overline{\mathbf{A}}, C_{\overline{\boldsymbol{\sigma}}}) = (1-\alpha) f_{NAC}(\overline{\mathbf{A}}) + \alpha f_{TBO}(C_{\overline{\boldsymbol{\sigma}}}) \tag{4}$$

其中，$\alpha \in [0,1]$ 是用户指定的参数，用于控制每种启发式方法的相对影响。因此，TETAM 在搜索解时同时考虑分配效能和调度。有关分配效能与完工时间之间权衡的理论分析，请参见第 4.3节。

4.2 调度与运动规划

E-ITAGS 的调度层会检查给定分配的调度可行性，并帮助计算其 TBO。为此，我们构建了一个混合整数线性规划（MILP），其中考虑了三类时间约束：优先约束、互斥约束和行驶时间约束。优先约束$\mathcal{P}$ 确保一个任务先于另一个任务执行（例如，必须先扑灭火灾，才能开始维修）。互斥约束 $\mathcal{M}$确保两个任务不会同时执行（例如，机器人不能同时拿起两个物体）。行驶时间约束确保机器人有充足的时间在任务地点之间行进（例如，先前往火灾地点，再进行灭火）。该混合整数线性规划形式如下：

$$\begin{aligned}
\min_{\{s_i\},\{p_{ij}\}} \quad & C \\
\text{s.t. } & C \geq s_i + d_i,\ \forall i = 1,..,M \\
& s_j \geq s_i + d_i + x_{ij},\ \forall (i,j) \in \mathcal{P} \\
& s_i \geq x_i,\ \forall i = 1,..,M \\
& s_i \geq s_j + d_j + x_{ji} - \beta p_{ij},\ \forall (i,j) \in \mathcal{M}^R \\
& s_j \geq s_i + d_i + x_{ij} - \beta(1 - p_{ij}),\ \forall (i,j) \in \mathcal{M}^R
\end{aligned}$$

其中 C 是完工时间，s_i 和 d_i 分别是任务 a_i 的（相对）开始时间和持续时间，x_{ij} 是从 a_i 所在地点前往 a_j 所在地点所需的时间，x_i 是分配的联盟前往任务 a_i 所在地点所需的初始时间；当且仅当 a_i 优先于 a_j时，$p_{ij} \in \{0,1\}$和 $p_{ij} = 1$ 为真；$\beta \in \mathbb{R}_+$ 是一个较大的标量，$\mathcal{P}$ 和 $\mathcal{M}$是整数对集合，分别包含优先约束和互斥约束列表，其中 $\mathcal{M}^R = \mathcal{M} - \mathcal{P} \cap \mathcal{M}$ 表示移除了优先约束的互斥约束。

E-ITAGS 通过欧几里得距离估算行进时间，以此初始化调度。在搜索过程中，调度层会反复查询运动规划器，以更准确地估算行进时间。E-ITAGS 会持续迭代，直到调度所需的所有运动规划方案均已实例化并记忆化。

4.3 分配次优界

为了更好地理解 E-ITAGS 的性能，我们分析了 α——用户指定的参数，它决定了两种启发式方法的相对重要性——对所得解的最优性的影响；该最优性是根据任务分配的效能定义的。我们证明，α 的取值决定了一个次优界；在温和的假设下，α 在 $\alpha = 0$ 和 $\alpha = 0.5$ 之间取值时，次优界会逐渐收紧。此外，这项分析以及 α 越接近 1，所得解越倾向于优先考虑完工时间这一事实，有助于直观理解分配效能与完工时间之间的关系。

下面，我们针对总分配效能，推导由 E-ITAGS 生成的解的严格次优界。

定理 1. 对于任意给定的问题域 $\boldsymbol{\mathcal{D}}$，令 $\boldsymbol{A}^*$ 为ˆ关于总分配效能的最优分配，而 $\boldsymbol{A}$ 为针对同一问题由 *E-ITAGS* 生成的解所对应的分配。假设增加机器人数量绝不会降低联盟的性能，则当 $\alpha < 0.5$ 在公式（*4*）中时，*E-ITAGS* 解相对于分配效能的次优性如下：

$$\Xi(\boldsymbol{A}^*) - \Xi(\hat{\boldsymbol{A}}) \leq \frac{\alpha}{1-\alpha}(\Xi(\boldsymbol{A}_{root}) - \Xi(\boldsymbol{A}_{null})), \tag{5}$$

其中 Ξ（$\boldsymbol{A}_{root}$）和 Ξ（$\boldsymbol{A}_{null}$）分别表示所有机器人均分配给所有任务时，以及没有机器人分配给任何任务时的总分配效能。此外，当 $\alpha = 0$ 时，*E-ITAGS* 给出最优解，且次优差距消失，即 Ξ（$\boldsymbol{A}^*$）- Ξˆ$\boldsymbol{A}$） $= 0$。

证明。由于对父节点的任何扩展都表示减去一个分配，因此给定节点 N 必然比分配给其父节点 N_p的机器人更少。结合我们的假设——当移除任意一个已分配的机器人时，联盟性能绝不会提高（即没有机器人产生负贡献）——可以看出，

$$f_{NAC}(N) \geq f_{NAC}(N_p), \tag{6}$$

因为分配效能越高，NAC 启发式值就越低，反之亦然。因此，我们可以推断，E-ITAGS 图中未展开集合 $\mathcal{U} \subseteq \mathcal{N}$ 的所有节点的 NAC 值都大于或等于它们各自在已展开集合 $\mathcal{O} \subseteq \mathcal{N}$ 中的前驱节点的 NAC 值。因此，未展开集合中的最小 NAC 值必定大于已展开集合中的最小 NAC 值：

$$\min_{N \in \mathcal{U}} f_{NAC}(N) \geq \min_{N \in \mathcal{O}} f_{NAC}(N) \tag{7}$$

无论最优分配 $\mathbf{A}^*$对应于开放集合还是未展开集合中的一个节点，式（7）中的不等式都意味着

$$\Xi(\mathbf{A}^*) \leq \max_{N\in\mathcal{O}} \Xi(\mathbf{A}_N) \tag{8}$$

其中，$\mathbf{A}_N$ 表示给定节点 N 的分配。我们用 N' 表示开放集中总分配效能最高的节点。也就是说，$N' = \arg\max_{N\in\mathcal{O}} \Xi(\mathbf{A}_N)$。

由于我们要求 E-ITAGS 的任何有效解都必须遵守^时间预算，因此解节点（N）的 TBO 值为零。因此，式（4）中定义的解节点的总 TETAM 启发式为

$$f_{TETAM}(\hat{N}) = (1-\alpha) f_{NAC}(\hat{N}) \tag{9}$$

由于 E-ITAGS 执行贪心最佳优先搜索，因此在找到^解 N 时，解节点的 TETAM 值保证小于或等于开放集中的所有节点：

$$f_{TETAM}(\hat{N}) \leq f_{TETAM}(N), \forall N \in \mathcal{O} \tag{10}$$

展开 TETAM 的定义并使用式（9），可得

$$\begin{aligned}(1-\alpha)\frac{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\hat{\mathbf{A}})}{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_{\text{null}})} \leq \\ \alpha f_{TBO}(N) + (1-\alpha)\frac{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_N)}{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_{\text{null}})}, \forall N \in \mathcal{O}\end{aligned} \tag{11}$$

利用式（8）中的界，以及 $f_{TBO}(\cdot) \leq 1$这一事实，我们将上述方程右侧改写为

$$\begin{aligned}(1-\alpha)\frac{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\hat{\mathbf{A}})}{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_{\text{null}})} \leq \\ \alpha + (1-\alpha)\frac{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}^*)}{\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_{\text{null}})},\end{aligned} \tag{12}$$

重新排列并消去两边的等价项后，我们得到

$$\begin{aligned}(1-\alpha)(\Xi(\mathbf{A}_{\text{root}}) - \Xi(\hat{\mathbf{A}})) \leq \\ \alpha(\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_{\text{null}})) + (1-\alpha)(\Xi(\mathbf{A}_{\text{root}})) - \Xi(\mathbf{A}^*))\end{aligned} \tag{13}$$

重新排列各项可得式（5）中的界。

此外，当 $\alpha = 0$ 时，式（5）可简化为 $\Xi(\mathbf{A}^*) - \Xi(\hat{\mathbf{A}}) \leq 0$。然而，由于最优解必须实现最高的总分配效能，$\Xi(\mathbf{A}^*) - \Xi(\hat{\mathbf{A}}) = 0$。因此，当 $\alpha = 0$ 时，E-ITAGS 返回具有最优分配效能的解。 □

请注意，只有当 $0 \leq \alpha < 0.5$ 时，上述定理才能对最优解与 E-ITAGS 的解之间的分配效能差异给出合理的界。当 $\alpha \geq 0.5$ 时，随着界超过 $\Xi(\mathbf{A}_{\text{root}})$ 与 $\Xi(\mathbf{A}_{\text{null}})$ 之间效能差异的最大值，它便失去意义。换言之，当 $\alpha \geq 0.5$时，式（5）显然成立，因为 $\Xi(\mathbf{A}_{\text{null}})$ 的效能指标在所有分配中最差，而 $\Xi(\mathbf{A}_{\text{root}})$ 的效能指标最好，这意味着大于 $\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_{\text{null}})$ 的界并无意义。

事后界：式（5）中的界可以在执行后进一步收紧，以便进行事后分析。

具体而言，我们可以不再将 $f_{TBO}(N)\ \forall N \in \mathcal{O}$界定为 1，而是针对 N' 精确计算 $f_{TBO}(N')$，其中该节点在 $\mathcal{O}$ 中具有最佳分配效能。然后，按照与上述类似的代数运算，可以得到一个更紧的次优差距界：

$$\Xi(\mathbf{A}^*) - \Xi(\hat{\mathbf{A}}) \leq \frac{\alpha}{1-\alpha}(\Xi(\mathbf{A}_{\text{root}}) - \Xi(\mathbf{A}_{\text{null}}))f_{TBO}(N')$$

5 可实现性感知主动学习

我们现在介绍一种高效学习特质效能映射的方法 $\{\xi_m(\cdot)\}_{m=1}^M$。如第3.2节所述，对于每个任务 m，其特质效能映射 $\xi_m(y_m)$ 描述了分配给它的集体特征 y_m 与所分配联盟的性能之间的关系。基于主动学习的思想，我们在每次迭代中都旨在采样y_m，$\forall m$。我们将相应的分配矩阵$\mathbf{A}$输入模拟器进行评估和标注。随后，根据集体特征与标签的配对迭代更新这些映射。

我们将分配给任务的集体特征与所得性能的联合分布建模为高斯过程（GP），记为 $P(y_m, \xi_m(y_m)) \sim \mathcal{GP}(\mu_m, \Sigma_m)\ \forall m$，因为已有研究表明，高斯过程能够有效地对广泛的函数进行建模（Singh 等，2023年；Srinivas 等，2009年a）。这一选择使我们能够使用高斯过程回归学习特质效能映射。相应地，我们使用置信上界（UCB）（Srinivas 等，2009年a）来评估采集函数中各点的效用。

如第3.2节所述，只有满足$\mathbf{Y} \in \mathcal{Y}_{\boldsymbol{Q}}$ 的样本才可直接实现。由于这些样本分布稀疏，因此仅占连续特质空间的很小一部分 $\mathcal{Y}$。因此，采用拒绝采样这一直观策略并不可靠，甚至可能不可行，因为它需要检查数量极其庞大的点，才能获取足够数量的可实现样本。

可实现性投影：克服拒绝采样效率低下的问题，并弥合采集函数与样本可实现性之间的差距，一种方法是使用投影。为此，我们首先选取效用 V最高的$\mathbf{Y} \in \mathcal{Y}$，并通过首先求解以下优化问题，将其投影到可实现集 $\mathcal{Y}_{\boldsymbol{Q}}$ 上，

$$\mathbf{A}' = \arg\min_{\mathbf{A}\in\boldsymbol{\mathcal{A}}} \|\mathbf{Y} - \mathbf{A}\boldsymbol{Q}\|_2^2, \tag{14}$$

这是一个以二次目标函数为目标的整数二次规划。通过求解式（14），我们得到与期望聚合特质 Y 最匹配的分配矩阵。随后，计算可实现任务特质矩阵 $\mathbf{Y}' = \mathbf{A}'\boldsymbol{Q}$。在本文余下部分，我们将求解式（14）并计算 $\mathbf{Y}'$ 称为可实现性投影。$\mathbf{A}'$ 和 $\boldsymbol{Q}$ 被输入模拟器，以获取表示联盟性能的标签。稍微滥用一下符号，我们将这些标签记为 $\{\xi_m\}_{m=1}^M$。我们使用配对数据 $(\mathbf{y}'_m, \xi_m)$ 训练与任务 m 对应的高斯过程，其中 $\mathbf{y}'_m$ 是 $\mathbf{Y}'_m$中对应于任务 m 的行。由于对 Y 进行采样时不施加任何约束，而可实现性完全通过投影赋予，因此我们将这种策略称为无约束策略。

无约束策略的一个突出失效模式是，高斯过程模型可能会反复依赖数据

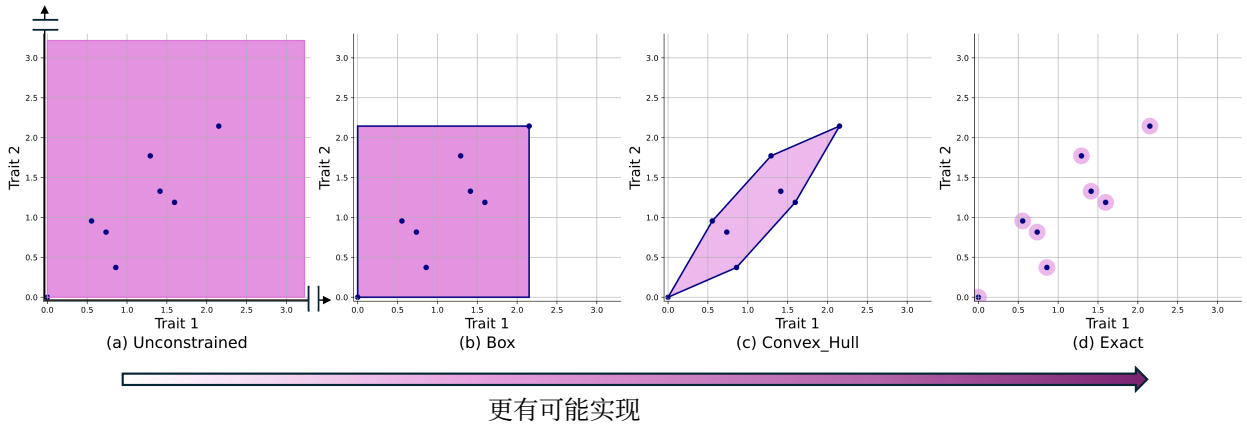


图 2. 我们展示了四种策略中可供采样的子空间，并按样本可实现性由低到高排列。场景中有 3 个机器人，每个机器人由 2 个特征刻画。对于每种策略，粉色区域表示采集函数选择额外特征组合进行标注时允许采样的子空间。请注意，在 (a) 中，无约束子空间以一个沿坐标轴延伸至无穷远的大粉色矩形表示。为清楚展示精确策略，我们仅在每个特征组合周围绘制粉色圆形区域。实际上，只有深蓝色的离散点才是有效的选择候选项。

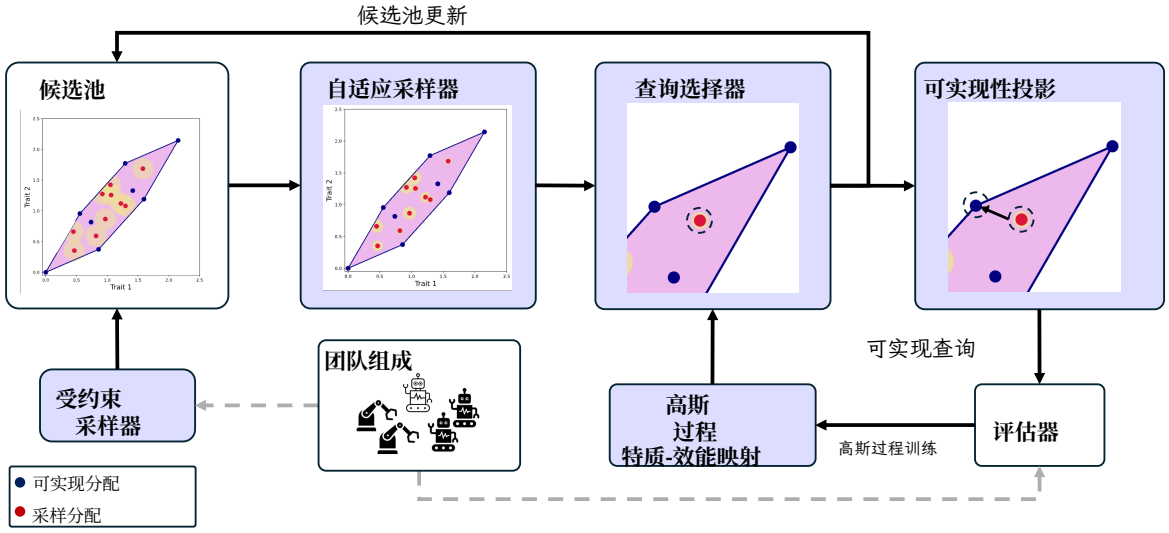


图 3. 所提出的可实现性感知主动学习流程。我们通过一个简单场景展示这一思路：场景中有 3 个机器人，每个机器人由 2 个特征刻画，并采用凸包约束策略。盒约束策略的流程相同，区别仅在于其允许采样候选项的区域不同。在表示特征空间的二维平面上，粉色区域由 $\mathcal{Y}_Q$ 中所有可直接实现的特征组合（深蓝色点）构成的凸包圈定。凸包内的红点是随机采样的候选项，每个候选项的邻域以黄色圆圈表示。请注意，所有邻域的初始大小相同，而缩放机制会自适应地缩小邻域，如“自适应采样器”中的较小黄色圆圈所示。基于 UCB 的选择器选出候选项（以虚线圆圈表示），随后投影器将其投影到 $\mathcal{Y}_Q$ 中的最近邻。可实现的样本被送入模拟评估器进行标注，并用于训练特质-效能映射。

点$\mathbf{Y}'$，而这些数据点未必具有潜力（低效用值 $V(\mathbf{Y}')$ 即可说明这一点），因为 $\mathbf{Y}'$ 可能与原始$\mathbf{Y}$相距任意远。这会减缓模型利用其关于聚合特征空间中哪些部分能带来更高奖励的知识。

约束采样：与无约束策略相比，我们可以将采样候选聚合特征的范围限制为 $\mathcal{Y}_Q$ 中可实现的特征。给定团队组成矩阵 Q，在每次学习迭代中，我们的目标是通过求解来选择 $\mathbf{Y}$

$$\max_{\mathbf{Y}\in\mathcal{Y}_Q} V(\mathbf{Y}), \tag{15}$$

这自然保证了$\mathbf{Y}$ 可由机器人团队实现，因此无需执行投影步骤。由于该策略只考虑$\mathbf{Y}$的精确可实现值集合，我们称之为精确策略。注意，等式（15）可以立即改写为：

$$\begin{aligned} \max_{\mathbf{Y},\,\mathbf{A}\in\mathcal{A}} \quad & V(\mathbf{Y}) \\ s.t. \quad & \mathbf{Y} = \mathbf{A}Q \end{aligned} \tag{16}$$

方程（16）是一个混合整数优化程序（MIP），其目标函数 $V(\cdot)$ 用于估计变量 $\mathbf{Y}$ 的效用。常见的 $V(\cdot)$ 包括最大熵和置信上界，它们依赖于特质效能映射分布的一阶和

二阶矩。由于我们采用高斯过程来建模特质效能映射，因此，在优化器 $V(\cdot)$ 中，每次迭代都需要从高斯过程中提取均值和方差，而大多数现成的混合整数非线性规划求解器（MINLP）并不直接支持这一操作。因此，求解方程（16）需要付出大量努力，将高斯过程嵌入 MINLP 软件，或设计代理模型来近似 $V(\cdot)$ 本身。

平衡可实现性与计算效率：为了在实现可实现性感知采样的同时避免造成沉重的计算或后勤负担，我们提出了一系列不同程度近似采样区域的方法。这些方法的灵感来自Baptista 和 Poloczek (2018年，我们通过从以下放宽后的连续子空间中采样来近似可实现集 $\mathcal{Y}_{\boldsymbol{Q}}$ $\mathcal{Y}$。

盒约束：给定团队特征矩阵 $\boldsymbol{Q}$，我们可以通过对其各行求和，计算整个团队提供的特征总量：$\bar{\mathbf{q}} = \mathbf{1}_N^\top \boldsymbol{Q} \in \mathbb{R}^U_{\geq 0}$，其中 $\mathbf{1}_N$ 是维度为 N 的全一向量。我们用向量 $\bar{\mathbf{q}} \in \mathbb{R}^U_{\geq 0}$表示求和结果。因此，采样$\mathbf{Y}$时，$\bar{\mathbf{q}}$ 可作为精确的上界。我们还将零向量$\mathbf{q}_0 = \mathbf{0} \in \mathbb{R}^U_{\geq 0}$定义为平凡下界。我们将采样区域限制在由 $\bar{\mathbf{q}}$ 和 $\mathbf{q}_0$，围成的盒状子空间内，记为 $\mathcal{B}$。我们将这种仅从 $\mathcal{B}$中采样的策略称为盒约束策略。在这种情况下，选择$\mathbf{Y}$ 被表述为求解

$$\max_{\mathbf{Y} \in \mathcal{B}} V(\mathbf{Y})$$

凸包：通过从 $\mathcal{Y}_{\boldsymbol{Q}}$ 内各点的凸包中进行采样，我们可以进一步得到对 $\mathcal{Y}_{\boldsymbol{Q}}$ 更紧致（但仍连续）的近似。我们将这一想法称为凸包约束策略，简称凸包策略。我们将$\mathbf{S} \in \mathcal{S} = [0,1]^{M \times N}$ 定义为分配矩阵$\mathbf{A}$的连续松弛对应形式。通过随机采样$\mathbf{S}$ 并计算$\mathbf{Y} = \mathbf{S}\boldsymbol{Q}$，我们可以获得上述凸包内的一个随机样本。选择要标注的样本，本质上就是在求解，

$$\begin{aligned} \max_{\mathbf{Y},\,\mathbf{S}} \quad & V(\mathbf{Y}) \\ s.t. \quad & \mathbf{Y} = \mathbf{S}\mathbf{Q}, \\ & \mathbf{S} \in \mathcal{S}. \end{aligned} \tag{17}$$

我们在附录中简要证明，样本$\mathbf{Y}$ 位于凸包内。A。为帮助理解串联四种策略的思路，我们在图2中展示了它们感兴趣的采样区域，即单个任务的特征空间。该图展示的是涉及 3 个机器人的分配问题，其中每个机器人由两个特征刻画（即对应于 x 和 y 二维平面上的坐标轴）。我们按照可实现性从低到高的顺序排列这四种策略：从图2(a) 中可实现性最低的（无约束）策略，到图2(d) 中完全可实现的（精确）策略。在这两个极端之间，图2(b) 和图2(c) 展示了遵循可实现性约束可以体现为控制采样所来自的子空间。

图3展示了我们具备可实现性意识的主动学习流水线概览。给定由 $\boldsymbol{Q}$编码的团队组成，受约束采样器

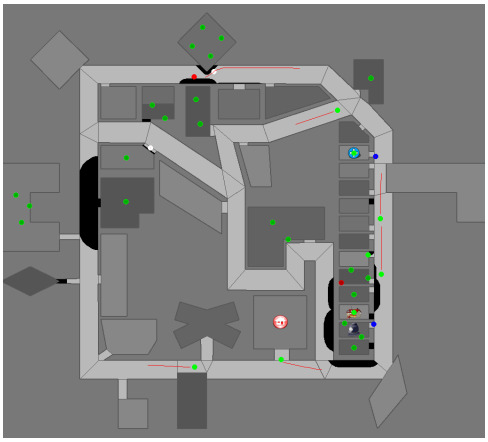


图4. 机器人世界杯搜索与救援场景的布局。黑色障碍物表示通道受到（部分）阻塞。绿色圆点代表等待救援的幸存平民。红色圆点代表消防队，蓝色圆点代表警察部队，白色圆点代表救护车。

会在凸包或盒状区域内随机采样一组特征组合，以初始化候选池。随后，主动学习过程会循环执行以下步骤，直至用于查询模拟器的预算耗尽：(i) 自适应邻域采样器在每个候选项周围的半径范围内采样少量点。我们称这些点为邻居点。(ii) 高斯过程模型通过汇总各邻居点的UCB分数来评估每个候选项；(iii) 基于UCB的选择器选出汇总分数最高的候选项；(iv) 将选出的候选项投影到 $\mathcal{Y}_{\boldsymbol{Q}}$中离它最近的邻居点，该点称为可实现查询。(v) 模拟器接受与可实现查询对应的分配矩阵$\mathbf{A}$ ，模拟该任务，并返回一个分数。(vi) 高斯过程模型使用这对新的可实现查询和分数进行更新。

请注意，在无约束、盒约束和凸包策略中，我们采用缩放机制，该机制源自Srinivas 等人(2009年b)；Singh 等人 (2023年)，以自适应地“缩放”到效用分数较高的有前景 $\mathbf{Y}$ 样本上。采样策略从一组随机选取的候选集体特征开始。每个候选项都关联一个局部邻域，我们通过汇总该邻域内一组邻居点的效用来评估每个候选项的效用。随着主动学习过程的推进，每个候选项的邻域大小会动态调整；被选中的频率越高，邻域就会缩小。这种自适应机制使我们能够聚焦于有前景的候选项，并以更精细的粒度仔细考察它们。请注意，Singh 等人 (2023年)通过将特征空间离散为均匀网格来实现缩放算法。对于我们的凸包约束策略，要在凸多面体内生成指定数量的网格点组成的均匀格点，本身就是一个颇具挑战的问题。因此，我们以随机均匀采样取代基于网格的离散化，同时保留自适应离散化的核心思想。

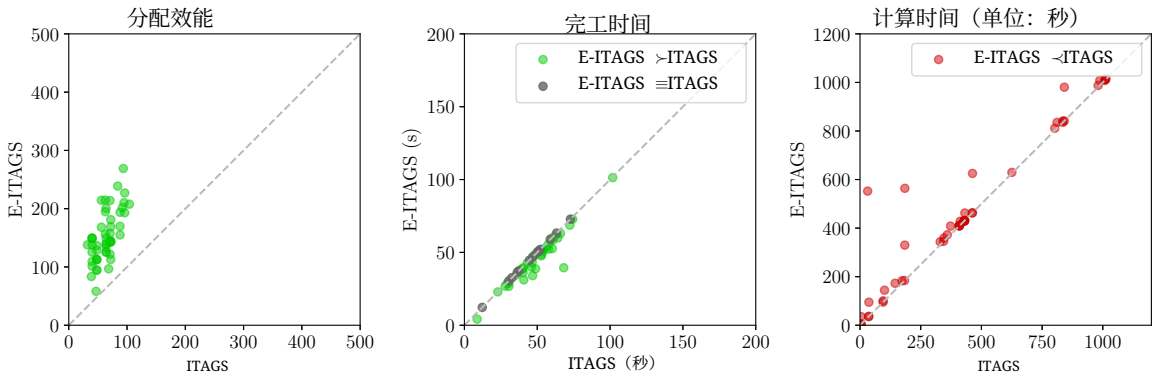


图 5. 比较 E-ITAGS 与 ITAGS 的分配效能、完工时间和计算时间。E-ITAGS 始终能生成效能更优的解（左），同时确保其完工时间优于或等于 ITAGS（中）。绿色圆点表示 E-ITAGS 的表现优于 ITAGS，灰色圆点表示 E-ITAGS 与 ITAGS 的表现相同。较大的分配效能和较短的完工时间更为理想。红色圆点表示 E-ITAGS 相较于 ITAGS 的优势是以略差的计算时间为代价的（右）。

6 实验与结果

我们在模拟应急响应领域中通过三组实验评估了 E-ITAGS（Kitano 等 1999；Bechon 等人 2014；Messing 和 Hutchinson 2020；Whitbrook 等人 2015；Zhao 等 2016）。首先，我们将 E-ITAGS 与一种最先进的基于特征的任务分配方法进行比较，考察其分配性能。其次，我们检验资源分配次优性的理论界在实践中是否成立。随后，我们评估了 E-ITAGS 中的可实现性感知主动学习模块。评估首先通过数值模拟开展，随后将学习到的模型纳入 E-ITAGS 的启发式搜索中进行整体评估。

6.1 实验领域

我们首先简要介绍评估中始终采用的实验领域。在第 6.2 节和第 6.3 节的实验中，我们采用 RoboCup 搜索与救援场景（Kitanoand Tadokoro2001；Kitano 等 1999；Federation 2025）作为贯穿全文的实验领域。在这一模拟领域中，异构机器人团队（包括警察部队、救护车等）负责在火灾等灾害发生后搜寻并营救平民。图 4 展示了该领域中一次搜索与救援任务期间的截图。此外，在第 6.5 节中，我们还借助 RoboCup 救援模拟器，为主动学习流水线提供真实标签。

6.2 评估时间预算下的效能优化

我们在 50 个问题实例上评估 E-ITAGS 的性能，并将其与 ITAGS（Neville 等，2021，2023）进行比较。我们选择将本方法与 ITAGS 进行比较，原因有二：i）已有研究表明，其性能优于其他最先进的扩展时间的任务分配算法；ii）ITAGS 同样采用基于特征的方法，具有相似的假设，因此能够进行更合理的比较。A
关键区别在于，ITAGS 在满足阈值化分配要求的前提下最小化完工时间，而 E-ITAGS 则在满足最大完工时间要求的前提下优化分配性能。考虑到 ITAGS 并不涉及特质效能映射这一概念，我们假设可以获取这些函数的真实值，并将其建模为联盟特质的加权线性组合。对于每个问题实例，我们首先使用 ITAGS 计算一个解。然后，使用真实特质效能映射计算 ITAGS 解的分配效能。接着，我们评估 E-ITAGS 在将其时间预算限制为与 ITAGS 解的完工时间相等时，提升 ITAGS 分配效能的能力。

如图所示 5 (左侧 和中间)，E-ITAGS 在所有问题中都能持续提升分配效果，同时确保其完工时间等于或短于 ITAGS。原因在于，E-ITAGS 能够确保完工时间不超过 ITAGS，同时继续搜索能够最大化任务性能的更优解。然而，如图所示，这种性能提升是以计算时间略有增加为代价的。5 (右侧)。

为了了解计算时间增加的来源，我们进行了更细粒度的分析，比较 E-ITAGS 和 ITAGS 分别在任务

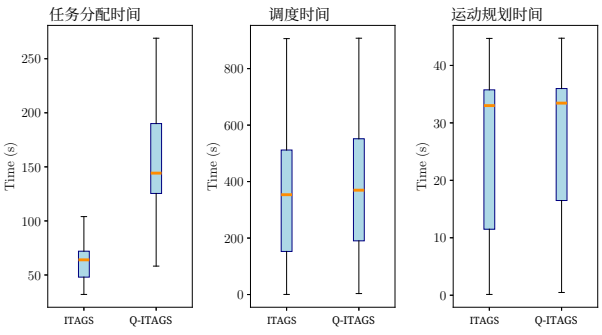


图 6。 我们比较了 E-ITAGS 和 ITAGS 在三个关键模块中的运行时间：任务分配、调度和运动规划。总运行时间的增加可归因于 E-ITAGS 花费更多时间搜索任务分配图。

分配、调度和运动规划上所花费的时间。结果以箱线图的形式汇总于图 6 中。我们观察到，尽管两种方法在调度和运动规划上花费的时间相近，但总体而言，E-ITAGS 在任务分配搜索上需要更多时间，这解释了图 5 中计算开销的增加。由于节点评估占任务分配搜索中大部分的计算量，我们对 E-ITAGS 和 ITAGS 为找到满足条件的分配方案而评估的节点数量进行了 Welch t 检验。检验得到的 p 值为 0.0047，因此拒绝零假设，并证实与 ITAGS 相比，E-ITAGS 在找到完工时间满足时间预算的分配方案之前，会在统计意义上探索更多节点。

6.3 最优性差距界限的验证

在第二项实验中，我们通过实证检验了第4.3节中关于分配效能的理论保证是否成立。为此，我们以 α 为步长，从 $[0,1]$递增至0.1，并针对每个 α值，求解上一节中的 50 个问题实例。对于每种问题与 α 值的组合，我们计算了实际归一化最优性差距，以及式（5）中相应的归一化理论界。我们发现，在 α 的所有取值下，最优性差距始终符合理论界（见图7）。正如预期，极端取值 $\alpha = 0$ （忽略 TBO）和 $\alpha = 1$ （忽略 NAC）分别带来最佳和最差的分配效能。

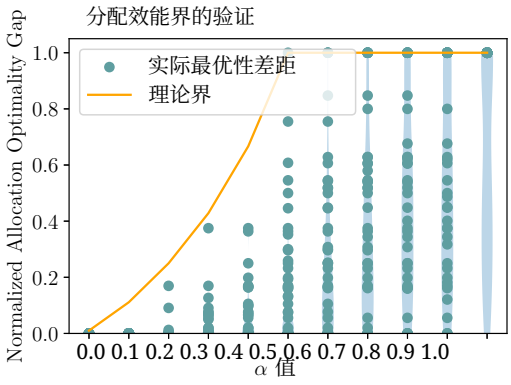


图 7。对于不同的 α 取值，理论界始终成立。归一化最优性差距为0 表示最优分配，为1 则表示实验中观察到的最差分配。

6.4 评估可实现性感知主动学习

在第 3.2 节中，我们提出了四种不同的策略来选择要查询的样本：无约束、精确、盒约束和凸包约束。现在，我们评估这四种策略，并比较它们学习未知特质-效能映射以及发现高奖励分配方案的能力。实验结果包括：(i) 在简单的一维特征空间上进行数值模拟，旨在定性展示不同策略的学习行为；(ii)

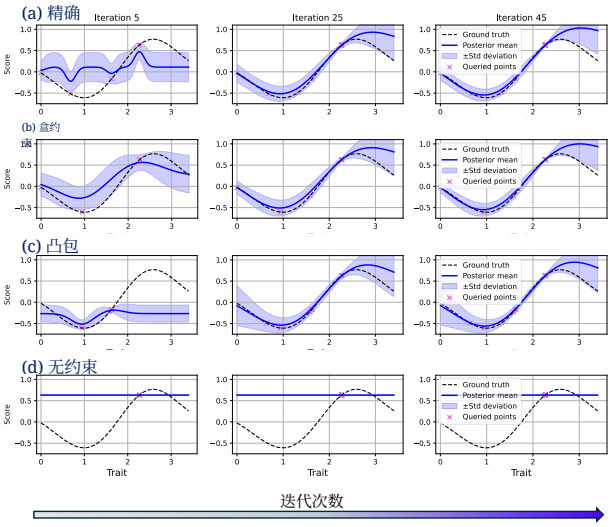


图 8. 在一个简单场景中迭代更新高斯过程的过程可视化，其中所有策略都学习将 3 个机器人分配到一个任务。为便于演示，仅考虑一种特征。我们展示真实的高斯过程、拟合函数（± 一个标准差）以及采样得到的特征值。除无约束策略外，其他策略均能够学习。

全面的定量数值模拟，展示不同策略下的遗憾值；以及 (iii) 在 RoboCup 搜救模拟器上的评估。对于 (i) 和 (ii)，真实的特质-效能映射函数从高斯过程中采样，其协方差由径向基函数（RBF）核指定。对于 (iii)，我们依靠模拟器根据机器人联盟及其能力返回一个分数作为标签。对于所有实验，式（14）中的整数二次规划均使用`cvxpy` (Diamond and Boyd 2016； Agrawal et al.2018) 求解。

我们首先讨论一个简单场景的数值模拟结果：将三台机器人（仅由一种特征表征）分配给一项任务。该任务的真实值特质-效能映射从高斯过程采样得到，并保持不变。机器人的特征值从 $[0,1]$中随机采样。为了深入了解不同策略在整个学习过程中的学习行为，我们对不同迭代次数下查询到的特征向量$\mathbf{Y}$ 以及拟合的高斯过程模型进行可视化；两者都会逐步更新。结果如图8所示。我们将高斯过程模型的均值显示为蓝色实线，方差显示为蓝色阴影，真实值特质-效能映射显示为黑色虚线，查询到的特征向量显示为紫色叉号。值得注意的是，即使经过 45 次迭代，无约束策略几乎仍未学到任何东西，而其他三种可实现性感知策略都能有效学习底层函数。无约束策略尝试的特征组合也明显更少，这表明其探索能力不足。这说明，无约束策略虽然能够采样出多样化的点集，却在整个特征空间中进行了无效探索。采样点可能与最近的可实现点相距甚远。从观察到的有害影响中，我们发现，若采样策略忽视可实现性，就会导致无效的多样性，从而妨碍高效学习。

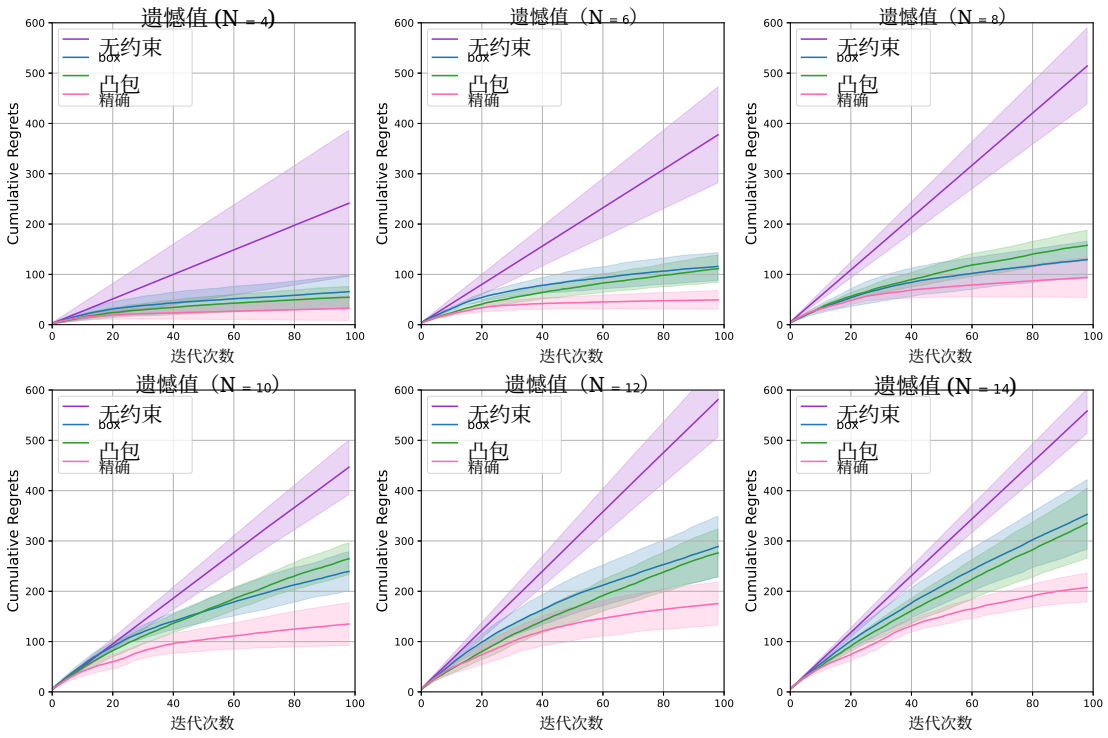
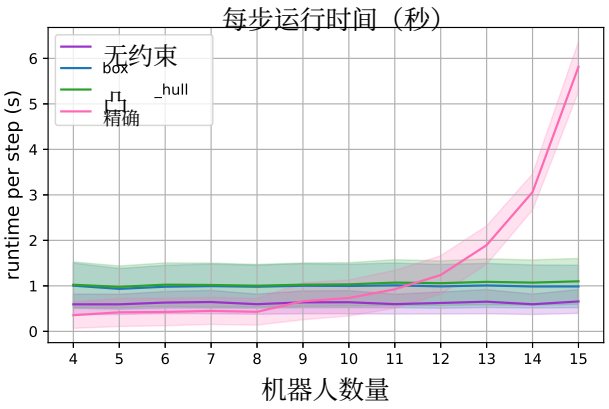


图9. 随着机器人数量增加，对四种方法进行比较的数值模拟结果。本实验包含三个任务。机器人和任务由4种不同特征建模。我们展示了5次试验的平均累积遗憾，阴影区域表示 ± 一个标准差。

基于对定性学习过程的认识，我们通过一系列实验全面比较四种策略；实验涉及 3 项任务，每项任务由 4 种不同特征表征。我们将待分配的机器人数量从 4 台调整到 15 台。由于各任务的真实值特质-效能映射已采样并已知，我们可以预先评估所有可能的分配方案，从而获得每项任务的真实最优分配。随后，我们可以利用它计算主动学习过程中的遗憾值。在图9中，我们展示了不同机器人数量下 5 次试验的平均累积遗憾。总体而言，我们观察到精确方法始终表现最佳，累积遗憾缓慢增长。相比之下，无约束策略的累积遗憾几乎呈线性增长，这表明它不断查询不具潜力的数据点。这一比较表明，忽视样本的可实现性、仅在事后通过投影进行补偿，效率非常低。与无约束策略相比，盒约束策略和凸包约束策略产生的累积遗憾显著较低，且性能与精确方法相近，尤其是在小规模问题中（有 ≤ 8 台机器人）。显然，通过引入可实现性意识，主动学习流水线实现了更合理的探索-利用权衡和更高的数据效率。我们还指出，在所有策略中，待分配的机器人数量越多，累积遗憾就越高。这是因为更大的团队规模自然会带来更大的组合空间，因此，针对奖励分配的主动学习也会更具挑战性。



使用sagej.cls图9所示实验的单步运行时间比较。结果表明，随着机器人数量增加，精确策略每步的运行时间急剧增长，而其他三种策略的运行时间则几乎保持不变。

我们还比较了四种策略的单步运行时间（见图 10）。无约束、盒约束和凸包策略的单步运行时间几乎保持不变，而机器人数量超过 10 个后，精确策略所需的时间则大幅增加。分配 15 个机器人时，精确策略运行单步几乎需要 6秒。计算时间急剧增加，是因为精确策略会检查每一种可能的分配及其对应的特征组合。为了评估一种特征组合，精确策略必须查询其高斯过程模型。由于可能的分配数量会随着机器人数量的增加而呈指数级增长，局部高斯过程查询的累积导致了观察到的巨大计算负担。

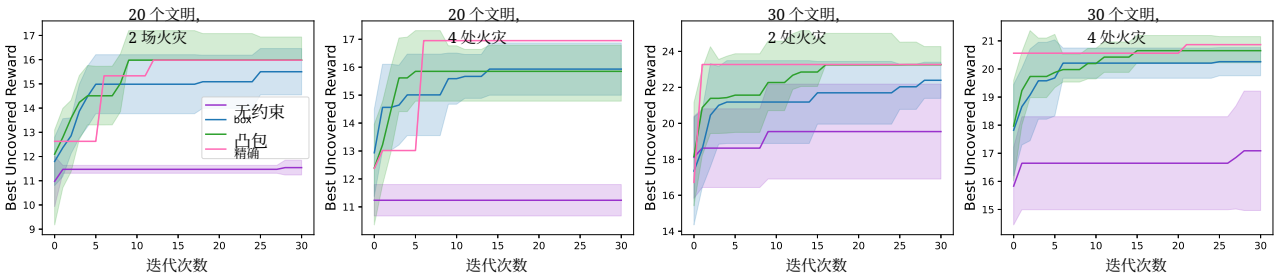


图11。 比较 RoboCup 搜救模拟器中四种约束策略下最佳未覆盖奖励（BUR）的变化。各子图表示四种不同的任务，其严重程度级别由需要救助的平民数量和火灾数量决定。对于每项任务和每种策略，我们运行主动学习模块 5 次，并以实线表示均值，以阴影区域表示标准差。

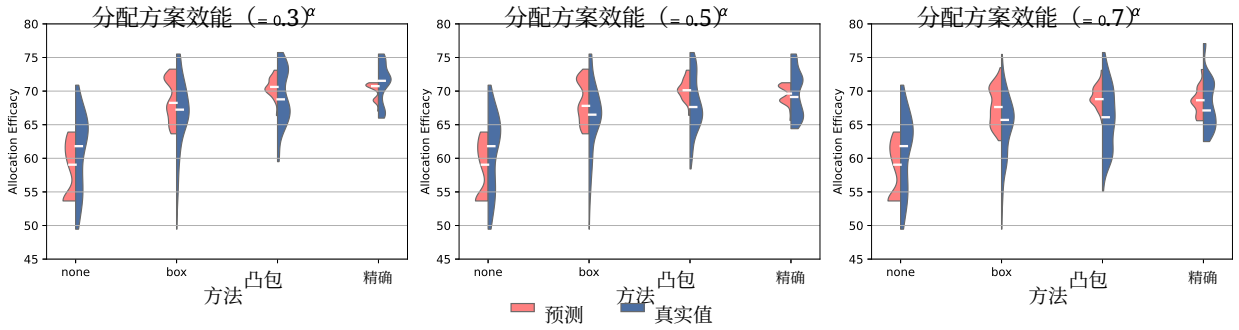


图12。 将学习到的模型应用于下游图搜索，并采用不同权重 $\alpha = 0.3$，0。5 和 0。7时，分配方案的总分配效能。所有红色小提琴图均依据模型预测的效能分数绘制，而所有蓝色小提琴图则依据真实值绘制。白色水平线表示效能中位数。

其他三种方法则避开了这一问题，因为它们在每次迭代中评估固定数量的候选项。结合图 9所示结果可知，对于只有少数机器人的小规模问题，精确策略更为可取。然而，随着团队规模扩大，盒约束策略和凸包约束策略凭借其较高的计算效率以及相当的学习效率和效果，成为更合理的选择。

6.5 整体评估

在最后一组实验中，我们通过学习特质-效能映射，并利用学习到的映射进行效能优化的任务分配，评估了我们方法的整体性能。

6.5.1 在机器人世界杯中学习特质-效能映射我们通过将主动学习模块与 RoboCup 搜救模拟器（Kitano and Tadokoro 2001；Kitano 等 1999；Federation 2025年）对接，进一步验证第 6.4 节中的发现（ RoboCup 搜索与救援模拟器（ Kitano and Tadokoro 2001；Kitano 等 1999；Federation 2025年）。RoboCup 模拟器允许我们更改待救援平民的数量和火灾数量，这些数量可作为任务严重程度和难度的替代指标。我们考虑三类作为待分配机器人的智能体：消防队、警察部队和救护车。它们分别负责扑灭火灾、清理受阻道路，以及将平民运送至医院。由于该模拟器不支持并行仿真，我们一次只仿真一个任务，并通过模拟器计算出的分数来衡量所分配联盟的性能（即效能）；分数越高，表示任务性能越好。我们分别仿真不同严重程度级别的任务。

由于我们无法再获取真实值特征效能图，因此我们报告最佳未覆盖奖励（BUR），作为评估四种采样策略的指标。BUR 是每次迭代中最高的未覆盖奖励，能够揭示各方法在识别高奖励机器人分配方面的效率。图 11 展示了四个相应任务的 BUR 结果。我们此前在数值模拟中观察到的无约束方法的相对劣势在此依然存在。总体而言，精确策略始终表现最佳，能够迅速发现可带来高分的联盟。盒约束和凸包约束策略的表现与精确策略相当，但有时会收敛到次优分配，或表现出更大的 BUR 方差。

6.5.2 利用学习到的 *STEAM* 特质-效能映射为了衡量主动学习方法对下游任务的影响，我们使用上一小节学习到的特质效能映射计算 NAC 启发式，以指导 E-ITAGS 的增量搜索。我们在 30 个问题实例中随机设置机器人的初始配置和任务配置，并改变任务时长和分配的时间预算 C_{max}。此外，我们还通过选择 $\alpha = 0.3$、0。5、0。7，重复进行了三次实验，以评估在分配效能和完工时间的相对优先级各不相同的情况下，学习到的效能模型在 E-ITAGS 中的表现。我们展示了真实值和

预测的总分配效能的半小提琴图，如图12所示。这些图使我们能够分析：在学习过程中，为提高计算效率而以不同程度近似可实现性约束，会如何影响任务分配过程中的下游任务性能。我们还可以看到不同方法的预测准确度。

正如预期，我们看到，精确采样策略产生的分配方案，其预测效能与真实值高度吻合。此外，与精确方法相关的解，其真实分配效能在四种采样策略中始终最高。值得注意的是，凸包约束策略紧随精确策略之后，其次是盒约束策略；这两种策略的总效能略有下降，可视为“近似的代价”。另一方面，无约束策略生成的分配方案效能最差，这表明，在学习特质效能映射时若完全不考虑实际可实现性约束，性能可能会显著降低。最后，随着我们调整超参数 α，这些趋势基本保持稳定；当我们通过较大的 α 更加重视时间预算而非任务分配效能时，各方法之间的差异略有缩小。

7 结论

在本研究中，我们提出了多机器人系统的时空效能优化分配（STEAM）问题，这是一种将分配效能与时间可行性明确结合起来的新问题表述。为应对这一挑战，我们开发了 E-ITAGS，这是一种交错图搜索算法，可在严格的时间预算下整合分配、调度与运动规划。不同于依赖脆弱假设（如任务分解方式或二元成功标准）的既有方法，E-ITAGS 利用通过可实现性感知主动学习模块学习得到的特质-效能映射，以刻画集体能力与任务性能之间细致且连续的关系。

我们的贡献主要有三点。首先，我们提出了一种可推广的形式化方法，将异构多机器人系统中的任务分配效能与时间可行性统一起来。其次，我们设计了一种计算高效的交错搜索过程，由新颖的启发式方法引导，在分配效能与完工时间之间取得平衡，并为次优性界提供相应的理论保证。第三，我们通过引入可实现性感知采样策略推进了多机器人任务分配中的主动学习，这些策略显著提高了组合领域中的数据效率和计算可处理性。

在仿真环境和应急响应领域开展的综合评估证实，与当前最先进的基线方法相比，E-ITAGS 始终能够在遵守严格时间与资源约束的同时，提供性能更高的分配方案。除性能提升外，我们的结果还证明，将主动学习整合到必须在任务需求知识有限的条件下运行的任务分配系统中是切实可行的。

总而言之，这些贡献使 E-ITAGS 成为一个稳健且可扩展的框架，能够在运行于严格时间和资源约束下的异构多机器人系统中实现时空协调。未来的工作将通过整合可实现性感知主动学习、大规模模拟器和真实场景部署，将这一框架扩展到更广泛的领域，从而在动态不确定环境中实现自适应的高质量性能。

致谢

本工作部分得到了美国陆军研究实验室项目资助，资助编号为 W911NF-17-2-0181（DCIST 合作研究协议）。

参考文献

Agrawal A、Verschueren R、Diamond S 和 Boyd S（2018）凸优化问题的重写系统。*Journalof Control andDecision* 5(1)：42–60。Baptista R 和 Poloczek M（2018）组合结构的贝叶斯优化。载于：*Internationalconference onmachine learning*。PMLR，第462–471页。Baras N 和 Dasygenis M（2019）机器人化仓库中异构车辆路径规划算法。载于：*5th PanhellenicConference onElectronics andTelecommunications*。Bechon P、Barbier M、Infantes G、Lesire C 和 Vidal V（2014）HiPOP：分层偏序规划。*STAIRS*：51–60。Bhargava N 和 Williams B（2019）多智能体析取时态网络。载于：*International Conference onAutonomous AgentsandMultiagent Systems*。Cannelli L、Zhu M、Farina F、Bemporad A 和 Piga D（2023）用于分布式黑箱优化的多智能体主动学习。*IEEE Control SystemsLetters* 7：1488–1493。DOI：10.1109/LCSYS.2023.3270347。Chakraa H、Gu´erin F、Leclercq E 和 Lefebvre D（2023）多机器人任务分配问题的优化技术：最新研究综述。*Robotics andAutonomous Systems*：104492。Choudhury S、Gupta JK、Kochenderfer MJ、Sadigh D 和 Bohg J（2022）不确定性和时间约束下的动态多机器人任务分配。*AutonomousRobots* 46(1)：231–247。Dai W、Rai U、Chiun J、Cao Y 和 Sartoretti G（2025）通过强化学习实现异构多机器人任务分配与调度。*IEEE Roboticsand Automation Letters*10(3)：2654–2661。DOI：10.1109/LRA.2025.3534682。Diamond S 和 Boyd S（2016）CVXPY：用于凸优化的嵌入式 Python 建模语言。*Journal ofMachine Learning Research*17(83)：1–5。Federation R（2025）机器人世界杯官方网站。https://www.robocup.org/。访问日期：2025年9月21日。Fu B、Smith W、Rizzo D、Castanier M、Ghaffari M 和 Barton K（2022a）学习多智能体任务分配中的任务需求和智能体能力。*arXivpreprint arXiv:2211.03286*。Fu B、Smith W、Rizzo DM、Castanier M、Ghaffari M 和 Barton K（2022b）能力不确定性下异构机器人团队的鲁棒任务调度。*IEEE Transactionson Robotics*39(2)。Gerkey BP 和 Matari´c MJ（2004）对多机器人系统中任务分配的形式化分析与分类。*International Journalof Robotics Research*。Giordani S、Lujak M 和 Martinelli F（2013）面向

移动机器人的分布式多智能体生产规划与调度框架。*Computers and Industrial Engineering*64(1)。DOI：10.1016/j.cie.2012.09.004。Gombolay M、Wilcox R 和 Shah J（2016）具有时空约束的多机器人团队快速调度。DOI：10.15607/rss.2013.ix.049。Guerrero J、Oliver G 和 Valero O（2017）带截止时间的多机器人联盟组建：复杂度分析与解。*PLoSONE* 12(1)。Hadfield-Menell D、Milli S、Abbeel P、Russell S 和 Dragan AD（2017）逆向奖励设计。载于：*Conference on NeuralInformation Processing System*。Igoe C、Ghods R 和 Schneider J（2022）多智能体主动搜索：一种强化学习方法。*IEEE Robotics andAutomationLetters*7(2)：754–761。DOI：10.1109/LRA.2021. 3131697。Jang D、Yoo J、Son CY、Kim D 和 Kim HJ（2020）利用分布式高斯过程进行多机器人主动感知与环境模型学习。*IEEE Robotics and AutomationLetters* 5(4)：5905–5912。DOI：10.1109/LRA.2020.3010456。Jose WJ 和 Zhang H（2024）异构多机器人协同调度中的动态子团队组建与自愿等待学习。载于：*2024IEEE International Conference onRobotics and Automation(ICRA)*。第4569–4576页。DOI：10.1109/ICRA57147.2024.10610342。Kailas S、Deolasee S、Luo W、Kim W 和 Sycara K（2025）整合多机器人自适应采样与信息性路径规划，用于时空自然环境预测。载于：*2025 IEEEInternational Conference on Robotics andAutomation (ICRA)*。第11413–11419页。DOI：10.1109/ ICRA55743.2025.11128099。Kitano H 和 Tadokoro S（2001）机器人世界杯救援：面向多智能体与智能系统的一项重大挑战。*AImagazine*22(1)：39–39。Kitano H、Tadokoro S、Noda I、Matsubara H、Takahashi T、Shinjou A 和 Shimada S（1999）机器人世界杯救援：将大规模灾害中的搜索与救援作为自主智能体研究领域。*Proceedings of theIEEE International Conferenceon Systems, Manand Cybernetics* 6：739–743。Koes M、Nourbakhsh I 和 Sycara K（2006）面向搜索与救援机器人技术的约束优化协调架构。载于：*IEEE ICRA*。Korsah GA、Kannan B、Browning B、Stentz A 和 Dias MB（2012）xBots：一种利用跨调度依赖关系生成和执行最优多机器人计划的方法。载于：*International Conference onRobotics and Automation*。Krizmancic M、Arbanas B、Petrovic T、Petric F 和 Bogdan S（2020）用于自动化施工任务的空地协同多机器人系统。*IEEERobotics andAutomationLetters* 5(2)：798–805。Le Ny J、Dahleh M 和 Feron E（2006）老虎机框架中的多智能体任务分配。载于：*Proceedings of the 45th IEEEConference on Decision andControl*。IEEE，第5281–5286页。Mallozzi P、Pardo R、Duplessis V、Pelliccione P 和 Schneider G（2018）MoVEMo：一种奖励函数工程化的结构化方法。载于：*IEEEInternationalConferenceonRobotic Computing2018*。Marino A、Restrepo E、Pacchierotti C 和 Giordano PR（2025）通过动态集群协议实现多智能体多资源分配的去中心化强化学习。*IEEE*《机器人与自动化快报》10(8): 8123–8130. DOI: 10.1109/LRA.2025.3581126. Matos D, Costa P, Lima J and Valente A (2021) 基于模拟退火算法的多移动机器人调度. 收录于：国际优化、学习算法与应用会议. Mayya S, D' Antonio DS, Saldana D and Kumar V (2021) 异构多机器人系统中的弹性任务分配.《机器人与自动化快报》. McCook CJ and Esposito JM (2007) 异构机器人集群的群集：军事车队场景. 收录于：年度东南系统理论研讨会论文集. pp. 26–31. Messing A and Hutchinson S (2020) 前向链式分层部分序规划. 机器人算法基础国际研讨会. Messing A, Neville G, Chernova S, Hutchinson S and Ravichandar H (2022) GRSTAPS：图形递归式同步任务分配、规划与调度.《国际机器人研究杂志》. Neville G, Chernova S and Ravichandar H (2023) D-ITAGS：弹性任务分配、调度与运动规划的动态交错方法. *IEEE*《机器人与自动化快报》8(2): 1037–1044. Neville G, Liu J, Chernova S and Ravichandar H (2024) Q-itags：具有时间预算的质量优化时空异构任务分配. *arXiv*预印本*arXiv:2404.07902*. Neville G, Messing A, Ravichandar H, Hutchinson S and Chernova S (2021) 基于特征的任务分配与调度的交错方法. 收录于：智能机器人与系统国际会议（*IROS*）. Neville G, Ravichandar H, Shaw K and Chernova S (2020) 异构多机器人团队的近似动态特征模型. *IEEE*智能机器人与系统国际会议. Nunes E, Manner M, Mitiche H and Gini M (2017) 具有时间与顺序约束的任务分配问题分类体系.《机器人与自主系统》90. Park J, Messing A, Ravichandar H and Hutchinson S (2023) 异构多机器人团队中的风险容忍型任务分配与调度. 收录于：*IEEE/RSJ*智能机器人与系统国际会议（*IROS*）. Prorok A, Hsieh MA and Kumar V (2017) 多样性对异构机器人集群最优控制策略的影响.*IEEE* 机器人学汇刊. Quinton F, Grand C and Lesire C (2023) 多机器人任务分配问题的市场方法：综述.《智能与机器人系统杂志》. Ravichandar H, Shaw K and Chernova S (2020) STRATA：面向异构智能体大型团队任务分派的统一框架.《自主智能体与多智能体系统》34(38). Rieskamp J, Busemeyer JR and Laine T (2003) 人们如何学习分配资源？两种学习理论的比较.《实验心理学杂志：学习、记忆与认知》29(6): 1066. Rieskamp J and Otto PE (2006) SSL：关于人们如何学习选择策略的理论.《实验心理学杂志：综合》135(2): 207.

Rold´an JJ, Garcia-Aunon P, Garz´on M, de Le´on J, del Cerro J and Barrientos A (2016) 用于绘制温室环境变量地图的异构多机器人系统. 《传感器》16(7). DOI:10.3390/s16071018. Schillinger P, Buerger M and Dimarogonas D (2018) 使用基于学习的滚动时域任务分配改进多机器人行为. *RSS* . Schlotfeldt B, Thakur D, Atanasov N, Kumar V and Pappas GJ (2018) 去中心化多机器人主动信息收集的任意时规划. *IEEE*《机器人与自动化快报》3(2): 1025–1032. DOI:10.1109/LRA.2018.2794608. Singh S, Srikanthan A, Mallampati V and Ravichandar H (2023) 多机器人任务分配中未知奖励的并发约束优化. *arXiv*预印本 *arXiv:2305.15288*. Slivkins A et al. (2019) 多臂老虎机导论. 《机器学习基础与趋势》12(1-2): 1–286. Srikanthan A a nd Ravichandar H(2022) 联盟形成中异构策略的资源感知型自适应. 收录于：自主智能体与多智能体系统国际会议（ *AAMAS*）. pp. 1732–1734. Srinivas N, Krause A, Kakade SM and Seeger M (2009年a) 以老虎机设定进行高斯过程优化：无悔与实验设计. *arXiv*预印本 *arXiv:0912.3995*. Srinivas N, Krause A, Kakade SM and Seeger M (2009年b) 以老虎机设定进行高斯过程优化：无悔与实验设计. *arXiv* 预印本 *arXiv:0912.3995*. Stadler M, Banfi J and Roy N (2023) 在不确定图中近似协作团队行动的价值，以实现高效的多智能体导航. 收录于：自动规划与调度国际会议论文集, 第33卷. pp. 677–685. Stroupe AW, Huntsberger T, Kennedy B, Aghazarian H, Baumgartner ET, Ganino A, Garrett M, Okon A, Robinson M and Townsend JA (2005年) 用于装配与维护的异构机器人系统. 欧洲航天局，*ESA SP*(603): 625–631. Tzes M, Bousias N, Chatzipantazis E and Pappas GJ (2022) 用于多机器人主动信息获取的图神经网络.*arXiv*预印本 *arXiv:2209.12091*. Wang Z, Liu C and Gombolay M (2022) 用于满足时空约束的可扩展多机器人调度的异构图注意力网络. 《自主机器人》46(1): 249–268. Weld DS (1994年) 最小承诺规划导论.《人工智能杂志》15(4): 27–27. Whitbrook A, Meng Q and Chung PW (2015年) 面向时间关键型多智能体系统的新型分布式调度算法. 收录于：*IEEE*智能机器人与系统国际会议，第2015-Decem卷. pp. 6451–6458. Zhao W, Meng Q and Chung PW (2016年) 启发式分布式任务分配方法（PIA）. *IEEE*《控制论汇刊》46(4): 902–915.

从凸包中采样

回顾第5节，对于凸包约束策略，我们从 $\mathcal{S} = [0,1]^{M\times N}$ 中采样 S，然后得到分配给所有任务的特征 Y $=$S$\boldsymbol{Q}$。在这里，我们证明，以这种方式得到的 Y 位于 $\mathcal{Y}_{\boldsymbol{Q}}$ 中可实现点集的凸包内。对于 M 个任务，若要分配 N 个机器人，则 $\mathcal{Y}_{\boldsymbol{Q}}$ 在特征空间中包含 $K = M \cdot 2^N$ 个顶点。注意， K 相对于 M 并非指数级增长，因为任务不一定会同时发生。我们将这些 K 个点记为 $\mathrm{Y}^1, \mathrm{Y}^2$、.....、$\mathrm{Y}^K$。相应地，我们定义 A^k 为满足 $\mathrm{Y}^k = \mathrm{A}^k\boldsymbol{Q}$ 的分配矩阵。注意，在分配空间中，存在一个由 $\{\mathrm{A}^k\}_{k=1}^K$ 构成的凸包，我们称之为 $\mathcal{CH}_{\mathbf{A}}$。根据松弛技术的性质，对于任意采样得到的 S，我们知道它位于 $\mathcal{CH}_{\mathbf{A}}$ 内，因为我们总能找到一组非负实数 λ_1、λ_2、.....、λ_K，满足 $\sum_{k=1}^K \lambda_k = 1$ 且 S $= \sum_{k=1}^K \lambda_k$ A^k。因此，我们知道：

$$\begin{aligned}\mathbf{S}\boldsymbol{Q} &= (\sum_{k=1}^{K} \lambda_k \mathbf{A}^k)\boldsymbol{Q} \\ &= \sum_{k=1}^{K} \lambda_k \mathbf{A}^k\boldsymbol{Q} \\ &= \sum_{k=1}^{K} \lambda_k \mathbf{Y}^k.\end{aligned} \tag{18}$$

公式 18 左侧反映了我们如何在凸包约束策略中采样一个点，而根据定义，右侧显然位于 $\mathcal{Y}_{\boldsymbol{Q}}$ 的凸包内。因此，我们证明了将 $\mathbf{Y}$ 生成为 $\mathbf{Y}$ $=\mathbf{S}\boldsymbol{Q}$ 可确保其位于 $\mathcal{Y}_{\boldsymbol{Q}}$的凸包内。