分类号 TP18 学号 12039002

UDC 密级 公开

工学博士学位论文

无人机集群系统侦察监视任务规划方法

博士生姓名 陈少飞

学科专业 控制科学与工程

研究方向 任务规划技术

指导教师 沈林成 教授

国防科学技术大学研究生院

二〇一六年六月

Planning for Reconnaissance and Monitoring using UAV Swarms

Candidate: Shaofei Chen

Supervisor: Professor Lincheng Shen

A dissertation

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Engineering

in Information and Communication Engineering

Graduate School of National University of Defense Technology

Changsha, Hunan, P. R. China

June 6, 2016

独 创 性 声 明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科学技术大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目：无人机集群侦察监视任务规划方法

学位论文作者签名：[signature]　　日期：2016 年 4 月 11 日

学位论文版权使用授权书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密学位论文在解密后适用本授权书。）

学位论文题目：无人机集群侦察监视任务规划方法

学位论文作者签名：[signature]　　日期：2016 年 4 月 11 日

作者指导教师签名：[signature]　　日期：2016 年 4 月 11 日

目 录

表 目 录

图 目 录

算 法 目 录

摘 要

如何将无人机集群系统部署于大范围环境中进行侦察监视，是未来无人机军事应用的重要问题之一。一方面，环境中往往分布着大量动态变化的子目标 / 子任务，亟需自动规划算法，实现无人机集群系统在不确定条件下进行连续侦察监视的快速规划；另一方面，无人机在复杂的环境中进行搜索时，往往需要人辅助提供一些关于环境的知识，所以需要设计良好的人与无人机进行交互的方式，实现在人辅助下进行搜索。基于此，论文的主要工作和创新点如下：

（1）针对具有子模性规划目标的多智能体部分可观马尔科夫决策过程（Multi-Agent Partially Observable Markov Decision Process, MPOMDP），首次提出了一种近似最优的多智能体在线规划算法。这种算法通过顺次分配技术（Sequential Allocation Technique）来依次计算每个智能体的策略，贪婪地最大化单个智能体对团队任务目标的边际贡献（Marginal Contribution），从而避免了直接考虑团队的联合策略（其导致的计算代价与智能体个数呈指数关系），使得计算复杂度随智能体个数呈多项式关系。论文通过理论证明该算法具有很好的近似最优性能。创新性工作为：使用顺次分配技术来计算智能体的策略，相比于其他的搜索团队联合策略空间的方法，这种方法具有很好的可扩展性，并能够满足问题的实时性要求。

（2）针对传递函数解耦的部分可观马尔科夫决策过程（Transition-Decoupled POMDP, TD-POMDP），首次提出了具有良好可扩展性的在线规划算法——传递函数解耦的部分可观蒙特卡洛规划（Transition-Decoupled Partially Observable Monte-Carlo Planning，TD-POMCP），即一种基于蒙特卡洛树搜索（Monte Carlo Tree Search, MCTS）和 max-sum 的分散式在线算法。TD-POMCP 的创新主要包括：1）根据局部智能体团队的局部联合动作和联合观测，利用 TD-POMDP 模型中变换依赖的弱耦合关系，对每个智能体分别构建一个前向搜索树；2）基于 MCTS 这种基于采样的搜索算法对每个智能体的搜索树进行扩展和更新；3）所有智能体通过分散式交互的方式同时进行对各自搜索树的扩展和更新。在搜索树中的每个规划步骤，通过 max-sum 这种分散式任意时间近似算法进行智能体的动作选择。论文通过理论证明这种方法具有很好可扩展性、鲁棒性、实时性和最优性，并且通过相关的仿真实验证明论文的方法可以成功应用于大规模 TD-POMDPs。

（3）基于 MPOMDP 和 TD-POMDP 模型，分别建立了在威胁环境和不确定性下的无人机集群系统进行侦察监视的集中式和分散式问题模型。这些问题的状态空间大小随环境中监视子目标个数的增加呈指数增长，使得通用的规划求解器无

法求解这种大范围的侦察监视问题，更无法满足实际任务的实时性要求。因此，论文设计了随监视子目标个数线性增长的信念状态表示方式，并进一步定义了无人机集群系统侦察监视问题的形式化模型。基于此，设计了仿真实验，实验结果表明论文的算法能够成功应用于大规模无人机团队在大范围环境中进行集中式/分散式侦察监视的规划问题。与现有的侦察监视问题的求解方法相比，论文设计的问题模型与求解算法的创新点包括：1）基于智能体模型描述了无人机与环境和与其他无人机之间的交互方式；2）环境的模型能够表达其在时间和空间上的部分可观和非静态性质；3）求解算法在进行蒙特卡洛仿真时，通过不断保持对健康约束的检测，在规划中能够处理无人机可能遭受的来自威胁的伤害。

（4）针对概率知识和人辅助下进行搜索的决策问题，建立了一个新颖的模型，即机器人和人的搜索问题（Robot-Human Search, RHS）。RHS 描述了一类不确定知识下的搜索问题：一架自主无人机在人的有限辅助下，搜索环境中的某一物体（如一架坠落战机的碎片）。环境中的不确定性体现在物体的回报值以及人的可利用性（Availability）。无人机的目标是最大化所获物体的回报值并最小化总的搜索代价。论文证明这一结合人辅助的搜索问题在多项式时间内可解，这一结论在之前的文献研究中并未得出过。进一步，通过仿真实验来验证论文提出的方法，实验结果证明论文的方法显著优于一些基准算法。

关键词: 无人机集群，侦察监视，多智能体系统，不确定规划，马尔科夫决策理论，协调与合作，潘多拉问题和潘多拉规则，人与机器人交互

ABSTRACT

The approach for deploying Unmanned Aerial Vehicle (UAV) Swarms to reconnoiter and monitor environments, is a key challenge for applying UAVs in future military missions. On the one hand, many dynamic sub-targets/sub-tasks are distributed in the environment, and auto-planning methods are needed to solve problems of many UAVs continuously monitoring the environment under uncertainty. On the other hand, for searching in complex environments, UAVs always need some prior knowledge about the environment that may provided by humans, well designed mechanism for the interactions between human and UAVs are then needed to implement that searching under human help. Given this, the main contributions of this dissertation are as follows:

(1) For general Multi-Agent Partially Observable Markov Decision Processes (MPOMDPs) with submodular objectives, we first propose a multi-agent online planning algorithm with bounded optimal guarantee. Instead of using joint actions and joint observations, our algorithm sequentially computes the policy of each agent by greedily maximising its marginal contribution. Although this method seems simply, we theoretically prove that it grantees approximation optimal performance. The novelty of this algorithm is that we use the sequential allocation technique to solve the joint reconnaissance and monitoring problems. Compared with algorithms that search joint policy space, our algorithm scales better and satisfies real time limitations.

(2) For general Transition-Decoupled POMDPs (TD-POMDPs), we first proposed a scalable decentralised online planning algorithm, Transition-Decoupled Partially Observable Monte-Carlo Planning (TD-POMCP), based on Monte Carlo Tree Search (MCTS) and max-sum. The novelty of TD-POMCP is as follows: 1) given the weekly coupled relationship among agents in TD-POMDP, we construct a search tree that features the joint actions and joint observations of a local team of each agent; 2) using MCTS to expand and update every search tree, i.e., at every planning step in an agent's search tree, we use max-sum, a decentralised anytime approximation algorithm to select agents' actions. We theoretically prove that our algorithm is scalable, robust, real time and optimal, and empirically show that it can be applied to large TD-POMDPs.

(3) We formulate the reconnaissance and monitoring problem for many UAVs under uncertainty and threats by MPOMDP and TD-POMDP, respectively. In these problems,

the state space increase exponentially with the number of targets, making general planners are intractable to solve large instances and cannot satisfy real time limitations. Thus, we designed a belief representation whose dimension increases linearly with the number of targets. Given this, we designed simulated experiments to evaluate our methods and show that our algorithms can be used to solve large problems with many agents. Given state-of-art models of reconnaissance and monitoring problems, the novelty of our model is as follows: 1) given agent based model, it described the actions of UAVs and the interactions between UAVs and environments; 2) the environment model features the temporal and spacial partial observability and non-stationary; 3）the algorithms check health constraints during Monte Carlo simulations to take threats to UAVs into account.

(4) For searching under uncertain knowledge and human help, we propose a new model, Robot-Human Search (RHS). RHS formulates a search problem involving a robot that is searching for a certain item in an uncertain environment (e.g., searching an important piece of plane debris in a flight crash region) that allows only limited interaction with humans. The uncertainty of the environment comes from the rewards of undiscovered items and the availability of costly human help. The goal of the robot is to maximize the reward of the collected items while minimising the search costs. We show that this search problem is polynomially solvable with a novel integration of the human help, which has not been studied in the literature before. Furthermore, we empirically evaluate our solution with simulations and show that it significantly outperforms several benchmark approaches.

Key Words: UAV Swarms, Reconnaissance and Mornitoring, Multi-Agent Systems, Planning Under Uncertainty, Markov Decision Making Theory, Coordination and Cooperation, Pandora's Problem and Pandora's Rule, Human Robot Interaction

主要符号使用说明

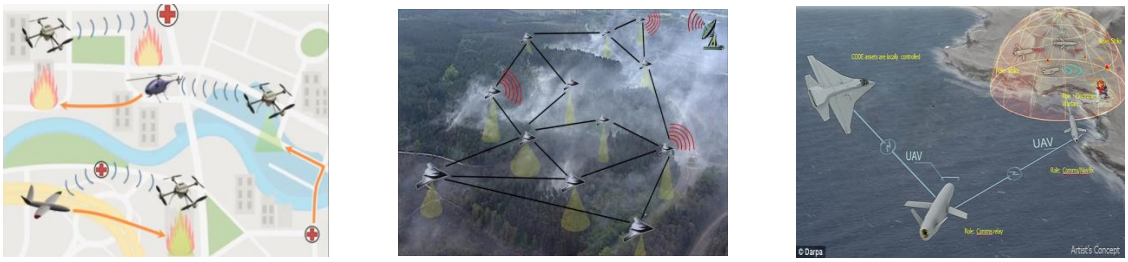
MDP	马尔科夫决策过程
POMDP	部分可观马尔科夫决策过程
MPOMDP	多智能体部分可观马尔科夫决策过程
Dec-POMDP	分散式部分可观马尔科夫决策过程
TD-POMDP	传递函数解耦的部分可观马尔科夫决策过程
TD-POMDP-HC	基于健康约束的 TD-POMDP
DCOP	分布式约束优化问题
MCTS	蒙特卡洛树搜索
POMCP	部分可观蒙特卡洛在线规划
TD-POMCP	传递函数解耦的部分可观蒙特卡洛在线规划
FMOP	基于特征信念的蒙特卡洛在线规划
TD-FMOP	传递函数解耦的基于特征信念的蒙特卡洛在线规划
HRI	人与机器人交互
RHS	机器人 - 人的搜索问题
$\mathcal{M}$	智能体集合
$\mathcal{S}$	有限的系统状态集合
s	系统状态 $s \in \mathcal{S}$
$\mathcal{A}$	全体智能体的联合动作的集合
a	全体智能体的联合动作 $a \in \mathcal{A}$
$\mathcal{O}$	全体智能体的联合观测的集合
o	全体智能体的联合观测 $o \in \mathcal{O}$
$\mathcal{T}$	状态转移概率的集合
Ω	观测概率集合
r	回报函数

T	规划的时间长度
π	全体智能体的联合策略
h	全体智能体的联合动作和联合观测序列的历史
$\Psi(h)$	全体智能体的联合策略在历史 h 上的特征信念向量
$V^{\pi}(h)$	全体智能体的联合策略 π 在历史 h 上的期望值
$V^{\pi}(h,a)$	全体智能体的联合策略 π 在历史 h 上执行动作 a 的期望值
$\mathcal{S}_m$	智能体 A_m 的局部状态集合
s_m	智能体 A_m 的局部状态 $s_m \in \mathcal{S}_m$
$\mathcal{A}_m$	智能体 A_m 可能采取的动作集合
a_m	智能体 A_m 可能采取的动作 $a_m \in \mathcal{A}_m$
Ω_m	智能体 A_m 可能获得的观测集合
o_m	智能体 A_m 可能获得的观测 $o_m \in \mathcal{O}_m$
r_m	智能体 A_m 的回报值函数
π_m	智能体 A_m 的策略
h_m	智能体 A_m 的动作和观测序列的历史

第一章　绪论

1.1　引言

近些年，无论在科学研究还是在应用市场中，无人机受到了越来越广泛的关注。同时，随着相关技术的快速发展，无人机的自身能力不断增强且性价比显著提高，加速了其在相关民用和军事领域的应用。然而，受限于当前的科学技术水平，目前无人机的应用主要为采用少量无人机来尝试完成在相对简单环境中的任务。对于复杂环境中的任务，特别是需要派出无人机集群系统时，如何通过集群的协调配合以发挥其优势来完成任务，已成为当前科学研究的一项重要挑战 [1]。此外，无人机集群系统在搜索救援、环境监视和军事作战等（如图1.1给出了灾难响应、森林火情监视和军事目标打击的例子）等领域具有广阔的应用前景，世界各国均投入大量的科研力量进行研究，比如美国海军计划于今年夏天进行首飞的其首支无人机集群系统 LOCUST [1]。论文针对无人机集群系统在复杂环境中侦察监视的任务规划问题展开研究，目标是设计规划算法，输入问题的模型，通过计算自动输出无人机所能执行的行动序列。



(a) 灾难响应　(b) 森林火情监视　(c) 军事目标打击

图 1.1　无人机集群的应用想定

1.2　问题提出

单 / 多无人机侦察监视问题的决策和规划，是人工智能和机器人研究领域的重要理论和应用问题之一。如图1.2所示，针对来自于真实世界的具体问题，规划器首先通过规划得到智能体的行为策略，然后通过控制器执行智能体的行动并与环境进行交互。然而，在对问题进行建模时，传统规划方法往往假设任务及环境具有完全可观和静态的特征。近些年，马尔科夫决策理论得到了快速发

[1]http://www.onr.navy.mil/Media-Center/Press-Releases/2015/LOCUST-low-cost-UAV-swarm-ONR.aspx

展，为侦察监视问题的不确定性规划，提供了具有理论支撑的形式化描述手段。具体来说，马尔科夫决策理论的相关模型，主要包括用于单智能体规划的马尔科夫决策过程（Markov Decision Process, MDP）和部分可观马尔科夫决策过程（Particularly-Observable MDP, POMDP），以及用于多智能体规划的多智能体部分可观马尔科夫决策过程（Multi-Agent POMDP, MPOMDP）和分散式部分可观察的马尔科夫决策过程（Decentralised POMDP, Dec-POMDP）。其中，MDP 和 POMDP 的相关算法已经满足了求解较大规模单智能体问题的需要，例如 POMCP 算法 [2] 实现了在线求解大规模 POMDP 这一难题。MPOMDP 和 Dec-POMDP 进一步提供了多智能体决策和规划的框架，具有非常丰富的表达能力，能够对大多数的多智能体合作规划问题进行建模。从而，无人机集群任务规划问题可以采用 MPOMDP 和 Dec-POMDP 进行形式化描述，来表达问题在不确定性规划框架下的各个方面的属性（比如智能体的动作和观测，相互间的交互与协调，环境的动态及部分可观的性质，规划目标等）。（马尔科夫决策理论的介绍详见第1.3.2节。）

图 1.2　规划器求解规划问题的过程

然而，关于多智能体系统（Multi-Agent System）如何进行分布式决策的理论研究仍然处于起步阶段，目前只有当问题的规模很小时才能够实际可解 [3]。具体地，在多智能体规划问题（包括 MPOMDPs 和 Dec-POMDPs）中，每个智能体在进行决策时，不仅需要考虑环境和智能体自身的模型，而且需要考虑其他智能体可能的执行策略，使得问题具有高度复杂的策略空间（随观测空间大小呈指数增长，并随智能体个数和规划步长呈双指数增长），随着问题规模的增加很快变得很难求解。从而，如何求解大规模 MPOMDPs 和 Dec-POMDPs 成为近几年学术界研究的热点和难点问题之一。虽然出现了一系列方法进行尝试，但是目前的这些算法只能针对小规模的问题实例进行求解，仍然无法求解大规模 MPOMDPs 和 Dec-POMDPs（如无人机集群系统侦察监视问题）。因此，论文面向无人机集群系统侦察监视任务规划问题，设计能够解决这种大规模的 MPOMDPs 和 Dec-POMDPs 问题的在线求解方法。

此外，当前无人机任务规划系统的自主（规划）能力，仍然与科学家和工程师们最初对这些系统的定位和期望有着很大的差距。在针对复杂环境的实际应用中，为了更有效地完成任务，往往需要人提供一些知识来辅助无人机进行决策。

因此，论文研究在线规划方法，一方面为学术界提供具有理论支撑的求解大规模侦察监视问题的方法；另一方面对于大规模无人机进行侦察监视问题的解决，

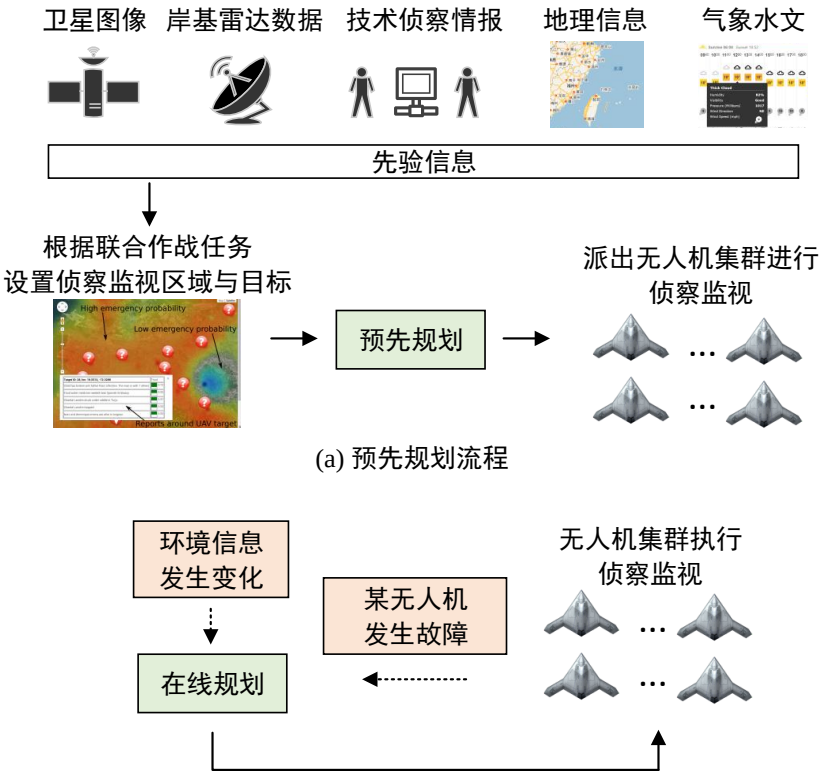


图 1.3　无人机集群进行海空情侦察监视的任务规划流程

将带来非常重要的军事应用。下面以海空情侦察监视为例，介绍无人机集群系统侦察监视任务规划过程的一种典型想定。

对于领海区域面临着的随时可能出现的军事威胁，如何利用装配的军事资源（包括航天卫星、岸基雷达和无人机等），对该区域海面与空中的情况（简称海空情）进行连续的侦察监视，成为一项重要课题。在海空情侦察监视中，首先，在通常条件下，覆盖该区域的航天卫星和岸基雷达能够提供一定程度的监视信息。其中，航天卫星能够提供该区域全局的海空情信息（包括电磁信息和图像信息）用于描绘出该区域的粗略的态势；岸基雷达能够对该区域可能出现的敌方目标（包括军舰或飞机）进行全面细致的连续监视。然而，这些敌方军事目标往往具有很强的隐蔽性。一方面，敌方目标往往混杂于民用舰船或飞机中，利用这些船只或飞机进行伪装；另一方面，敌方可能会采用强烈的电磁干扰来压制航天卫星和岸基雷达的正常工作。这些隐蔽手段使得航天卫星和岸基雷达所提供的信息，只能提取出模糊的、（空间上）局部的和（时间上）间断的海空情。因此，需要进一步主动派出无人机团队，灵活地、近距离地进行海空情侦察监视，即派出以集群的方式执行任务的无人机团队，在参谋人员的协助下，对大范围的海面以及空中区域中的大量可疑目标，详细地、动态地进行“搜索发现、识别查证和跟踪监视”。

无人机集群系统海空情侦察监视的任务规划流程如图1.3所示。具体流程如下：1）提取和构建规划问题的先验信息。这些信息来源于不同方面，包括上文提到的航天卫星和岸基雷达所获得的信息，以及技术侦察手段间接地、迂回地获得的相关情报。2）指挥员与领域专家结合这些先验信息，以及监视区域的地理信息和气象水文等数据，根据联合作战任务，分析制定需要搜索发现、识别查证和跟踪监视的，分布于区域中的大量的可疑的目标区域。3）指挥机构派出有限数量的无人机资源，对确定出的各个地点及周边区域进行连续的侦察监视。因此，在执行任务时，无人机团队需要快速部署于该区域中，一方面通过不断移动和访问来发挥各自的能力，另一方面通过协调配合完成整个监视任务。在进行侦察监视之前，通过预先规划器，规划出详细的任务计划。在任务执行中可能出现的意外情况，如环境信息的变化，以及监视团队成员可能发生故障退出的情况，在线规划器[2] 通过快速计算，重新计算或者调整任务计划。

此外，无人机对信息的理解能力和决策能力的仍然具有限，常常需要人提供一些知识。然而，鉴于参谋人员很难在所有时间监控集群中的所有无人机，如果单个无人机在决策过程中能够自主决定何时向参谋人员进行求助，将能够实现更为有效的决策。因此，通过无人机之间的良好合作，以及和参谋人员的交互，针对有限的无人机集群资源进行在线规划，制定其行动策略，将在很大程度上提高整个任务的侦察监视和搜索的效能。

基于以上背景，求解无人机集群侦察监视问题的难点如下：

- **性能保证**：即使只是单个无人机的规划（如任务规划和路径规划），在计算求解上都是非常困难的问题。一般地，可能的路径或计划个数，随着规划步数呈指数关系增长，使得通过比较所有可能解的方法在计算上并不可行。因此，规划方法应具有一定的最优性能保证，即保证生成的规划方案具有最优或者近似最优的性能。
- **可扩展性**：在无人机集群（即大量无人机）系统中，因为无人机之间需要相互协调进行各个子任务的规划，从而规划问题的规模随智能体个数呈双指数增长，所以规划算法应该具有有效降低搜索空间规模的能力，以处理大量无人机交互协调问题。
- **实时性**：由于在执行任务时往往伴随着一些突发事件或紧急事件，比如某个环境子目标突然出现或突然消失，或者指挥员命令追踪 / 打击某个子目标，以此规划方法应具有实时的计算求解能力，能够在线应对可能发生的变化。

[2] 预先规划（离线规划）在离线条件下计算出整个任务计划，而“在线规划”，作为人工智能和决策学科领域的专有名词，表示智能体在每个单个的决策周期内交替进行规划和（计划的）执行。

- **鲁棒性**：无人机集群系统的构成可能会发生变化，如团队中某个无人机毁伤或者被命令撤离这项任务，或者有新的成员加入团队来帮助应对更为复杂的环境。因此，规划方法需要在损失某个成员时，由余下成员构成的团队来继续完成任务；并在新的成员加入后，通过团队成员间的协调更有效地完成任务。
- **可执行性**：当环境中分布的大量威胁时，考虑无人机需要深入危险的环境中执行任务，在规划时需要考虑智能体对于威胁的承受程度；否则，规划生成的策略很可能在实际任务中无法有效执行。因此，规划方法需要平衡无人机所能够获得的信息价值与可能遭受的伤害的关系。

针对以上挑战和困难，论文将针对无人机集群系统的侦察监视问题，重点研究两种特殊类型的针对大规模多智能体 POMDPs 的具有良好扩展性在线求解方法，以及单个无人机在人辅助下的搜索方法。具体地，论文开展以下几个方面的研究：

(1) **研究 MPOMDP 问题的在线求解算法**。MPOMDP 为全局通信方式下的多智能体马尔科夫决策过程。对于这种一般不存在稀疏交互性质的多智能体问题，其巨大的策略空间将无法进行分解：即问题的状态、动作和观测空间很难从某个角度得到解耦，从而随着智能体的个数的增加呈双指数增长。论文研究近似在线求解算法，实现求解大规模的 MPOMDPs。研究在保证近似最优性能指标的同时，如何有效减小每个智能体在搜索过程中的策略空间，使其随智能体个数的增加由双指数增长下降为多项式增长。

(2) **研究针对传递函数解耦的部分可观马尔科夫决策过程（Transition Decoupled POMDP, TD-POMDP）问题的分散式在线求解算法**。TD-POMDP 是一类特殊的 Dec-POMDPs，只允许智能体之间局部通信，而且智能体具有稀疏交互特性（即每个智能体只需要很少的交互伙伴）。与一般的 Dec-POMDPs 相比，TD-POMDPs 在保证最优性的同时具备更好的可扩展性。论文研究分散式在线算法，实现求解大规模 TD-POMDPs，即避免使用所有智能体的联合策略进行决策，而是研究通过局部决策和智能体之间的分散式交互实现对 TD-POMDPs 进行全局规划的方法：任务执行过程中，每个智能体通过不断更新关于局部环境的信念状态，并与其他相邻的少量智能体进行交互，然后每个智能体进行局部决策，并通过分散式规划算法，使得智能体系统通过这种分散式协调来完成对全局任务目标的规划。

(3) **建立无人机集群侦察监视的问题模型**。首先通过定义执行任务的环境模型以及无人机的智能体模型，对侦察监视问题进行建模。然后通过 TD-POMDP

和 MPOMDP 对两种无人机集群侦察监视问题进行形式化描述。进一步设计仿真评价方法，对论文中的在线规划算法进行评价。

(4) **建立人辅助下的搜索问题模型和求解方法**。建立如下的问题模型：在不确定的环境下，无人机需要自主决策进行搜索，同时在每个时刻，根据自己的信念判断是否需要向人求助某些知识。进一步，设计这类搜索问题的求解算法并进行仿真实验评价。

1.3 研究现状及基础

如上所述，在威胁环境和不确定条件下无人机集群系统侦察监视问题的求解方法需要具有处理多智能体协调规划的能力、处理动态任务环境中的不确定性和约束条件的能力，以及人与无人机协作的能力。本节对相关研究的现状和基础进行综述，首先综述无人机集群侦察监视问题的相关研究，然后综述多智能体规划相关问题的理论研究现状
（包括不确定性规划和多智能体系统协调与合作），最后综述人与机器人协作搜索问题的研究现状。

1.3.1 多无人机侦察监视

使用无人机集群系统进行联合侦察监视的规划问题，是当前联合任务规划系统研究与开发的典型问题之一。目前采用的规划方式，主要是指挥员与军事专家，结合相关的（包括环境以及敌我双方的武器装备）数据模型，根据指挥经验，进行规划。对于战役 / 战术层级的规划问题，由于问题规模巨大，采用这种方式进行规划均需要很长的作业时间。基于人工智能的自动化规划技术，能够有效处理大规模的规划问题，在战争中能够快速提供给指挥员一套或几套有效的作战计划。近些年，飞行器任务规划技术的研究取得了一系列的重要进展 [4–11]，为无人机集群的规划方法奠定了一定的技术基础。然而，与本课题研究的大规模无人机团队进行侦察监视直接相关的规划方法仍处于空白状态。

当前学术界的研究，对于未考虑环境威胁的，智能体侦察、感知和收集环境态势的问题和方法，可以归类为移动智能体信息收集问题（Information Gathering Problems）[12]。在这类问题中，智能体连续采集并实时提供环境中的态势 / 状况。对于动态环境，之前的研究 [12–14] 考虑完全可观的（即智能体可以直接观测环境的基本状态）静态的（即状态的联合概率分布随时间固定不变）模型。文献 [15] 考虑智能体只能获取当前位置的具体状态，提出了一个部分可观的问题模型。针对巡逻过程中存在的拥有决策能力的入侵者 / 逃跑者尝试突破防线 / 摆脱追捕的问题，当前有一系列基于博弈理论的方法 [16–18] 研究如何利用有限的资源进行

表 1.1　信息收集问题相关算法考虑的问题性质

算法	性质				
	在线	连续监视	时空动态性	非静态	策略对手
Singh et al. [25]					
Singh et al. [26]	√				
Meliou et al. [27]	√		√		
Paruchuri et al. [28]					
Tsai et al. [29]					√
Basilico et al. [30]		√	√		√
Agmon et al. [31]		√	√		√
Elmaliach et al. [32]	√	√	√		√
Grocholsky et al. [33]	√	√			
Fiorelli et al. [34]	√	√			
Martinez-Cantin et al. [35]	√	√			
Ahmadi and Stone [36]	√	√	√		
Jerome et al. [15]	√	√	√	√	
Stranders et al. [37]	√	√	√		
Stranders et al. [38]	√	√	√		
Stranders et al. [12]	√	√	√		

巡逻，来保护环境中的一些重要目标。与论文考虑收集环境的信息所不同，这类问题主要的挑战是研究如何最大程度的监视和抓获入侵者 / 逃跑者从而减少损失。而且，这些方法均未考虑在巡逻过程当中智能体自身可能遭受的损害。

对于侦察监视问题的大多数研究，主要考虑静态的威胁和环境。文献 [14] 研究了针对具体的动态环境现象（如湖水赤潮的增长和河水中盐的浓度）进行信息收集的问题。但是这种动态性与本论文中考虑环境的随机性并不相同。对于环境的随机性，马尔科夫模型（Markov Models）广泛应用于对非静态随机的环境对象的建模，例如飞行器或传感器的所要探测的地面目标 [15, 19–21]，无线网络的物理行为 [22]，通信系统频道的存储容量 [23, 24] 等等。然而，这些研究中所使用的马尔科夫模型均额外添加了一些不同的假设条件。例如，文献 [15, 23] 中假设每个目标在不同时刻只在两个状态之间切换，文献 [24] 中的马尔科夫模型矩阵必须同时满足其中列出的四个条件。本研究中的模型使用更为通用的马尔科夫模型对环境中的信息和威胁进行建模。

下面，将对近些年学术界对于一般的多智能体信息收集问题相关研究的理论与方法研究进行归纳总结。参考文献 [12] 中的综述方式，根据算法的不同特性，将（多）机器人 / 传感器的监视 / 侦察 / 探测问题可以分作以下几类：

- **离线 vs 在线：**离线算法在侦察监视资源部署之前进行计算，而在线算法在侦察监视资源运行过程中进行决策。所以，在线算法更适用于环境变化的场景，或者某些资源可能出现故障的场景。
- **有限 vs 无限规划步长：**对于有限步长的规划方法，计算时考虑最大化有限时间步长累积的回报。无限规划步长算法最大化无限时间步长下回报总和的期望值。
- **连续监视 vs 单次遍历：**连续监视的方式主要针对动态变化的环境。单次遍历的侦察监视方式适用于需要获取环境的一次性快照的情况。
- **策略 vs 非策略对手：**策略性监视尝试减少由完全理性（Perfected Rational）敌方入侵带来的损失。非策略性侦察监视问题中不存在这种策略性的对手。

进一步，考虑环境中的特性，可以将这些方法分为：

- **空间动态性 vs 时空动态性：**对于仅仅为空间动态性（Spatial Dynamics）的环境，智能体的观测值随着空间坐标进行变化，而时空动态性 (Temporal-Spatial Dynamics）是指观测函数随时间和空间坐标变化。前者适用于一些随时间静止的环境现象，比如地形高度或者建筑物布局；后者适用于随时间和空间共同变化的现象，比如天气条件和核辐射等。
- **静态 vs 非静态：**静态（Stationary）的环境是指环境状态的联合概率分布随时间固定不变，非静态（Non-Stationary）的环境是指环境状态的联合概率分布随时间动态变化。所以，一般所谓的动态（Dynamics），可能是静态的，也可能是非静态的，而非静态性提供了接近更为复杂环境的描述方式。

表1.1给出了信息收集问题相关不同算法对以上问题性质的考虑。下面将进一步对静态环境的信息模型进行简单综述。

智能体执行侦察监视可以看做是为了收集环境中的信息。这类问题中的一个挑战就是需要通过有限的数据来预测环境中其他坐标的信息。对于环境现象的建模，近年的工作常常采用高斯过程（Gaussian Processes, GPs）[39]，一种能够有效描述环境时空关系的模型。

对环境的不确定性，可以采用（与观测坐标相关的）探测质量准则进行定量表达，这些准则包括共同信息 [40]、熵 [41]、信息片模型 [42] 和区域覆盖模型 [43]。例如，采用共同信息（Mutual Information, MI）准则来进行不确定性的度量时，对于一个坐标集合，MI 准则定义如下：

$$\mathrm{MI}(X_y; X_A) \equiv H(X_y) - H(X_y|X_A), \tag{1.1}$$

其中，$H(X_y)$ 为考虑到过去观测集合后指定位置 X_y 的环境熵的标量，且 $H(X_y|X_A)$ 为探测完坐标集合 X_A 之后位置 X_y 的环境熵的标量。

具体地，GPs 为一类非参数概率模型，并且可以将其中多变量高斯分布扩展至涵盖整个位置和时间集合的一个无限个数随机变量的模型。GPs 的主要优势为其能够描述环境时间空间现象且能够提供关于确定性预测的一个值，即对环境不确定性的确定性预测。GP 可以完全由均值函数 $m(\mathbf{x})$ 和协变量函数（或称为核）$k(\mathbf{x},\mathbf{x}')$ 确定，即 GP 可以看做为不同函数的概率分布，其中每个随机变量代表了一个函数 f 在一个指定点的值，其形式上为：

$$f(\mathbf{x}) = \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}')).$$

为了简化，将随机变量集合 X_A 的均值向量表示为 μ_A。假设已测量的列向量为 x_A，那么 GP 能够利用这些信息来预测任何其他位置，从而提供对这些位置不确定性的确定性预测。基于这些观测，可以得到 X_y 的服从均值 $\mu_{y|A}$ 和协变量 $\sigma_{y|A}$ 的高斯分布如下：

$$\begin{aligned}\mu_{y|A} &= \mu_y + \sigma_{yA}\sigma_{AA}^{-1}(x_A - \mu_A),\\ \sigma_{y|A} &= \sigma_{yy} - \sigma_{yA}\sigma_{AA}^{-1}\sigma_{Ay}.\end{aligned}$$

一般地，协变量函数为一个随时间和空间的递增函数。例如：对于描述平滑现象，一种典型的建模方法为一个协变量随距离指数递减的二次指数函数：

$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\left|\mathbf{x}-\mathbf{x}'\right|^2/l^2\right),$$

其中 σ_f^2 和 l 为超参数，用于建模信号的变化以及现象的长度规模。后面这项决定了现象随时间和空间变化的快慢。从而，方差 σ^2 的常分布的熵可以定义为：

$$H(X_y|X_A) = \frac{1}{2}\ln(2\pi e\sigma^2_{X_y|X_A}).$$

在本节综述的无人机侦察监视问题的研究中，文献 [12] 提出了一个基于 MDP 模型的求解方法，用于计算完全可观环境中多智能体连续收集环境信息的行为策略。值得注意的是，本论文的工作结合了文献 [12] 中的考虑的信息收集问题并将其进行扩展，建立威胁、部分可观和非静态环境中的监视问题的理论框架并设计一系列求解方法。

1.3.2 不确定性规划

规划，是指生成一个用于未来多个时间步将要执行的计划。在人工智能和运筹学领域中，对于不确定性问题进行规划，一般可以将其形式化描述为 MDP 进行

求解。从 1950 年代 Bellman 和 Howard 的研究 [44, 45] 开始，在运筹学领域，MDP 模型逐渐受到广泛的关注和深入的研究。在随后的三十年中取得了一系列成果，包括逐步建立起的 MDP 的相关理论，以及一系列求解算法 [46]。人工智能领域从 1990 年代开始引入 MDP 的模型，用于和当时的一个热点 - 经典规划的相关研究形成联系 [47–50]。逐渐地，MDP 模型在人工智能领域的概率规划（Probabilistic Planning）和增强学习（Reinforcement Learning）[51] 两个方向的研究中得到广泛使用。其中，概率规划的研究，假设已知模型的完整知识，重点关注如何设计有效的方法对问题进行求解；增强学习研究更难的规划问题：假设对于环境的模型事先并不掌握完全的知识，需要在实践过程中进行学习。

概括来说，一个 MDP 问题包括以下几个部分：世界状态的集合、智能体动作集合、一个状态转移概率模型（当前状态下执行某个动作到达某个新状态的概率）和一个目标函数（最大化一个时间序列下智能体获得的累积回报值）。一个 MDP 的解可以看做一个映射：将状态空间中每个状态映射为一个动作。在 MDP 的每个决策周期内，智能体完全掌握系统当前所处的状态，并根据该状态选择一个行动（Action）作用到环境中。一方面，这个动作将导致系统从一个状态转移到另一个状态，且 MDP 通过转移函数（Transition Function）描述系统从某一个状态转移到另外某一个状态的概率（即实现了对环境和动作的不确定性进行建模）。另一方面，这个动作使智能体获得一定的收益（Reward），且 MDP 通过收益函数（Reward Function）表达在特定状态下，智能体执行某个行动所获得的回报。MDP 问题的复杂度为 P，即存在多项式时间的最优算法。然而，在一些实际大规模问题下，系统所有状态的个数可能非常的多，MDP 的快速求解仍然非常困难。

MDP 模型的一种自然扩展就是考虑问题中传感器的局限性（即当前状态未必可以完全掌握）并将这类问题描述为 POMDP。在 POMDP 中，通过观测（Observation）来描述传感器得到环境的局部反馈，并通过观测函数（Observation Function）来描述状态观测的部分可观性和不确定性。在 POMDP 的决策过程中，智能体不仅需要考虑当前的观测信息，同时也需要考虑过去的历史信息（包括过去的动作和观测），从而得到当前所处状态的一个概率分布。这个关于状态的概率分布也被称为信念状态。由于信念状态的概率分布是针对一个连续空间的，所以 POMDP 也可以看作连续状态空间的 MDP，其求解难度要比 MDP 大得多。对于进一步实际应用，POMDP 的求解主要面临两个方面的挑战：一方面，POMDP 模型本身不具有明确表达约束（Constraints）的能力，POMDP 中通常将约束看作一个绝对值很大的负的回报，然而这种方式不能保证严格满足约束条件；另一方面，POMDP 的求解往往依赖于精确的状态转移模型和观测模型，然而实际的任务环境比较复杂，其中的这些模型很难准确获得，或者随着时间的发展可能发生变化。

这些困难将会使得 POMDP 的求解难度与求解时间进一步增大。

论文关注的 MPOMDP 和 Dec-POMDP，是 POMDP 在多智能体协作问题上的自然扩展。MPOMDP 和 Dec-POMDP 的求解复杂度比 POMDP 要难得多。关于多智能体协调与合作的研究将在第1.3.3节进行综述。

此外，其他的相关研究包括旅行商问题（Traveling Salesman Problems, TSP）和车辆路径问题（Vehicle Routing Problems, VRP），下面将分别进行综述。

TSP 描述的问题如下：给定城市列表以及其中每两个城市间的行驶距离，找到一条最短的路径满足：从指定的城市出发，访问每个城市一次，然后回到出发的城市。TSP 是组合优化中典型的 NP 难问题。TSP 有很多变形，包括奖励收集 - 旅行商问题（Prize-Collecting TSP, PC-TSP）[52] 和概率旅行商问题（Probabilistic TSP，PTSP）[53]。在 PC-TSP 中，对各个城市指定一个非负的“奖励”值，旅行商需要对城市集合的一个子集进行访问，目标是最小化总的行程长度并最大化所获得的奖励。这类问题中对“奖励”的获取与论文关心的对环境中的“信息”进行收集类似。作为 TSP 的扩展，PC-TSP 仍然为 NP 难问题。k-TSP[54] 是 PC-TSP 的一种进一步变形，即每个城市拥有相等的奖励，商人至少访问 k 个城市，且目标为最小化总的路程长度。另外的一类问题，定向越野问题（The Orienteering Problem, OP）[55] 也可以看做 PC-TSP 的另一种变形，即最大化所获得奖励的同时需要保证旅行的路程小于一定的阈值。OP 在随机性领域的一种变形为基于随机利润的定向越野问题（Orienteering Problem with Stochastic Profits, OPSP）[56], 假设每个点的可能奖励服从一定的概率分布，且 OPSP 的目标为在有限的路径长度内，最大化如下结果发生的概率：能够保证收集到一定值的奖励总和。在 TSP 的基础上，PTSP 考虑每个城市的客户出现的概率为某个具体值，且不同城市的客户之间的出现概率相互独立。

VPR 首次由 Dantzig 和 Ramser 在文献 [57] 中进行了形式化描述：生成一个路径的集合，满足只访问每个客户一次，并且目标为最小化行程的时间和（或）运行代价。当考虑 VRP 中一些参数的不确定性时，VRP 问题扩展为随机车辆路径问题（Stochastic VRP, SVRP）[58, 59]。对 SVRP 不同的研究主要考虑过的不同的不确定性要素包括：客户是否出现、客户需求等级、城市间交通时间以及对每个客户的服务时间。

虽然这些 TSP 和 VRP 问题的相关研究均考虑了实际问题的一些不确定性和随机性，其中的一些求解方法获得了一定的求解性能，但是这些模型表达问题不确定性的能力仍然不足，特别是针对威胁环境中的动态变化以及部分可观等性质将无法进行处理。

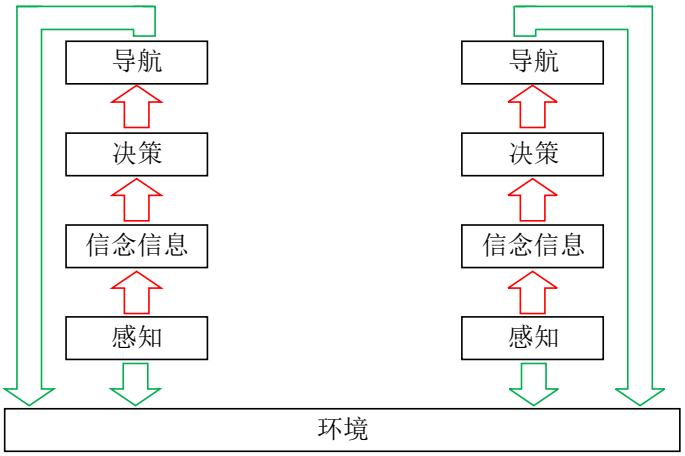
1.3.3 多智能体协调与合作

随着现代技术的发展，在人为参与较少时，交互的计算机系统如何构成团队合作运行，成为现代计算机科学的主要挑战之一。这一挑战主要涉及人工智能、机器人和机器学习三个学科的交叉研究。其中，这些系统中的单个单元，在动态环境中，具有感知、推理和行为的能力，通常被称作自主智能体 [60]。

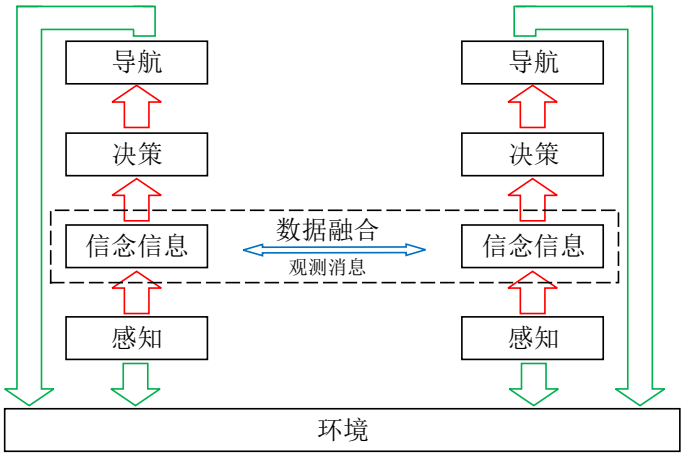
在智能体相关的理论与技术当中，智能体之间如何协调（Coordination）合作（Cooperation 或 Collabration）并进行决策，引起了广泛关注 [61, 62]。这种协调与合作是指，每个自主智能体的决策过程，需要考虑与其进行交互的其他智能体的策略，来达到更好的效果。从而，团队的集体策略通过所有单个智能体策略集合的联合形式表现出来。例如，当某个智能体知道其他智能体的策略时，智能体可以基于具有明确协调性的方式，进行动作的选择，进一步提高团队的性能。多智能体协调技术有着广泛的应用，包括无线传感器网络 [63]、灾难响应中的无人机 / 自主车系统 [38, 64]，以及分布式计算机系统中的多处理器任务等 [65]。进一步，多无人机系统也广泛应用于目标搜索问题 [62]、搜索及定位问题 [66]、搜索及跟踪问题 [67] 和仿真定位与建图问题 [68] 等。

此外，多机（包括无人机和有人飞机）协同任务规划成为机器人与控制领域近年来关注的热点问题之一 [10, 69–75]。这些研究采用分层递阶的思路，将任务规划问题分解为任务分配、航迹规划和编队控制，并分别进行求解。其中，任务分配为顶层问题，将待完成的任务集合分配给各个飞机或者编队；航迹规划和编队控制为底层问题，可以看作对于少量的几架飞机和其将要执行的少量任务目标，进行基于控制的飞行规划。这种方式最主要的问题是任务分配的方式过于严格，在任务动态变化时（如任务的添加与取消，飞机的加入或退出以及环境状态的变化等），任务分配的方式很难灵活地进行调整和优化，进而影响整个任务的执行效果。因此，论文将建立灵活的在线规划的方法，提供有效的理论支撑，解决动态变化的、大量任务目标的规划问题。

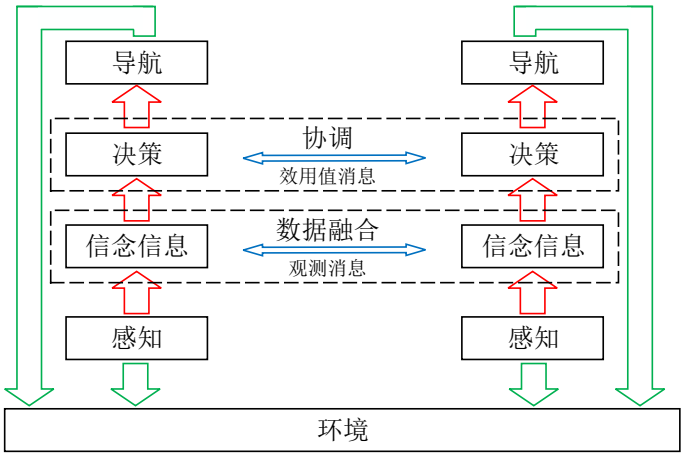
对于多无人机的决策制定过程，图1.4（a）、（b）和（c）给出了无人机之间进行协调的三种体系。这些体系协议中的模块包括感知、信念信息、决策和导航。每个无人机根据这一体系规划并执行任务。同时，无人机集群系统也可以根据这一体系来设计和执行团队的协调算法。基于此，根据无人机之间信息共享的程度，可以将无人机分散式协调的方法分为三类：1）对于无协调方法（如图1.4（a）），智能体之间行为独立，且相互之间不存在任何交互，从而每个智能体只需要根据收集到的所有局部信息来进行决策 [76]。实际中，这种无协调的方法将可能得到非常差的结果。2）对于隐含协调的方法（如图1.4（b）），智能体之间通过分享它们的观测来构建公共的信念模型，但是独立进行决策 [33]。这种隐含协调方式的弊端



(a) 不协调方法的体系



(b) 隐含协调方法的体系



(c) 明确协调方法的体系

图 1.4　三种层次协调方法的体系

为，由于掌握共同的知识，智能体可能会做出相同的多余的策略（如访问同一片区域）。3）从而需要明确协调的方式（如图1.4（c）），即智能体之间共享各自的观测和策略，智能体需要考虑其他团队的行为来制定自己的策略 [77]。这种明确协调的缺点主要是寻找最优联合策略的计算代价很大（随智能体个数呈指数增长）。

这些多智能体协调的方法主要包括集中式和分散式两种 [61]：

- 集中式（*Centralised*）：在完全集中式的系统中，可以由某个智能体来对整个团队进行协调，即通过收集整个团队的所有相关信息并对整个团队的行为进行规划。理论上，这个智能体能够计算并生成最优解。但是在实际中，完全集中式的方式很难应用于大规模团队中，这是因为某个点的故障可能造成整体任务失败，对通信条件要求很高，且通常对于局部变化反应迟钝。因此，集中式的方法主要应用于小的团队，且环境为静态的或者全局信息容易得到的问题中。
- 分散式（*Decentralised*）：在完全分散式的系统中，智能体主要依靠局部知识进行决策。这些方法的特点为速度快，能够灵活应对变化且能够鲁棒地应对。但是由于有时很难将好的局部解联合成为好的全局解，分散式的方法可能产生次优解。所以，完全分散式的方法往往更适用于大规模的团队执行一系列相对简单任务且不严格要求最优性的问题中。

论文第2章和第3章分别针对集中式和分散式的侦察监视问题展开研究。

1.3.4 人与机器人协作搜索

人类进行决策和制定计划方式正在发生着新的变革，任务规划系统逐渐需要人、软件智能体及硬件智能体的集合进行共同规划 [78]。具体地，随着计算机和网络技术的快速发展，越来越多、各种各样的软件（如各种数字设备的软件和应用等）和硬件（如配有传感器的手持设备等）充斥于人们的生活和工作当中；数据和信息以前所未有的速度和规模，通过更大范围的来源途径，不断地生成并提供给用户；在这些基础上，未来的计算机系统及任务规划系统还将不断产生一些新的方式和特征。从而，人 / 用户的行为 / 操作将越来越多的被任务规划系统依赖，并和软件智能体交叉于任务规划系统当中。这种现象是由两个方面原因造成的：一方面，受限于人工智能等相关技术的发展，当前软件智能体进行自主和独立规划的能力，仍然与科学家和工程师们最初对这些系统的定位和期望有着很大的差距；另一方面，软件智能体很难掌握和考虑到人的一些属性（包括经验和偏好等等），即这些属性很难在软件系统中进行有效的建模表达。随着这一趋势的发展，任务规划系统将明显涉及到人与智能体在系统中共生交错的情况。

这种人与智能体集合构成的系统，与一般的多智能体系统具有的相似点包括：

- 系统都是由自主“参与者”（包括人和智能体）构成；
- 参与者需要协同 / 协商；
- 参与者的行为（包括团队行为和个体行为）不可预测；
- 参与者具有角色和责任；
- 问题具有不确定性和动态性；
- 目标：优化 / 搜索 / 预测等等。

而且，与传统多智能体系统相比，在这种人与智能体集合当中，相比于其他一般智能体，人的特点如下：

- 人通常不需要与工具（如传感器）直接进行交互，而计算机需要；
- 人在操作中通常不接触和使用数学模型；
- 对于事物如何工作 / 运行 / 发展，人的大脑中能够建立一些相应的特殊模型，并且这些模型可以随时间改变；
- 关于人如何做决策 / 判断，很难对其进行数学建模。

近些年，一些人与智能体集合的雏形开始出现在人们的视野当中。比如，人们越来越多的使用交互式的交通管理软件来辅助制定出行的路线和方式。根据用户的偏好和一些实际的情况，更多的软件智能体开始主动来帮助人们安排城市交通中的拼车服务，以及旅行中的住宿和餐馆等。虽然一些相关领域如人工智能、人机交互（Human Computer Interaction）、计算机支持协同工作（Computer-Supported Cooperative Work）和普适计算（Ubiquitous Computing）涉及到了这种人与智能体集合的部分特征，但是如何建立和设计人与智能体协作的系统，以及怎样使得这种系统能够被接受和部署在真实世界中，仍是一项具有挑战的研究课题。

考虑到机器人能力的限制，大量的工作 [79–85] 研究如何实现机器人在运行中能够寻求人的辅助。与本论文的研究相似，一些人与机器人交互（Human-Robot Interactions, HRI）的模型 [79, 80] 允许运行中的机器人能够寻求人的辅助来降低环境中的不确定性。特别的是，在文献 [80] 的基于 POMDP 形式化描述中将人建模为观测信息提供者。POMDP 为一种非常通用的模型，能够表达真实世界中的多种因素，用来支持不确定决策制定问题，如机器人在不确定环境中进行搜索。然而，由于 POMDP 的很高的求解复杂度（PSPACE 难）[86]，现有的求解方法很难最优求解大规模问题。

使用机器人进行搜索，已经在机器人领域和运筹学研究领域得到了广泛的关注。这些研究大多关注如何规划机器人的路线并将这类问题建模为旅行商问题 [52]、定向越野问题 [87] 或者树 / 图搜索问题 [88]。然而，这些研究通常假设确定性参数的模型。与论文相似，[89] 研究了具有随机价格的旅行购物问题，其中真

实商品的价格未知但是已知其概率分布函数。购买者消耗不同的代价来行走于这些位置之间[3]。然而，文献 [89] 中指出这种变化代价的旅行商问题为 NP 难问题。为了避免考虑这种变化的代价，论文中假设访问不同位置的代价为固定的且相互之间独立，基于此期望找到一个多项式时间求解方法。这种假设满足于很多真实问题当中，如一个自主探测车需要探测的物品之间距离很近，从而这种物理移动耗费的能量或时间远远小于开采这些矿石所消耗的；又如无人机搜索某个坠落直升机的碎片时，更为重要的代价为环境中威胁，而且不同位置之间的这种代价或威胁可以看作是相互独立的。

受协商问题中的用户和智能体交互机制 [90] 的启发，在对机器人搜索问题建模时，论文考虑机器人允许主动寻求人的辅助而且考虑这种人与机器人交互是不确定的。在数学领域中，最优停止理论 [91] 考虑选择某个时间执行某个特殊动作的问题，来最大化期望收益或者最小化期望代价。最优停止问题的具体例子包括经典的秘书问题 [92]、搜索理论 [93]、交易选择 [94] 和协商问题 [95]。与论文紧密相关，文献 [90] 研究了协商问题中的最优渐进式偏好提取问题，其中的自动智能体能够代表用户来进行协商并逐渐提取用户对于不同结果的偏好。因此，论文第4章中的机器人同样采取主动与人进行交互的方式。

以上对论文相关研究的现状进行了综述。基于此，下面将给出论文的主要研究内容和创新点。

1.4 论文研究工作

1.4.1 主要研究内容

论文的研究，首先针对两类多智能体规划问题（即 MPOMDPs 和 TD-POMDPs），设计在线求解方法。然后，研究无人机集群侦察监视问题，基于 MPOMDs 和 TD-POMDPs 建立问题的形式化模型，设计针对这类问题的求解算法，并设计仿真实验对论文提出的算法进行验证和评价。具体的研究内容如下：

(1) **大规模 MPOMDPs 的在线近似求解算法。** 与 Dec-POMDPs 只允许局部通信或者无通信相比，在全通信的 MPOMDP 中，所有智能体均可以掌握各自获得的观测并共同协调相互的策略。对于不存在稀疏交互性质的多智能体问题（包括 MPOMDPs），其联合策略空间将无法进行解耦，因此大规模的实际问题仍然无法求解。然而，在很多实际问题的规划或优化中，问题的目标函数均具有子模性（Submodular）。例如在侦察监视问题当中，这种子模性可以直观地理解为：相比已经获得了很多信息的情况，对于当前只是拥有很少信

[3] 文献中商品的价格可以看作论文中物品的负的回报值。

息情况，添加同一份新的信息，将具有更有益的帮助。对于这种具有子模性规划目标的问题，数学界拥有一系列多项式算法可以获得具有性能保证的近似最优解。基于此，论文首次研究这种具有子模性回报函数的 MPOMDPs，尝试找到一种在线规划方法，在理论上保证近似最优性能的同时，问题的策略空间随智能体的个数呈多项式规模增长。

(2) **大规模 TD-POMDPs 的分散式在线求解算法**。由于具有稀疏交互的性质，TD-POMDPs 呈现出弱耦合的性质，使得这类 Dec-POMDPs 表现出一定程度的可扩展性。具体地，造成 Dec-POMDPs 不可解的根本原因是其联合策略空间随问题的规模快速增长，即其 NEXPSAPCE 的问题复杂度。TD-POMDP 对状态空间、动作空间和观测空间进行一定程度的解耦，并通过解耦后的局部状态之间和局部观测之间的变换依赖（Transition-Dependent）关系，以及回报值依赖（Reward-Dependent）关系来保持智能体之间的弱耦合性质。这种模型的结构实现了将 Dec-POMDP 模型分解成为弱耦合的大量（每个智能体一个的）局部 POMDP 模型构成的集合。然后，这些局部 POMDP 模型之间通过其变换依赖的关系实现相互影响（Influence）。

论文重点研究 TD-POMDPs 中的如下问题：1）智能体局部策略之间进行相互影响的逻辑关系，以及局部决策与全局规划目标的关系；2）如何实现单个智能体根据局部观测以及与少量部分智能体的策略进行局部 POMDP 的在线规划求解；3）如何实现大量智能体进行分散式交互来协调各自的局部策略，从而有效地最大化全局的规划目标（Social Welfare）。4）马尔科夫决策理论的模型本身不具有明确表达约束（Constraints）的能力，通常将约束看作一个负的很大绝对值的回报，然而这种方式不能保证严格满足约束条件，论文将研究如何在 TD-POMDPs 的求解算法中如何严格满足这种问题中的约束。

(3) **基于 MPOMDPs 和 TD-POMDPs 的无人机集群侦察监视问题**。研究无人机集群侦察监视这类问题，并基于马尔科夫决策理论对问题进行形式化描述。定义无人机的行为、相互间的交互与协调、环境的动态性和部分可观的性质、规划目标，以及问题的不确定性等。具体地，首先定义环境模型，对环境中分布的信息和威胁进行数学表达，特别是需要考虑环境随时间和空间进行动态变化的性质。然后定义侦察监视智能体模型，即定义智能体在环境中进行移动、获得观测和进行侦察监视的动作，以及对于获得信息多少的评价函数。无人机执行任务的一些限制可以通过一系列的约束进行表达（包括无人机的生命力，如续航能力等）。进一步，针对不同的无人机集群侦察监视问题，基于 MPOMDPs 和 TD-POMDPs 分别进行形式化描述与求解，并设计

仿真实验环境对论文的方法进行验证与评价。

(4) **概率知识和人辅助下的无人机搜索方法**。研究一类不确定知识下的搜索问题，其中一架自主无人机搜索环境中的一项物体（如一架坠落战机的碎片）而且这个无人机在搜索过程中允许主动寻求人的辅助。在这个问题中，不确定性体现在物体的回报值以及人的可利用性（Availability）。无人机的目标是最大化所获物体的回报值并最小化总的搜索代价。据作者所知，当前研究没有提出过类似的模型，即描述机器人能够主动寻求人的辅助的搜索问题，并且现有的多项式算法不能直接用于对于这一问题进行最优求解。因此，论文设计一个新颖的模型来表达这类搜索问题，并提出一种多项式时间算法实现对问题的最优求解。

1.4.2 创新点分析

基于以上研究内容，论文的创新点如下：

(1) **针对具有子模性规划目标的 MPOMDPs 问题，首次提出了一种近似最优的多智能体在线规划算法**。这种算法通过顺次分配技术（Sequential Allocation Technique）来依次计算每个智能体的策略，贪婪地最大化单个智能体对团队任务目标的边际贡献（Marginal Contribution），从而避免了直接考虑团队的联合策略（其导致的计算代价与智能体个数呈指数关系），使得计算复杂度随智能体个数呈多项式关系。论文通过理论证明该算法具有很好的近似最优性能。创新性工作为：使用顺次分配技术来计算智能体的策略，相比于其他的搜索团队联合策略空间的方法，这种方法具有很好的可扩展性，并能够满足问题的实时性要求。

(2) **针对 TD-POMDPs 问题，首次提出了具有良好可扩展性的在线规划算法 TD-POMCP，即一种基于蒙特卡洛树搜索（Monte Carlo Tree Search, MCTS）和 max-sum 的分散式在线算法**。TD-POMCP 的创新主要包括：1）根据局部智能体团队的局部联合动作和联合观测，利用 TD-POMDP 模型中变换依赖的弱耦合关系，对每个智能体分别构建一个前向搜索树；2）基于 MCTS 这种基于采样的搜索算法对每个智能体的搜索树进行扩展和更新；3）所有智能体通过分散式交互的方式同时进行对各自搜索树的扩展和更新。在搜索树中的每个规划步骤，通过 max-sum 这种分散式任意时间近似算法进行智能体的动作选择。论文通过理论证明这种方法具有很好可扩展性、鲁棒性、实时性和最优性，并且通过相关的仿真实验证明论文的方法可以成功应用于大规模 TD-POMDPs。

(3) **基于 MPOMDP 和 TD-POMDP 模型，分别建立了在威胁环境和不确定性下的无人机集群系统进行侦察监视的集中式和分散式问题模型。** 这些问题的状态空间大小随环境中监视子目标个数的增加呈指数增长，使得通用的规划求解器无法求解这种大范围的侦察监视问题，更无法满足实际任务的实时性要求。因此，论文设计了随监视子目标个数线性增长的信念状态表示方式，并进一步定义了无人机集群系统侦察监视问题的形式化模型。基于此，设计了仿真实验，实验结果表明论文的算法能够成功应用于大规模无人机团队在大范围环境中进行集中式 / 分散式侦察监视的规划问题。与现有的侦察监视问题的求解方法相比，论文设计的问题模型与求解算法的创新点包括：1）基于智能体模型描述了无人机与环境和与其他无人机之间的交互方式；2）环境的模型能够表达其在时间和空间上的部分可观和非静态性质；3）求解算法在进行蒙特卡洛仿真时，通过不断保持对健康约束的检测，在规划中能够处理无人机可能遭受的来自威胁的伤害。

(4) **针对概率知识和人辅助下进行搜索的决策问题，建立了一个新颖的模型，机器人和人的搜索问题（Robot-Human Search, RHS），并设计相应的基于潘多拉规则的多项式时间的最优求解算法。** 受限于当前的相关技术水平，机器人 / 无人机脱离于人独立完成任务的能力仍然十分有限，往往需要人的辅助。然而，通用的规划和搜索算法很难应用于每个动作都可能产生大量（可能为无穷多）的不同值的搜索问题中。基于这一背景，论文对概率知识和人辅助下进行搜索的决策问题进行了研究，创新点主要包括：1）RHS 不仅考虑了环境中搜索目标回报值的不确定性，而且考虑了机器人能够主动地寻求人的辅助；2）基于经济学搜索问题的潘多拉规则，设计了针对这类无人机搜索问题的多项式时间最优搜索算法。

1.4.3 论文组织结构

本节介绍论文研究内容的组织结构。论文总体上分三个部分：第一部分为绪论，为第一章，主要介绍研究现状和基础，并提出论文的研究内容；第二部分为自动规划方法，包括第二章和第三章，分别研究集中式和分散式无人机集群侦察监视的规划方法；第三部分为人机交互部分，为第四章，主要研究概率知识和人辅助下的无人机搜索问题。

论文的具体组织结构如图1.5所示，绪论之后各章的主要内容组织如下：

第二章研究集中式侦察监视规划。首先，建立问题的模型，并对单个智能体进行侦察监视的问题进行研究并设计求解算法。然后，将该算法扩展至求解多智

能体集中式侦察监视的规划问题中。最后，对论文设计的算法进行理论分析和仿真实验验证。

第三章研究分散式侦察监视规划。首先，研究大规模 TD-POMDPs 并设计分散式在线求解算法。然后，基于 TD-POMDP 对分散式侦察监视问题进行建模并设计其分散式在线求解算法。最后，对论文设计的算法进行理论分析和仿真实验验证。

第四章研究人辅助下的搜索方法。首先，建立概率知识和人辅助下的无人机搜索问题模型，以及决策问题的动态规划形式化表达。然后，设计多项式时间求解算法，并基于潘多拉问题的潘多拉规则证明该算法的最优性。最后，设计仿真实验验证算法的性能。

第五章总结与展望。对全文的工作进行总结，给出主要结论与研究成果，并展望进一步的研究工作。

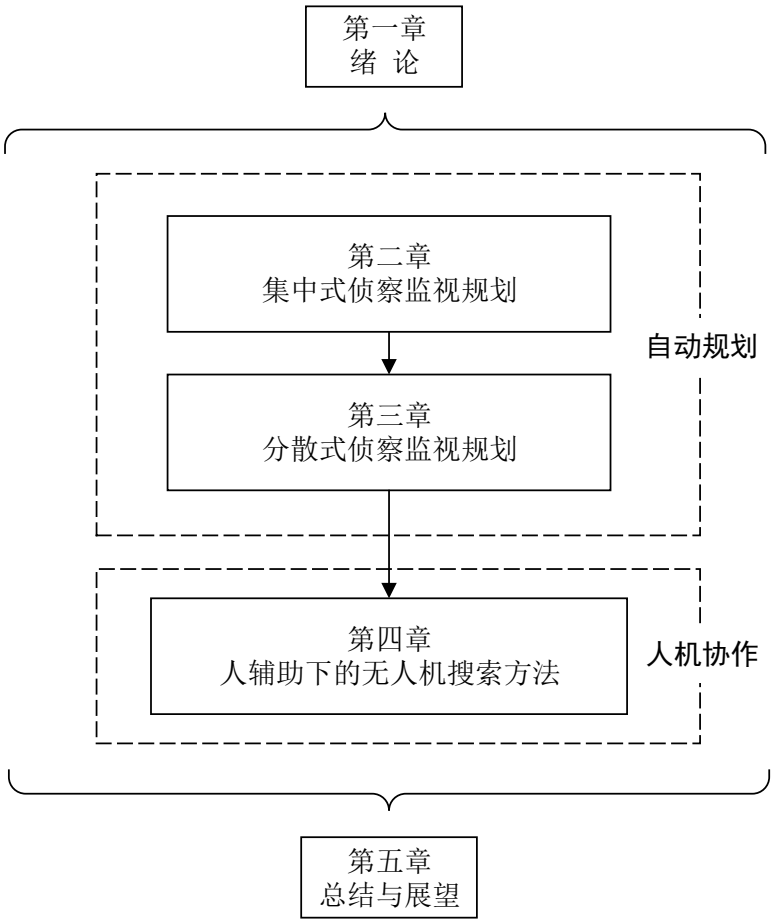


图 1.5　论文组织结构

第二章　基于顺次分配技术的集中式侦察监视规划

2.1　引言

无人机集群系统侦察监视问题具有一系列特殊的性质。首先，这些问题往往天生具有很强的动态性（如待侦察敌方目标可能进行移动，或者大火可能蔓延和扩散等）和不确定性（如敌方目标的位置未能完全确定，或者着火的地点和火势并非完全可以观测到）。同时，环境中分布的威胁可能对智能体造成伤害（如无人机有时需要靠近大火中的建筑物，或者环境中的碎片可能降落并碰到无人机）。这些威胁也具有动态变化的性质，而且常常分布于环境中具有重要信息价值的区域。智能体需要不断衡量并尽量减少这些威胁可能带来的伤害。从而，在这种快速变化的环境中，智能体需要根据所掌握的信息，快速做出反应，规划进一步的策略。

本章研究的集中式侦察监视规划，通过集中式协调和规划各个智能体的行为，使用有限数量的智能体资源，在任务时间内，获取最大程度的态势认知。由于智能体的观测范围不可能对整个环境实现完全覆盖，因此需要对环境的动态性和不确定性进行充分考虑，对这些不直接可观的区域的状态进行预测和估计。从而，智能体的决策（即如何对这些大量环境子目标进行连续侦察），需要考虑为在任务时间内的多步决策（即智能体不仅要考虑当前侦察动作的效果，还需要考虑该动作对未来决策的影响）。

近些年，学术界涌现出了大量的用于求解无人机团队侦察监视问题的方法。然而，在对环境进行建模时，大多数的研究考虑其为完全可观的和静态的（详见第1.3.1节）。而且，这些方法均无法解决在部分可观和非静态环境中，无人机怎样避开威胁并执行任务。

对于不确定环境中的规划问题，基于智能体建模领域的研究在近几年取得了很大的进展 [96]。在基于智能体的模型中，一个智能体可以看做一个封装的计算机系统，而且其行为具有一定的灵活性和自主性，并针对所设计的子目标部署于一些环境当中 [60]。其中，这些智能体可以为某个软件或者硬件（如机器人或无人机）。特别是，这些模型中的智能体能够应对一些不确定的情形，主要包括：动作的执行结果可能与理想的结果有偏差，智能体执行任务时环境动态变化，以及智能体的观测并不完全准确的情形。

基于以上背景，本章结合智能体模型，研究在不确定和威胁环境中的侦察监视问题，并设计一种新颖的算法来求解这一规划问题。具体地，本章的创新点可以总结如下：（1）首次提出了不确定和威胁下的多智能体侦察监视问题的求解方法。本章的模型不仅考虑了动态环境的部分可观和非静态的特性，而且考虑了侦

察监视智能体的健康状态；（2）设计了一种预测性启发式，来评估智能体当前时刻所有可能计划的值，并基于此设计了一种单智能体的在线规划算法；进一步，采用顺次分配技术，提出了一种依次计算单个智能体计划的多智能体算法。特别地，论文证明了相对于当前的其他一些基准算法，这种多智能体算法能够扩展到更大规模的问题中（即 15 个智能体）；（3）通过仿真实验证明本章算法的表现显著优于基本测试算法，其中规模为 10 个智能体的问题超过 44%，15 个智能体的问题超过 21%。

本章余下各节安排如下：首先介绍这类问题的数学描述。然后考虑单个智能体进行侦察监视的问题建模与算法设计。基于此，论文设计一种依次求解单个智能体策略的方法来计算多智能体系统的策略。最后，设计仿真实验对论文的方法进行验证。

2.2 问题描述

本节首先给出一个例子来解释在任务想定中智能体是如何执行任务的。然后具体给出问题的定义。

例子：考虑智能体进入某个失火的建筑物中，其中大火的等级（威胁状态变量）和关于遇害者和物资的信息值（信息变量）随时间变化。当对该建筑物进行探索时，智能体能够获取一定的信息，同时可能受到来自于大火的威胁。在每个时间步，基于环境中每个位置的信息及威胁状态的观测，某个智能体选择某个子目标进行访问。然后智能体获得基于信息值的回报值，并且承受相应威胁状态造成的损失。

下面对本章的侦察监视问题进行定义，具体介绍其构成：侦察监视环境、侦察监视智能体及其任务目标。

2.2.1 侦察监视环境

通过定义物理上的空间、时间和动态属性对侦察监视环境进行描述。例如，战场环境中的一些重点地点的具体状况需要快速得到关注，其中通往这些区域的途径可能有限（例如环境中存在树，碎片或者其他自然障碍物），可以将这一特性描述为智能体在这些地点之间来往的一些路径。进一步，将环境中的空间特性描述为图，来表达智能体如何在环境中移动。

定义 (图): 将侦察监视环境中的物理区域建模为一个无向图 $G = (V, E)$，其中 V 表示欧式空间中顶点的坐标集合，E 表示边的集合且其中每条边表示智能体能够在该边所连接的两个顶点之间进行往返运动。定义顶点个数为 N, 即, $N = |V|$。

在一些作战想定中，每个受关注地点可以表示为图的一个顶点，图的一条边表示该边所连接的两个地点之间无人机可以通过某路径直接到达。

定义 (时间): 通过一系列的时间步 $\{1,2,\ldots,T\}$ 来表示时间，智能体可以在环境中每一个时刻 $t \in \{1,2,\ldots,T\}$ 访问环境中的某个位置。

为了定义环境中的动态属性，考虑每个顶点上拥有两个状态变量：信息和威胁。

定义 (信息状态变量): 一个信息状态变量 e_v^I 表示顶点 $v \in V$ 的不同信息等级。

例如，在灾难环境中，某个位置有多少人需要进行救援或帮助，以及某座大桥的损毁状态，均可以表示为信息状态变量。

定义 (威胁状态变量): 一个威胁状态变量 e_v^R 表示顶点 $v \in V$ 的不同威胁等级。

例如，在灾难环境中，某个位置的大火等级或者烟雾浓度，均可以定义为典型的威胁状态变量。

定义 (马尔科夫信息 / 威胁模型): 每个顶点的信息和威胁状态变量随时间按照离散时间多态马尔科夫链进行变换。

为了表示环境状态变量转移的性质，本文采用马尔科夫链的模型。具体地，对于一个 K 状态 $S=(S_1,S_2,\ldots,S_K)$ 的马尔科夫链，每对状态之间的转移概率矩阵为：

$$P=\begin{bmatrix} p_{11} & \cdots & p_{1K} \\ p_{21} & \cdots & p_{2K} \\ \vdots & \ddots & \vdots \\ p_{K1} & \cdots & p_{KK} \end{bmatrix},$$

其中，p_{ij} 为状态 S_i 通过一个时间步长转移为 S_j 的概率，$S_i,S_j \in S$。接下来介绍马尔科夫威胁和信息模型。图2.1 给出了一个某顶点的马尔科夫信息和威胁模型的例子。图2.1（a）为一个 2- 状态（即 R_1 和 R_2）的威胁模型，图2.1（b）为一个 3- 状态（即 I_1, I_2 和 I_3）的信息模型，其中给出了每两个信息 / 威胁状态之间的转移概率（例如如一个时间步长内，R_1 变换为 R_2 的概率是 0.1）。

顶点 v_n 的信息状态集合 $I^n=\{I_1^n, I_2^n, \ldots, I_{K_I^n}^n\}$ 对应智能体访问该位置时可能获取的 K_I^n 个不同信息值。信息值大小由函数 $f^n: I^n \to \mathbb{R}^+$ 确定。每个位置的信息状态按照 K_I^n- 状态的马尔科夫链模型的转移概率矩阵 P_I^n 独立进行变化。

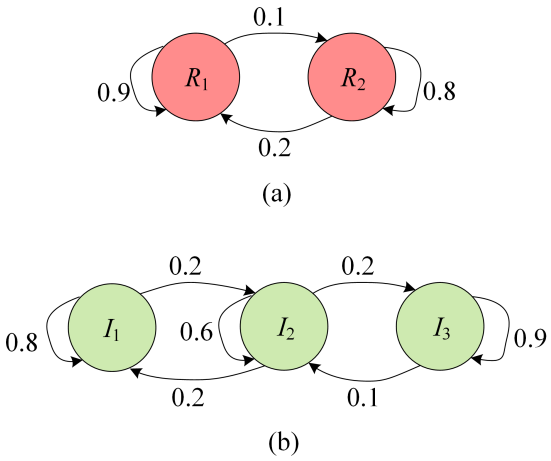


图 2.1　某顶点信息和威胁模型的例子

类似地，顶点 v_n 的威胁状态集合 $R^n = \{R_1^n, R_2^n, \ldots, R_{K_R^n}^n\}$ 对应该顶点的可能的 K_R^n 个不同威胁等级。智能体访问顶点 v_n 时所遭受的伤害大小由函数 $c^n : R^n \to \mathbb{R}^+$ 确定。每个位置的信息状态按照 K_R^n- 状态的马尔科夫链模型的转移概率矩阵 P_R^n 独立进行变化。

以上定义了智能体执行任务的环境，接下来将对智能体模型及其规划目标进行设计。

2.2.2　侦察监视智能体

侦察监视智能体分布于上节中定义的环境当中。

首先，定义一个侦察监视智能体为一个移动物理实体，部署于上文定义的环境当中，通过访问环境中的顶点，能够收集环境中的信息，同时可能遭受环境中威胁的伤害。然后定义智能体的移动和访问的能力。在图 G 中进行侦察监视时，在每个时刻 t，每个智能体位于图 G 中的某个顶点，且多个智能体可以同时位于同一个顶点。智能体的移动为基元的，在图 G 的框架下，发生在任意两个连续时间步之间，即智能体位于某个顶点 $v_i \in V$，移动至 G 中的某个相邻顶点 $v_i' \in adj_G(v_i)$。假设 $\forall v_i \in V, v_i \in adj_G(v_i)$，即一个智能体可以待在某个顶点不动。智能体的速度在一个单位之间内足以到达相邻的顶点。因此，如果无人机能够在 5 分钟内实现在两个位置之间移动，那么可以将模型中的时间步长设置为 5 分钟。

从而，如果智能体当前位于 v_n，将对该节点进行访问。通过这个访问，一方面，智能体掌握顶点 v_n 的当前的信息状态和威胁状态，不妨记为 I_i^n 和 R_j^n；另一方面，该智能体获得回报 $f^n(I_i^n)$ 并遭受损失 $c^n(R_j^n)$。对于智能体访问的具体操作时间，假设可以忽略不计。便于下文的介绍，由 $F^n = [f^n(I_1^n), \ldots, f^n(I_{K_I^n}^n)]$ 表示顶点 v_n 的信息值向量，其中 $f^n(I_k^n)$ 为信息状态是 I_k^n 时智能体可以获得的信息

值（例如，某个顶点的信息包括 3 个状态，对应 3 个信息值 $[0,2,5]$）。类似地，由 $C^n=[c^n(R_1^n),\ldots,c^n(R_{K_R^n}^n)]$ 表示顶点 v_n 的伤害值向量，其中 $c^n(R_k^n)$ 为威胁状态是 R_k^n 时智能体将会遭受的伤害值（例如某个位置的火势等级包括 4 个状态，则对应 4 个等级的伤害值 $[0,4,6,10]$，又如某个位置的烟雾浓度的 3 个状态，则对应 3 个等级的伤害值 $[0,2,5]$）。对于每次访问，如位置 v_n，智能体获得该位置的信息值后将该位置的状态重置为 I_1^n（I_1^n 意味着 v_n 自从上次被访问后，还没有新的信息生成）。由于每个顶点的状态随时间变化，且智能体只能够掌握当前正在访问位置的状态，从而侦察环境可以看做为非静态的（即状态的联合概率分布随时间发生变化）和部分可观的。

以上定义了侦察监视问题，现在需要基于智能体行为和观测的历史和环境模型，对智能体接下来的序列侦察监视动作进行规划。因此，接下来，首先提出单个智能体在图中进行侦察监视的 POMDP 形式化描述，并设计一个算法对其进行求解。基于此，将提出多智能体侦察监视的求解方法。

2.3 单智能体侦察监视问题求解

本节首先基于 POMDP 的框架建立单智能体侦察监视问题的数学模型。然后将提出针对本文的 POMDP 的一种信念空间的简洁表示方法。基于此，将定义预测性启发式和在线的单智能体求解算法。

2.3.1 基于 POMDP 的形式化建模

将单智能体侦察监视问题描述为如下的 POMDP $\langle\mathcal{S},\mathcal{A},\mathcal{O},T,\Omega,r\rangle$：

- $\mathcal{S}$ 为状态的集合。一个状态定义为 $s=[v,(s_R^1,\ldots,s_R^N),(s_I^1,\ldots,s_I^N)]\in\mathcal{S}$，其中 v 为智能体当前位置，$s_R^n\in R^n$ 和 $s_I^n\in I^n$ 分别为顶点 $v_n\in V$ 处的威胁和信息状态。便于描述，通过 s_e=$[(s_R^1,\ldots,s_R^N),(s_I^1,\ldots,s_I^N)]\in\mathcal{S}_e$ 表示出所有位置的威胁和信息状态。可以看出，状态空间 $\mathcal{S}$ 的大小随顶点个数呈指数增长。

- $\mathcal{A}$ 为所有动作的集合。智能体选择一个相邻顶点进行访问作为一个动作。

- $\mathcal{O}$ 为所有观测的集合。定义一个观测 $o=(v_i,s_I^i,s_R^i)\in\mathcal{O}$ 为当前位置 v_i 及其信息状态和威胁状态。

- $\mathcal{T}$ 为状态转移条件概率的集合。本文中假设位置 v 的转移为确定性的，且由智能体运动的目的位置来确定。基于上文中定义的马尔科夫模型，s_e 按照其中的 $\prod_{n=1}^N K_R^n K_I^n$ 个状态以离散时间马尔科夫过程的方式进行转换。

- Ω 为观测概率的集合。由于每个观测 o 直接为一些状态的局部部分，观测概率可以表示为：当观测 o 与某个状态 s 中的相应部分一致时 $\Omega(o|s',a)=1$，否则 $\Omega(o|s',a)=0$。

- $r:\mathcal{A}\times\mathcal{O}\to\mathbb{R}$ 为回报函数。$r(a,o)$ 为智能体执行动作 a 得到观测 o 时获得的回报：

$$r(a,o)=\alpha f^i(s_I^i)-(1-\alpha)c^i(s_R^i), \tag{2.1}$$

其中 α 为权重系数。

由于模型中的状态无法直接获得，按照一般 POMDP 问题中的方式，定义标准的信念向量为 $B(h)=[b_1(h),\ldots,b_M(h)]$，即状态空间 $\mathcal{S}$ 中所有可能状态的后验概率的分布，其中 $b_m(h)$ 为历史 h 下的第 m 个环境状态的概率。由文献 [97] 中的证明可知，对于任何时刻 t，这种信念状态表示都是一种用于设计最优动作的充分统计（Sufficient Statistic）。从而，一个策略 π 指定在任何给定的信念状态下将要执行的动作。最优策略 π^* 为使得智能体在任务时间 T 内累积获得的回报值最大的策略。然而，由于每一个环境状态为所有顶点信息和威胁状态的集合，因此这个 POMDP 中状态空间 $\mathcal{S}$ 包含的状态个数为 $\prod_{n=1}^{N}K_R^nK_I^n$。可以看出，这个状态个数随顶点个数呈指数增长。从而，由于信念向量由所有可能状态的后验概率组成，所以信念空间的规模同样随顶点个数呈指数增长。

为了解决这一问题，下面首先定义的一种降维的信念向量表示方法，进一步提出一种基于预测性启发式的在线求解算法，在保证求解质量的同时降低搜索空间的规模。

2.3.2 简洁的信念表示

可以看出，各个顶点的威胁状态和信息状态变量均独立变化，而且 v 是确定性的。因此，与文献 [23, 24] 中的方法类似，能够找到一个维数随 N 线性增长的满足充分统计的表示，用于设计最优策略。接下来，本节将介绍这种简洁的信念表示方式以及这种表示方式的转移函数。

鉴于每个顶点上的信息和威胁变量均独立进行变化，从而可能存在一种维数随顶点个数 N 线性增长的信念表示。具体地，对于历史 h，定义条件概率 $\Psi(h)=[\Psi_R(h),\Psi_I(h)]$ 为特征信念表示，其中 $\Psi_R(h)$ 定义为：

$$\begin{cases}\Psi_R(t) &= (w_R^1(t),\ldots,w_R^N(t))\\ w_R^n(t) &= (w_{R1}^n(t),\ldots,w_{RK_R^n}^n(t)),\end{cases} \tag{2.2}$$

其中 $w^n_{Rk_1}(h)$ 为如下条件的概率：顶点 v_n 的威胁状态为 $R^n_{k_1}$。并且 $\Psi_I(h)$ 定义为：

$$\begin{cases} \Psi_I(t) &= (w^1_I(t),\ldots,w^N_I(t)) \\ w^n_I(t) &= (w^n_{I1}(t),\ldots,w^n_{IK^n_I}(t)). \end{cases} \tag{2.3}$$

其中 $w^n_{Ik_2}(h)$ 为如下条件的概率：顶点 v_n 的信息状态为 $I^n_{k_2}$。从而，任何 $\Psi(h)$ 包含 $\Sigma^N_{n=1}(K^n_R+K^n_I)$ 个元素，即元素个数随 N 线性增长。

定理 2.1: 对任意 h_t，$\Psi(h_t)$ 为 $B(h_t)$ 的一种等价表示。

证明：证明当不同顶点的信息和威胁状态独立变化时，$\Psi(h_t)$ 可以从 $B(h_t)$ 得到，其中 $B(h_t)$ 为 s_e 中所有元素的概率分布。不失一般性，考虑 $N=2$。定义 τ_n 表示最近的顶点 v_n 被访问的时刻，可以通过历史 h_t 得到。从而可以将 $b_m(t)$ 中的项列出如公式（2.4）中所示，通过推导得到的项即为 $\Psi(h_t)$ 中的各项。因此，$\Psi(h_t)$ 为 $B(h_t)$ 的一种简洁的等价表示。

$$\begin{aligned} & \Pr\left[s^1_R(t)=i', s^2_R(t)=i'', s^1_I(t)=j', s^2_I(t)=j'' \mid \mathcal{I}(t)\right] \\ = \ & \Pr\left[s^1_R(t)=i', s^2_R(t)=i'', s^1_I(t)=j', s^2_I(t)=j'' \mid s^1_R(\tau_1)=o'_R, s^2_R(\tau_2)=o''_R, \tau_1, \tau_2\right] \\ = \ & \Pr\left[s^1_R(t)=i' \mid s^1_R(\tau_1)=o'_R\right] \Pr\left[s^2_R(t)=i'' \mid s^2_R(\tau_2)=o''_R\right] \\ & \Pr\left[s^1_I(t)=j' \mid \tau_1\right] \Pr\left[s^2_I(t)=j'' \mid \tau_2\right]. \end{aligned} \tag{2.4}$$

假设可以获得每个顶点的初始概率信息。然后，对于任何动作 a 和观测 o，特征信念向量 $\Psi(h)$ 的由下式更新为 $\Psi(hao)$：

$$\begin{aligned} w^n_R(hao) &= \begin{cases} \widetilde{I}_k & \text{if } v_n \in v(hao), s^n_R(h)=R^n_k \\ w^n_R(h)P^n_R & \text{if } v_n \notin v(hao) \end{cases} \\ w^n_I(hao) &= \begin{cases} \widetilde{I}_k & \text{if } v_n \in v(hao), s^n_I(h)=I^n_k \\ w^n_I(h)P^n_I & \text{if } v_n \notin v(hao) \end{cases}. \end{aligned} \tag{2.5}$$

其中 $\forall v_n \in V, R^n_k \in R^n$，$I^n_k \in I^n$，而且 $\widetilde{I}_k$ 为第 k 个元素为 1 的单位向量。如公式（2.5）所示，对于某个智能体正在访问的顶点 v_n 的威胁信念向量 $w^n_R(hao)$，基于该顶点的观测 $R^n_k(h)$ 进行更新；对于当前没有智能体访问的顶点，通过该顶点的威胁信念向量 $w^n_R(h)$ 和威胁马尔科夫模型 P^n_R 进行更新。对于 $\forall v_n \in V$，信息信念 $w^n_I(hao)$ 的更新与 $w^n_R(hao)$ 类似。

基于上面的转移函数，策略 π 表示为行动序列 $\pi=[\pi(1),\pi(2),\ldots]$，其中 $\pi(t)$ 为在时刻 t 所选择的进行访问的顶点。从而，最优策略可以表达为：

$$\pi^* = \arg\max_{\pi} \mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty} \gamma^t \mathcal{R}^{\pi(t)}\left(\Psi(t)|\Psi(0)\right)\right], \tag{2.6}$$

其中 $\mathcal{R}^{\pi(t)}(\Psi(t))$ 为当信念状态为 $\Psi(t)$ 时获得的期望回报值，$\gamma \in [0,1]$ 为折扣因子。

虽然信念状态的维数降低了，但是这个问题仍然是一个大规模 POMDP 问题，很难求得最优解。下一节中，将基于这个低维数的信念向量表达方式，建立预测性启发式和在线单智能体算法。

2.3.3 预测性启发式

在定义用于在线策略选择的预测性启发式之前，首先引入关于马尔科夫状态转移矩阵为单调矩阵的假设，这个假设直观上可以理解为：如果当前某个顶点拥有更高的信息 / 威胁值，那么下一时刻该顶点也更可能拥有更高的信息 / 威胁值。然后再基于这种转移矩阵的单调性，表示出未来的期望回报作为预测性启发式。

随机占优（Stochastic Dominance）的概念，在经济学、金融学和统计学得到了广泛应用 [98]。其在相关的问题研究中都有过类似的假设，比如对通信系统的频道状态 [23, 24] 和无人机侦察目标的状态 [15]。具体地，两个 Z 维随机变量 x, y 之间的随机占优 $\succ$ 定义为 $x \succ y$，如果满足下面的条件：

$$\sum_{j=i}^{Z} x(j) \geq \sum_{j=i}^{Z} y(j), \quad \text{for } i = 2, 3, \ldots, Z. \tag{2.7}$$

假设马尔科夫信息模型和马尔科夫威胁模型为单调矩阵，即转移概率矩阵 P_R^n 和 P_I^n 满足：

$$\begin{aligned} P_{RK_1}^n &\succ P_{RK_1-1}^n \succ \ldots \succ P_{R_1}^n, \\ P_{IK_2}^n &\succ P_{IK_2-1}^n \succ \ldots \succ P_{I_1}^n. \end{aligned} \tag{2.8}$$

当转移函数矩阵 P_R^n 和 P_I^n 满足上面的假设时，P_R^n 和 P_I^n 为单调矩阵 [99]。从而，如果某个顶点当前的状态信息值高于另外某个顶点，该顶点下一个的状态信息值高于另外那个顶点下一个的状态信息值的可能性更大，即若 $w_I^n(t) \succ w_I^{n'}(t)$，则 $w_I^n(t)P_I^n \succ w_I^{n'}(t)P_I^{n'}$。由公式（2.8）可知，如果某两个顶点当前均没有智能体访问，它们的信息状态的概率向量在下一时刻将保持随机占优的关系。明显可以看出，即若 $w_I^n(t) \succ w_I^{n'}(t)$, 则 $w_I^n(t)F^n \geq w_I^{n'}(t)F^{n'}$。这意味着随机占优的信息信念向量更有可能拥有更高的信息值。同理可知，随机占优的威胁向量更有可能拥有更高的伤害值。特别地，由于当某个智能体访问某个节点时将该节点的信息状态复位于 I_1，信息状态的信念向量为 $(1, 0, \ldots, 0)$ 且被任何没有正在访问的顶点的信息信念向量随机占优，从而最近访问的顶点通常会有相对较低的期望信息值。

值得注意的是，这种单调性的假设并非要求信息值（或伤害值）随时间增长，而是一种使得信息（或威胁）的概率向量满足随机占优的性质。下面通过举例对

此进行说明，一个 4 状态的马尔科夫威胁模型如下：

$$P_R = \begin{bmatrix} 0.8 & 0.1 & 0.1 & 0 \\ 0.4 & 0.5 & 0.0 & 0.1 \\ 0.2 & 0.1 & 0.6 & 0.1 \\ 0 & 0.0 & 0.4 & 0.6 \end{bmatrix}.$$

由此可以看出，P_R 中的向量满足方程（2.8）中的条件，即 $P_{R4} \succ P_{R3} \succ P_{R2} \succ P_{R1}$，其中以 $P_{R3} \succ P_{R2}$ 为例，P_{R2} 和 P_{R3} 中的元素满足方程（2.7）中的随机占优条件：

$$\begin{aligned} &0.1 + 0.6 + 0.1 \geqq 0.5 + 0.0 + 0.1, \\ &0.6 + 0.1 \geqq 0.0 + 0.1, \\ &0.1 \geqq 0.1. \end{aligned}$$

例如，如果顶点 v_1 和 v_2 的威胁信念向量分别为 $w_R^1 = [0.1, 0.2, 0.5, 0.2]$ 和 $w_R^2 = [0.2, 0.4, 0.3, 0.1]$，即 $w_R^1 \succ w_R^2$，那么 v_1 的下一个威胁状态比 v_2 的下一个威胁状态更可能更严重。然而，一个时间步长过后的同一个顶点的具体威胁状态，既有可能变得更严重，也有可能变得缓和了。

基于这种矩阵单调性假设，可以通过不同顶点在上一时刻的信念状态来“预测”当前没有正在访问的顶点在当前时刻的信念状态。基于此，智能体可以对将要在将来几步访问的顶点可以获得的期望回报值进行估计。将智能体 D 时间步长的可执行的策略表示为 $\pi_D(t) = (\pi_{t+1}, \ldots, \pi_{t+D})$，其中包含了 D 个连续的可执行 / 访问的动作 / 顶点。

下面定义策略 $\pi_D(t)$ 的预测性启发式，即执行 $\pi_D(t)$ 的预测性期望回报值 $\mathbb{E}[\hat{\mathcal{R}}(\pi_D(t))]$，即 $\pi_D(t)$ 每一步动作的期望回报值的和：

$$\mathbb{E}[\hat{\mathcal{R}}(\pi_D(t))] = \sum_{i=1}^{D} \gamma^t \left(\alpha \hat{w}_I^{\pi_{t+i}}(t+i) F^{\pi_{t+i}} - (1-\alpha) \hat{w}_R^{\pi_{t+i}}(t+i) L^{\pi_{t+i}} \right), \tag{2.9}$$

其中，$[\hat{w}_I^{\pi_{t+i}}(t+i), \hat{w}_R^{\pi_{t+i}}(t+i)]$ 为顶点 π_{t+i} 在时刻 $t+i$ 的预测性信念向量。可以由当前信念向量 $\Psi(t)$、当前位置 $a(t)$ 和观测 $\theta(t)$ 计算得到 $t+1$ 时刻的信念向量 $[\hat{w}_I^{\pi_{t+1}}(t+1), \hat{w}_R^{\pi_{t+1}}(t+1)]$，即 $\Psi(t+1) = \delta\left(\Psi(t) | a_t^*, \theta(t)\right)$（公式（2.5））。对于 $\{t+2, \ldots, t+D\}$，可以得到基于公式（2.5）并忽略观测信息后得到的信念向量的转移函数如下：

$$\begin{aligned} \hat{w}_R^n(\tau+1) &= \hat{w}_R^n(\tau) P_R^n \ , \\ \hat{w}_I^n(\tau+1) &= \begin{cases} P_{I1}^n & \text{if } v_n = \pi_\tau \\ \hat{w}_I^n(\tau) P_I^n & \text{if } v_n \neq \pi_\tau \end{cases} , \end{aligned} \tag{2.10}$$

算法 2.1 单智能体侦察监视算法

已知: $\{P_R^n\}$: 马尔科夫威胁模型
已知: $\{P_I^n\}$: 马尔科夫信息模型
已知: $\Psi(t)$: 当前时刻的信念状态
已知: $o(t)$: 当前位置获得的观测
已知: $v(t)$: 当前位置
求: $a^*(t+1)$: 智能体的下一个动作

▷ *Step* 0*:* 获得所有可执行的策略 $\Pi_D(t)$*;*
▷ *Step* 1*:* 计算最佳策略：
for $\pi_D(t) \in \Pi_D(t)$ **do**
▷ *Step* 1.1*:* 获得接下来 D 向量：
  $\Psi(t+1) \leftarrow \delta\left(\Psi(t)|v(t), o(t)\right)$
  for $\tau \in \{t+1, \ldots, t+D-1\}$ **do**
    for $v_n \in V$ **do**
      $\hat{w}^n(\tau+1) \leftarrow \hat{\delta}\left(\hat{w}^n(\tau)|\pi_\tau(\tau)\right)$
    end for
  end for
▷ *Step* 1.2*:* 计算 $\pi_D(t)$ 的预测性期望回报：
  $\mathbb{E}[\hat{\mathcal{R}}(\pi_D(t))] = \alpha w_I^{\pi_{t+i}}(t+i)F^{\pi_{t+i}} + \beta w_R^{\pi_{t+i}}(t+i)L^{\pi_{t+i}}$
▷ *Step* 1.3*:* 比较 $\pi_D(t)$ 和存储的最佳策略：
  if $\mathbb{E}[\hat{\mathcal{R}}(\pi_D(t))] > \mathbb{E}[\hat{\mathcal{R}}(\pi_D^*(t))]$ **then**
    $\pi_D^*(t) \leftarrow \pi_D(t)$
  end if
end for
▷ *Step* 2*:* 根据最佳策略 π_i^* 返回下一步的动作：
return $a^*(t+1) \leftarrow \pi_{t+1}^*$

其中 $\tau = \{t+1, \ldots, t+D-1\}$。

基于预测性启发式和向前看 D 步的策略，智能体将比较所有长度为 D 的可执行路径，然后选择能够获得最高预测性期望回报的路径，选取其中的第一步作为下一个将要访问的位置。下一节中将详细介绍在线单智能体算法中如何使用这种启发式。

2.3.4 单智能体侦察监视规划算法

基于预测性启发式，提出单智能体侦察监视问题的在线求解算法（见算法2.1）。

首先，计算由当前位置 $v(t)$ 出发的所有可执行策略的集合 $\Pi_D(t)$（*step* 0），其中参数 D 称为最大视野，即 POMDP 中的策略的向前看的步数。然后，计算所有策略的预测性期望回报。对于每个策略，$t+1$ 时刻的信念状态由 t 时刻的信念状

态、位置及观测信息通过方程（2.5）进行更新。$\{t+2,\ldots,t+D\}$ 时刻的预测性信念状态由方程（2.10）计算得到（line 3-7）。基于此，计算策略的预测性回报（line 8）。从而最佳策略为：

$$\pi_D^*(t) = \arg\max_{\pi_D(t)} \mathbb{E}[\hat{\mathcal{R}}(\pi_D(t))]. \tag{2.11}$$

最后选择下一步动作为最佳策略的第一个动作 $a^*(t+1) = \pi_{t+1}^*$（line 13）。

以上定义了在不确定性和威胁下的单智能体侦察监视的方法，下面将扩展研究集中式的无人机集群侦察监视问题及策略。

2.4 多智能体侦察监视问题求解

在集中式的无人机集群系统侦察监视问题中，无人机可以通过全通信的方式共享相互之间获得的观测信息以及各自的策略。在实际应用中，为了满足集中式的指控方式，环境中部署有地面控制中心，协调多架无人机实现跟踪观测连续变化的环境状态，其中每个无人机都能够与控制中心进行交互通信。然而，对于一些复杂环境中，由于传输距离的限制或者带宽的限制，无人机很难与地面控制中心进行通信，无人机与无人机之间也只能实现在有限范围内的通信，这种环境条件只能使用分散式的方法来规划侦察策略。论文将在第3章对分散式监视问题进行研究。另外，本章对于多智能体监视的问题，给出如下假设：

假设：当两个或两个以上智能体同时访问某个位置时，智能体团队仅获得一份信息值，但是对该位置进行访问的每个智能体都要遭受一份伤害。

这个假设适用于这样一些想定中：单个智能体传感器获得信息的能力等价于多个智能体的能力。因此，在侦察过程中，智能体之间需要基于各自的局部观测对各自的策略进行相互协调。基于此，论文将集中式的多智能体监视问题形式化描述为 MPOMDP，并设计具有可扩展性的在线多智能体算法，协调智能体的策略。

2.4.1 基于 MPOMDP 的形式化建模

MPOMDP 问题可以简化看作一个 POMDP 问题：由中心控制器执行联合动作并获得联合观测 [100]。下面将这种多智能体问题形式化描述为一个 POMDP 问题：

- $\mathcal{S}$ 为状态集合。一个状态定义为 $\vec{s} = [\vec{v}, (s_R^1,\ldots,s_R^N), (s_I^1,\ldots,s_I^N)] \in \mathcal{S}$, 其中 $\vec{v}$ 为智能体的当前位置集合，$s_R^n \in R^n$ 和 $s_I^n \in I^n$ 为顶点 $v_n \in V$ 的威胁和信息状态。记 $\vec{s_e} = [(s_R^1,\ldots,s_R^N), (s_I^1,\ldots,s_I^N)] \in \mathcal{S}_e$ 表示环境中所有位置的信息和威胁状态集合。

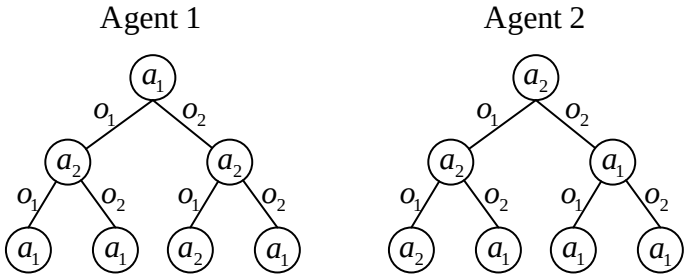


图 2.2　两个智能体的策略树举例

- $\mathcal{A}$ 为联合动作集合。智能体选择各自相邻顶点进行访问作为一个联合动作。
- $\mathcal{O}$ 为联合观测集合。对于所有智能体当前位置和这些位置的信息和威胁状态，定义为一个联合观测 $\vec{o}=\{\vec{v},\{o^i|\forall v_i\in\vec{v}\}\}\in\mathcal{O}$，其中，$o^i=(s_R^i,s_I^i)$ 为智能体 i 获得的观测。
- $\mathcal{T}$ 为状态转移概率的集合。假设位置状态的转移为确定性的且由智能体的联合动作的目的地集合确定。$\vec{s_e}$ 按照一个包括 $\prod_{n=1}^{N}K_R^nK_I^n$ 个状态的离散时间马尔科夫链进行转换。
- Ω 为观测概率集合。由于问题中一个观测 $\vec{o}$ 确切为某些状态的局部，因此若 $\vec{o}$ 与某个状态一致，则观测概率 $\Omega(\vec{o}|\vec{s}',\vec{a})=1$；反之，$\Omega(\vec{o}|\vec{s}',\vec{a})=0$。
- $r:\mathcal{A}\times\mathcal{O}\to\mathbb{R}$ 为回报函数。$r(\vec{a},\vec{o})$ 为执行联合动作 $\vec{a}$ 并获得观测 $\vec{o}$ 时的回报值：

$$r(\vec{a},\vec{o})=\sum_{v_i\in\vec{v}}\left(\alpha\frac{1}{n_{v_i}}f^i(s_I^i)-(1-\alpha)c^i(s_R^i)\right),$$

其中 n_{v_i} 为同时访问顶点 v_i 的智能体的个数。

基于此，智能体的目标是: 通过依次选择移动动作，来实现最大化 T 时间步的累积期望回报。如图2.2举例给出两个智能体的策略树。

2.4.2　基于顺次分配技术的多智能体侦察监视规划算法

方程（2.2）和（2.3）中的信念向量表示仍然能够用于多智能体 POMDP 的信念向量。然而，这个 POMDP 的联合动作空间和联合观测空间为所有单智能体动作空间和观测空间的笛卡尔积。因此，联合动作空间和联合观测空间的维数随着智能体个数呈指数增长，只有少量智能体的问题可以求解。然而，本文将采用顺次分配技术，即依次计算单个智能体策略的方式求解多个智能体的策略（见算法2.2），从而实现避免求解团队的联合策略。虽然这种方法看起来简单，但是能够满足有界最优的条件（第2.4.3节将对此进行理论分析）。

类似的算法已经成功应用于求解多智能体问题 [12, 14]。由于这些算法不同于本文的 POMDP 问题框架，不能直接应用其中的算法来求解本文的问题。因此，论文将要重点考虑部分可观的问题中单智能体如何依次使用单智能体算法求解其策略。

当依次运行单个智能体算法进行决策时，明显存在智能体选择策略的顺序：第 1 个智能体完成其最佳的 D 步策略，然后第 2 个 ... 等等。第 i 个智能体的未来 D 步策略 $\pi_D^i(t)$ 的期望值由当前的位置 $v_i(t)$、信念向量 $\Psi(t)$ 和之前智能体计算出的策略集合 $\mathcal{M}_{-i}=\{1,\ldots,i-1\}$ 来进行计算得到。

从而多智能体侦察监视中，第 i 个智能体的最佳策略递归求解如下：

$$\begin{aligned}
\widehat{\pi}_1^* &= \arg\max_{\widehat{\pi}_1} \mathcal{R}'\left(v_1(t),\Psi(t)\right)\\
\widehat{\pi}_2^* &= \arg\max_{\widehat{\pi}_2} \mathcal{R}'\left(v_2(t),\Psi(t),\widehat{\pi}_1^*\right)\\
&\vdots\\
\widehat{\pi}_i^* &= \arg\max_{\widehat{\pi}_i} \mathcal{R}'\left(v_i(t),\Psi(t),\widehat{\pi}_1^*,\ldots,\widehat{\pi}_{i-1}^*\right).
\end{aligned} \tag{2.12}$$

其中 $\widehat{\pi}_i^*$ 表示智能体 i 的最佳策略。

为了保证只考虑临界回报值时回报函数的计算准确性，需要消除对某一些位置的两种类型的重复访问。一种是同步重复访问，即当两个智能体同时访问一个顶点时信息值的回报将得到两次。另一种是异步访问，第 i 个智能体在时刻 t_1 访问顶点 v_n，但是之前已经有第 j $(j<i)$ 个智能体在其 D 步策略中的 t_2 $(t_1<t_2)$ 时刻访问了顶点 v_n（即 $t_1\geq t_2$）。此外，对于智能体 j 先于智能体 i 访问顶点 v_n 的情形，在通过方程（2.9）计算 $\mathbb{E}[\hat{\mathcal{R}}(\pi_D(t))]$ 时已经考虑了。

这里介绍异步重复访问，即智能体 i 决定在时刻 t_1 访问顶点 v_n，然而智能体 j $(j<i)$ 已经在其策略中的时刻 t_2 $(t_1<t_2)$ 访问过了顶点 v_n。不失一般性，考虑智能体 i 的策略 $\pi_D^i(t)$ 中顶点 v_n 只被智能体 j 访问过。如果 $\mathcal{M}_{-i}=\{1,\ldots,i-1\}$ 中有多于一次访问顶点 v_n，我们假设时刻 t_2 最为靠近 t_1（由转移方程（2.10）可知，只需要考虑最近时刻的那个访问）的那个时刻。基于以上假设，可以看到智能体 j 访问顶点 v_n 所收集的期望信息被过高估计了，因为没有意识到智能体 i 将在时刻 t_1 对该位置的信息进行复位。因此，我们引入罚因子 $\hat{p}\in\mathbb{R}^+$，用于智能体 i 补偿智能体 j 多计算的回报值，如下：

$$\mathcal{R}_i'\left(v_i(t),\Psi(t),\widehat{\pi}_1^*,\ldots,\widehat{\pi}_{i-1}^*\right)=\mathbb{E}[\hat{\mathcal{R}}(\pi_D^i(t))]-\hat{p}, \tag{2.13}$$

算法 2.2 集中式多智能体侦察监视算法

已知: $\{P_R^n\}$: 马尔科夫威胁模型
已知: $\{P_I^n\}$: 马尔科夫信息模型
已知: $\Psi(t)$: 当前时刻的信念状态
已知: $\vec{o}(t)$: 智能体集合获得的当前观测集合
已知: $\vec{v}(t)$: 智能体当前位置集合
求: $\vec{a}^*(t+1)$: 智能体的下一个动作集合
▷ 依次计算每个智能体的策略:
1: **for** $i = 1 \to |\mathcal{M}|$ **do**
▷ 计算智能体 i 每个可能策略的期望回报:
2: **for** $\pi_D^i(t) \in \Pi_D^i(t)$ **do**
▷ 根据重复访问计算罚因子:
3: $\hat{p} \leftarrow \hat{r}_{\text{expected}} - \hat{r}_{\text{revised}}$
▷ 计算期望回报:
4: $\mathcal{R}_i'\left(v_i(t), \Psi(t), \widehat{\pi}_1^*, \ldots, \widehat{\pi}_{i-1}^*\right) \leftarrow \mathbb{E}[\hat{\mathcal{R}}(\pi_D^i(t))] - \hat{p},$
5: **end for**
6: $\hat{w}^n(\tau+1) \leftarrow \hat{\delta}\left(\hat{w}^n(\tau)|\pi_\tau(\tau)\right)$
▷ 计算智能体 i 的最佳策略:
7: $\widehat{\pi}_i^* = \arg\max_{\widehat{\pi}_i} \mathcal{R}'\left(v_i(t), \Psi(t), \widehat{\pi}_1^*, \ldots, \widehat{\pi}_{i-1}^*\right).$
8: **end for**
▷ 根据最佳策略集合返回下一步将要执行的动作集合:
9: **return** $\vec{a}^*(t+1) \leftarrow \{\widehat{\pi}_i^*\}$

其中 $\mathbb{E}[\hat{\mathcal{R}}(\pi_D(t))]$ 为方程（2.9）中定义的期望回报函数，$\hat{p}$ 为智能体 j 在智能体 i 之后访问顶点 v_n 的损失，定义如下：

$$\hat{p} = \hat{r}_{\text{expected}} - \hat{r}_{\text{revised}}, \tag{2.14}$$

其中 $\hat{r}_{\text{expected}} \in \mathbb{R}^+$ 为智能体 j 计算访问顶点 v_n 的期望回报，$\hat{r}_{\text{revised}}$ 为考虑到智能体 i 的动作后修订的智能体 j 访问顶点 v_n 的期望回报。定义在时刻 $[t_1+1, \ldots, t_2]$ 顶点 v_n 的修订的期望信念状态为 $\{\tilde{w}^n(t_1+1), \ldots, \tilde{w}^n(t_2)\}$，通过方程（2.10），以及预测性信念状态 $\hat{w}^n(t_1)$ 和动作 $a(t_1) = v_n$ 来确定，从而，修订的期望回报如下：

$$\hat{r}_{\text{revised}} = \gamma^{t_2}\left(\alpha\tilde{w}_I^n(t_2)F^n - (1-\alpha)\tilde{w}_R^n(t_2)L^n\right). \tag{2.15}$$

通过算法2.1计算单个智能体长度 D 的策略，从而获得每个智能体下一步的动作，进一步构成团队的联合策略。既然每个智能体 i 的策略是基于 $\mathcal{M}_{-i}$ 的策略通过贪婪方式计算依次得到的，那么得到的这种团队策略并不能保证最优。然而，依然能够保证相对于最优联合策略的一个性能下界。

2.4.3 近似最优性证明

论文的单智能体算法为一种近视的算法，然而其全局最优性仍是一个开放性问题。本节对论文算法最优的性能分析，主要关注这种多智能体贪婪选择策略的近似最优的下界：

定理 2.2: 设 η 为单智能体侦察监视规划算法的近似最优性能，则论文这种算法通过依次贪婪分配规划的多智能体算法能够达到的近似性能的保证为 $1+\eta$。

证明：证明基于文献 [14] 中的方法进行。设 $E_i = \pi_1 \cup \ldots \cup \pi_i$ 为近似算法运行至第 i 次迭代得到的智能体路径集合[1] 所构成的观测集合 $(E = \emptyset)$，并设 E_i^* 为最优算法至第 i 段为止所获得的观测的集合 $(E_0^* = \emptyset)$。对于一个观测集合 π，定义剩余信息，$f_{E_{i-1}}(\pi) = f(E_{i-1} \cup \pi) - f(E_{i-1})$。容易证明得到，如果 f 为标准化的、单调的和子模的函数，那么剩余信息 $f_{E_{i-1}}$ 同样为标准化的、单调的和子模的。

基于此，对于第 i 阶段，近似算法能够保证找到一个计划，满足：

$$f_{E_{i-1}}(\pi_i) \geq \frac{1}{\eta}(f_{E_{i-1}}(\pi_i^*)),$$

在 $|N|$ 个阶段后，总的收集的信息满足:

$$\sum_{i \in N} f_{E_{i-1}}(\pi_i) \geq \frac{1}{\eta}\left(\sum_{i \in N} f_{E_{i-1}}(\pi_i^*)\right), \tag{2.16}$$

由于左边的求和可以合并，从而得到：

$$\sum_{i \in N} f_{E_{i-1}}(\pi_i) = f(\cup_{i \in N}\pi_i) = f(E_{|N|}) \tag{2.17}$$

右边得到：

$$R.H.S. = \frac{1}{\eta}\left(\sum_{i \in N}(f(E_{i-1} \cup \pi_i^*) - f(E_{i-1}))\right),$$

两边同时加 E_{i-1}^* 并基于子模性可以得到：

$$\begin{aligned} R.H.S. &\geq \frac{1}{\eta}\left(\sum_{i \in N}\left(f(E_{i-1} \cup E_i^*) - f(E_{i-1} \cup E_{i-1}^*)\right)\right) \\ &= \frac{1}{\eta}[f(E_1^*) - 0 + f(E_1 \cup E_2^*) - f(E_1 \cup E_1^*) + \ldots + \\ &\quad f(E_{|N|-1} \cup E_{|N|}^*) - f(E_{|N|-1} \cup E_{|N|-1}^*)], \end{aligned}$$

[1]为了简化，此证明中 π_i 表示第2.4.2中第 i 个智能体的路径 $\hat{\pi}_i^*$ 对应的观测。

重新排列各项：

$$R.H.S. \geq \frac{1}{\eta}\left[f(E_{|N|-1} \cup E^*_{|N|}) - \sum_{i=1}^{|N|-1}\left(f(E_i \cup E^*_i) - f(E_{i-1} \cup E^*_i)\right)\right].$$

基于 $(f(E_{|N|-1} \cup E^*_{|N|}) \geq f(E^*_{|N|}))$ 的单调性和 $(f(E_i \cup E^*_i) - f(E_{i-1} \cup E^*_i) \leq f(E_i) - f(E_{i-1}))$ 的子模性，可以得到：

$$\begin{aligned} R.H.S &\geq \frac{1}{\eta}[f(E^*_{|N|}) - \sum_{i=1}^{|N|-1}\left(f(E_i) - f(E_{i-1})\right)] \\ &= \frac{1}{\eta}[f(E^*_{|N|}) - f(E_{|N|-1})], \end{aligned}$$

基于 $(f(E_{|N|}) \geq f(E_{|N|-1}))$ 的单调性，得到：

$$R.H.S \geq \frac{1}{\eta}[f(E^*_{|N|}) - f(E_{|N|})]. \tag{2.18}$$

将方程（2.17）和（2.18）带入方程（2.16），得到：

$$f(E_{|N|}) \geq \frac{1}{\eta}[f(E^*_{|N|}) - f(E_{|N|})],$$

且因此得到：

$$f(E_{|N|}) \geq \frac{1}{(1+\eta)} f(E^*_{|N|}),$$

即得到的近似保证为 $(1+\eta)$。

以上，对侦察监视问题进行了形式化描述并且设计了单智能体算法和多智能体算法，接下来将在下一节中对这些方法进行仿真验证。

2.5 仿真实验验证

2.5.1 实验设置

为了通过实验的方式来验证论文提出的方法，本章将这些方法应用在 10 和 15 个智能体在大的图（图中包含 350 个顶点和 529 条边）中进行连续侦察监视的任务。因为智能体需要在一定时间要求内决定访问哪些位置，所以设置在线计算时间限制为 0.5s。由于单智能体算法可以看作是多智能体算法的一个特例，本节只针对多智能体算法的实验结果进行分析。论文对如下两种想定进行仿真：

- **想定 A:** 图中每个顶点都设置为相同的马尔科夫信息和威胁模型；

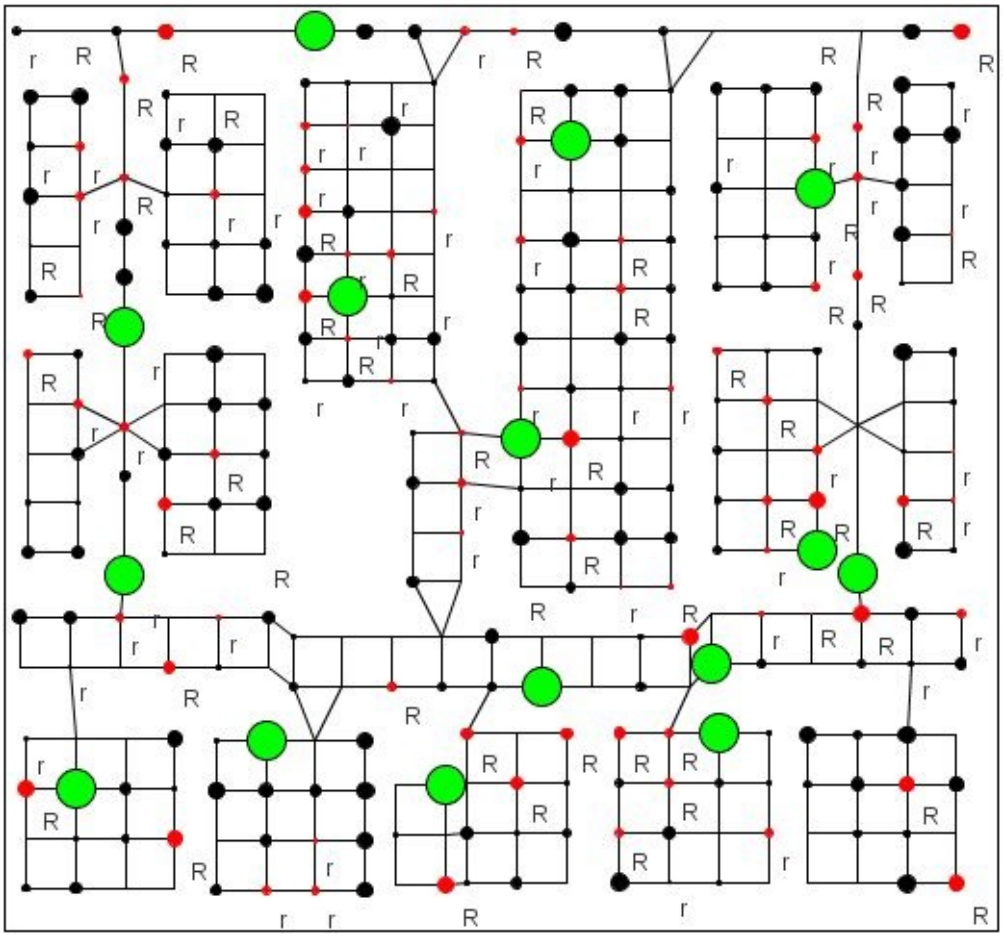


图 2.3　15 个智能体在大范围环境中进行侦察监视

- **想定 B:** 图中的各个顶点设置为 3 种不同信息和威胁模型中的一种。

值得注意的是，虽然想定 A 中的信息和威胁模型是相同的，但是由于这些信息和威胁都是非静态的，不同位置的威胁和信息将呈现出不同的状态。论文使用想定 A 来研究在环境中不同位置具有相同类型的威胁和信息状态的侦察监视问题；使用想定 B 来研究在不同位置采用异构的威胁和信息的侦察监视问题。

设置回报值（即方程（2.1））和值函数（即方程（2.6））的参数如下：权重系数 $\alpha = 0.33$ 和折扣因子 $\gamma = 0.9$。更具体地，在想定 A 中定义两类马尔科夫链如下：

$$P_I = \begin{bmatrix} 0.8 & 0.1 & 0.1 & 0 & 0 \\ 0.2 & 0.7 & 0.0 & 0.1 & 0 \\ 0.1 & 0.1 & 0.7 & 0.1 & 0 \\ 0 & 0.0 & 0.1 & 0.8 & 0.1 \\ 0 & 0 & 0.0 & 0.1 & 0.9 \end{bmatrix},$$

$$P_R = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0.4 & 0.4 & 0.2 \\ 0.0 & 0.2 & 0.8 \end{bmatrix}.$$

这里的传递函数 P_I 和 P_R 满足方程（2.7）的单调性假设。例如，P_R 的前两行满足方程（2.8）的约束，为 $0.9 \geq 0.4, 0.9+0.1 \geq 0.4+0.4$ 和 $0.9+0.1+0.0 \geq 0.4+0.4+0.2$。信息和威胁值向量分别设置为 $F = [0 \quad 1 \quad 2 \quad 3 \quad 4]$ 和 $L = [0 \quad 1 \quad 2]$。对于想定 B，将在这些顶点中设置一些不同的马尔科夫模型。15 个智能体的一个仿真实例见图2.3。其中每个顶点的圆圈的大小表示该位置回报值的绝对值，颜色表示符号（黑表示正，红表示负），智能体当前位置表示为绿色圆圈，每个顶点右下角的“R”和“r”分别表示威胁等级为“2”和“1”。

2.5.2 基准测试算法

对于标准的 POMDP 求解器如 POMCP，问题的联合动作空间和联合观测空间随智能体的个数呈指数增长，从而这些求解器无法求解具有大量可能动作和观测的多智能体侦察监视问题。因此，使用一个随机算法（Random）和一个基准算法（Baseline），与论文的算法进行比较，并对智能体分别使用不同方法所累积获得的信息值和遭受的伤害进行评价。具体地，

- **Random** 使智能体随机移动到某个与当前位置相邻的位置。
- **Baseline** 使智能体移动到与当前位置相邻的获得期望值最大的位置。并假设这种基准算法依次计算单个智能体的策略来避免不同的智能体同时访问某个顶点，所以与算法 PH-1 相似。
- **PH-D** 为论文提出的多智能体侦察监视算法，其中 D 为最大规划步长，即进行策略搜索时向前看的步长。从集合 $\{2, 4, 8\}$ 中选取 D 的取值来研究更多的规划步长能够带来多少收益。论文给出不同规划步长对应算法所生成的求解结果。

2.5.3 实验结果与讨论

各个智能体的初始位置随机分布在环境中，并在这个连续变化的环境中连续运行 3000 时间步。对于每个想定和每个算法，运行 1000 次并将结果在图2.4和图2.5 中表示出来，其中的误差条表示结果的均值在置信区间 95% 的可能取值范围。非覆盖的误差条指示出 $\alpha = 0.05$ 的零假设（Null Hypothesis）。在两个想定中，随机算法表现的很差，所获期望值小于其他两个算法的 30%。在想定 A 中，PH-8

和贪婪算法均表现很好，PH-8 超过贪婪算法 5%。然而，在想定 B 中，对于 10 个智能体和 15 个智能体，本文的算法分别优于贪婪算法 43% 和 21%。此外，对于不同的规划步长，由 PH-8 获得的回报值随 D 进行增长，同时计算时间随 D 呈指数增长。对于 $D > 8$，每一步的计算时间超过了在线决策时间限制。因此，论文中的预测启发式算法的计算效率会受到严重影响，从而需要在计算中权衡求解质量和计算时间。然而，这种算法的表现始终优于基准算法。

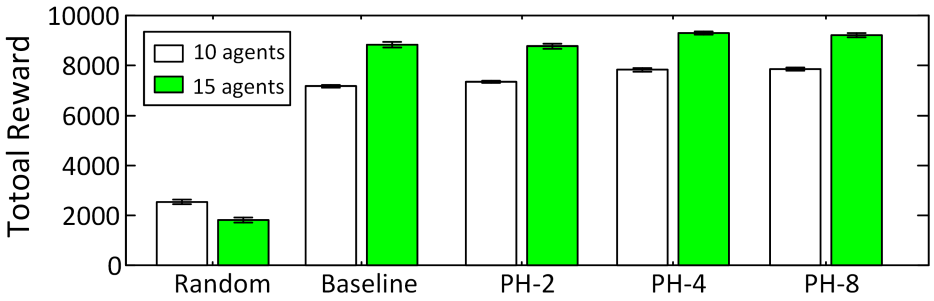


图 2.4　想定 A 中各算法获得的回报值

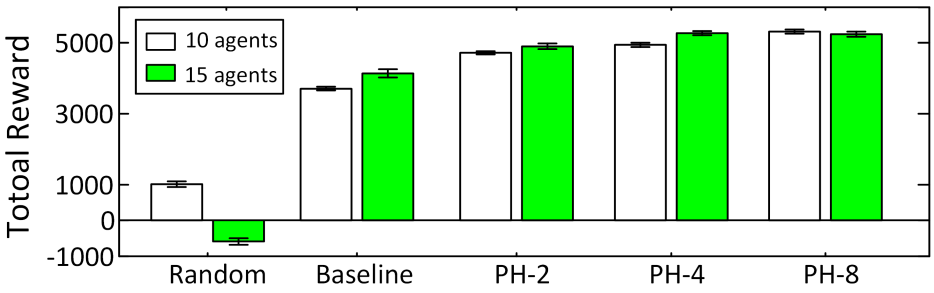


图 2.5　想定 B 中各算法获得的回报值

2.6　本章小结

本章研究了在大范围的部分可观随机环境中分布着信息和威胁的多智能体侦察监视问题。特别地，在单智能体算法中使用预测性启发式来评估向前看几步的策略。对于多智能体算法，扩展了依次计算单个智能体策略的方法到论文的部分可观问题当中。通过实验验证，对于 10 个智能体的问题，本文的算法优于贪婪算法 43%。一方面，由于这是第一个针对在威胁环境和不确定性下多智能体侦察监视问题的算法，在进一步的研究中将探索新的启发式。另一方面，将考虑其他的多智能体团队决策问题，包括智能体之间的通信系统中某个智能体由于遭受伤害导致局部通信故障或者一些智能体可能由于累积伤害过大而死亡时的情景。

第三章　基于树搜索和 max-sum 的分散式侦察监视规划

3.1　引言

本章关注分散式侦察监视问题，由于其中每个无人机负责在某块区域侦察监视而且这些区域中可能有些区域相互重叠，所以在任务中无人机之间需要分散式交互方式。无人机集群常常采用这种方式进行部署，主要是为了满足相互之间避撞以及一些空中交通法规。基于这一特性，本章将证明每个无人机只需要和少数的其他无人机进行交互（即每个无人机与少量无人机进行有限的交互）。这种多智能体之间的稀疏交互（Sparse Interactions）可以看作典型的分布式约束优化问题（Distributed Constraint Optimization Problem, DCOP）[101] 并进行处理。特别地，一种近似算法，max-sum 算法，在求解大规模 DCOPs 中表现出很好的求解效率和求解效果。

然而，如果只是使用 max-sum，则只能求解单步决策（One Shot）问题，将仍然无法处理无人机在执行充满不确定性的多步任务过程。Dec-POMDP[102] 提供了不确定下的多智能体分散式序贯决策制定问题的框架，而且这一框架还可以描述环境以及其他智能体相关信息的不完整性或局部性。然而，由于 Dec-POMDP 问题的高度复杂性，使得其很难扩展并应用于大规模智能体问题 [102]（详见第1.3.2节）。而且，很少有研究基于约束的 Dec-POMDP（描述的情形为最大化某一准则（回报）并限制其他准则（代价）在一定的阈值之内）。

基于这一背景，本章提出一种新的模型来描述在不确定性和威胁环境下的分散式侦察监视问题，并提出一种新颖的算法来求解这一问题。具体地，本章的创新点如下：（1）首次提出了一种在不确定和威胁下的分散式侦察监视问题模型。这种形式化方式不仅考虑了动态环境中的部分可观和非静态特性以及智能体的健康约束，而且明确考虑了分散式交互的运行模式；（2）基于 MCTS 和 max-sum 设计了具有可扩展性的分散式在线规划算法，这种方法的新颖性在于每个智能体构建并扩展一个向前看的搜索树，且通过相互之间分散式的消息传播的方式不断更新各自的搜索树；（3）通过仿真实验证明，对于 6 个智能体的问题，论文的方法的求解结果超过基准测试算法 56%，并且能够扩展至求解超过 24 个智能体的问题。

本章余下各节安排如下：首先介绍分散式交互任务的问题想定，并采用基于 TD-POMDP 的模型对问题进行形式化描述。然后设计基于 MCTS 和 max-sum 的求解算法并进行理论分析。最后设计仿真实验对论文提出的方法进行比较验证。

3.2 问题描述

关于侦察监视的问题，主要通过侦察监视环境、侦察监视智能体及其任务目标来定义。由于集中式和分散式侦察监视智能体均可以部署于相同的环境中，而两者的区别主要是智能体之间的交互方式不同，因此，侦察监视环境的描述参见第2.2.1节，本节主要描述分散式侦察监视问题中侦察监视智能体。

定义 (侦察监视智能体): 一个侦察监视智能体（简称智能体）为一个移动物理实体，通过访问环境中的顶点，能够收集环境中的信息，同时可能遭受环境中威胁的伤害。智能体的集合表示为 $\mathcal{M} = \{A_1, \ldots, A_{|\mathcal{M}|}\}$。

定义 (监视区域): 为每个智能体 $A_m \in \mathcal{M}$ 指定了一个相对较小的区域 $g_m = (V_m, E_m)$ 进行监视，其中 g_m 为 G 的一个子图。每一个监视区域可能和其他的监视区域有交叠（即共享一些顶点和 / 或边）[1]。

图3.1给出了两个问题实例中对于监视区域的划分，其中每个包络 g_m 覆盖智能体 A_m 的监视区域，方块顶点表示智能体的当前位置。

定义 (健康预算): 定义每个智能体 $A_m \in \mathcal{M}$ 在整个 T 时间步长监视过程中健康预算为 $\beta_m \in \mathbb{R}$。健康预算综合反应作战武器平台和操作人员的作战能力的状况。随着作战过程的进行，武器平台可能会受到一定的损害，操作人员的身体状况以及精神状态也可能下降。

智能体的移动和访问的能力定义如下：

定义 (移动): 在图 G 中监视时，每个时刻 t，每个智能体 A_m 位于子图 g_m 中的某个顶点。多个智能体可以同时位于同一个顶点[2]。智能体的移动为基元的，即在子图 g_m 的框架下，发生在任意两个连续时间步之间，即智能体 A_m 由 g_m 中的某个顶点 $v_i \in V_m$ 移动至相邻顶点 $v_i' \in adj_{g_m}(v_i)$。假设 $\forall v_i \in V, v_i \in adj_{g_m}(v_i)$，即一个智能体可以待在某个顶点不动。智能体的速度在一个单位时间之间内足以到达某个相邻的顶点[3]。

[1]监视区域可能有交叠主要原因是不同智能体可能配置不同类型的传感器，而且某些区域需要收集不同类型的数据信息。

[2]例如，某个顶点是一个多个智能体可以同时访问的房间。这个假设是为了包含不同类型的传感器或者一些物体需要同时从多个角度进行观测的情况。

[3]如果某两个顶点之间的距离过大，智能体以最大速度在单位时间内都无法到达时，可以通过在这两个顶点之间添加一个或多个没有信息的顶点。

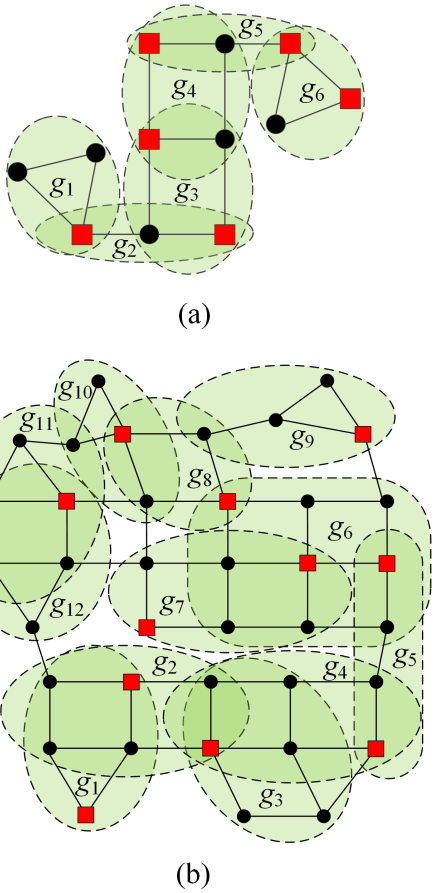


图 3.1 （a）6 个智能体侦察监视和（b）12 个智能体侦察监视的实例

定义 (访问): 每个智能体 A_m 在每个时刻访问当前的位置 v_n。一方面，智能体通过访问 v_n，掌握该位置的当前信息和威胁状态，比如分别为 I_i^n 和 R_j^n。另一方面，这个智能体获得一个回报 $f^n(I_i^n)$，同时从它的健康预算 β_m 中受到损失 $c^n(R_j^n)$。假设智能体对某个顶点进行访问所消耗的时间可以忽略不计。

由于每个顶点的状态随时间随机变化，而且智能体只能观测到当前位置的状态，从而，监视环境为非静态的（即状态集合的联合概率分布随时间变化的）和部分可观的。

定义 (邻居): 智能体 A_m 的一个邻居是指一个和 A_m 的监视区域直接有重叠部分的智能体。A_m 的邻居的集合表示为 $\mathcal{M}_m \subseteq \mathcal{M}$，并假设 $A_m \in \mathcal{M}_m$。

定义 (通信): 每个智能体能且只能和它的邻居之间进行通信。

定义 (合作性能): 为了表示不同个数智能体同时在一个顶点收集信息的能力，定义函数 $\alpha:\{0,\ldots,|\mathcal{M}|\}\rightarrow[0,1]$。$\alpha(n)$ 表示 n 个智能体同时访问某个顶点时所能获得的信息量，其中 $n \in \{0,\ldots,|\mathcal{M}|\}$。

因此，智能体团队需要考虑到历史的观测信息和不同组合的合作性能，通过分散式协调进行决策。特别地，智能体的目标是，最大化收集得到的信息的同时，将累积受到的伤害限制在健康预算当中。

基于以上对环境和智能体的定义，对分散式侦察监视问题的想定进行总结和举例如下。无人机集群的智能体团队 $\mathcal{M}$ 在环境中的 $N > |\mathcal{M}|$ 个位置之间进行移动和访问。智能体在某个位置收集信息的同时，可能受到该位置的威胁所带来的攻击。每个位置的信息和威胁分别按照多态马尔科夫链随时间独立变化，直到某个位置被某个智能体访问时，该位置的信息才能够被智能体确切掌握。然而，这些马尔科夫链的模型是已知的[4]。因此，智能体能够估计环境中没有正在访问的各个位置的状态。通过这一模型，智能体能够根据当前智能体和环境的状况，连续选择将要访问的顶点。

以上定义了分散式侦察监视问题，接下来需要基于环境的模型和智能体的动作及观测的历史信息，设计分散式方法来规划每个智能体的序列动作。下面，将对监视问题形式化描述为 TD-POMDP-HC 模型，并设计相应的求解算法。

3.3 TD-POMDP-HC 形式化建模

3.3.1 TD-POMDP 基本模型

定义 (TD-POMDP): TD-POMDP 基本模型描述的是一组弱耦合的智能体在不确定性环境下的决策过程。一个典型 TD-POMDP[103] 可以表达为元组 $\langle \mathcal{M}, \{\mathcal{S}_m\}, \{\mathcal{A}_m\}, \{\mathcal{O}_m\}, \{\mathcal{T}_m\}, \{\Omega_m\}, \{r_m\}, \{\bar{m}_m\}, \{\mathcal{T}_m^U\}, \{\mathcal{T}_m^L\}, T \rangle$ 如下：

- $\mathcal{M} = \{A_1, \ldots, A_{|\mathcal{M}|}\}$ 为智能体的集合。对于 $|\mathcal{M}| = 1$ 的 TD-POMDP 等价于单智能体的 POMDP 模型。
- $\mathcal{S}_m \subseteq U_m \times L_m \times N_m$ 为智能体 m 的“局部状态”空间，即“非受控”，“局部可控”和“非局部可控”特征空间的交叉乘积。
- $\mathcal{A}_m$ 为智能体 m 的“局部动作”空间。
- $\mathcal{O}_m$ 为智能体 m 的“局部观测”空间。
- $\Omega_m : \mathcal{A}_m \times \mathcal{S}_m \times \mathcal{O}_m \to [0, 1]$ 为智能体 m 的“局部观测函数”。
- $r_m : \mathcal{S}_m \times \mathcal{A}_m \to \mathbb{R}$ 为智能体 m 的局部回报值函数。从而，所有智能体的局部回报值组合得到智能体团队的回报值 $r(s, a) = \sum_{m=1}^{|\mathcal{M}|} r(s_m, a_m)$

[4]实际应用中，模型的转移矩阵可以通过领域专家进行指定，或者通过对大量的训练数据进行复杂的机器学习来获得。

- $\bar{m}_m$ 为智能体 m 的共同模式特征，其中的每个特征至少与一个其他智能体相关。这种共同模式特征使得 TD-POMDP 具有模式依赖的性质。
- $\mathcal{T}_m^U : U_m \times U_m \to [0,1]$ 为非受控特征的转移函数。
- $\mathcal{T}_m^L : \mathcal{S}_m \times \mathcal{A}_m \times L_m \to [0,1]$ 为局部可控特征的转移函数。
- T 为规划的时间长度。

世界状态 s，可以分解成为一些状态特征集合，且其中每个特征表示环境的某一方面特性。换而言之，世界状态空间可以表示为特征域的乘积：$S = (B \times C \times D)$。这种世界状态的分解，使得我们能够将决策模型中的变量（状态特征、观测特征、动作和回报值）之间表示为条件独立的关系 [47, 104]。基于这种分解方式，将可以利用问题中环境信息分布于智能体之间的性质，将 TD-POMDP 的世界状态 s 表示为所有智能体局部状态的集合 $s = \langle s_1, \ldots, s_{|\mathcal{M}|} \rangle$。

在设计具体世界的一个 TD-POMDP 时，设计者需要针对下面定义的一些约束来将一些环境的特征指定至某个智能体 A_m 的局部状态 s_m。具体地，智能体 m 的局部状态 $s_m \in \mathcal{S}_m$ 表示世界状态特征的一个子集且这些特征满足的性质如下：

(1) 对于每个世界状态特征 f，如果没有包含在局部状态 s_m 中，那么将包含在其他某个智能体的局部状态中。

(2) 如果某个世界状态特征 f 对于智能体 A_m 可观，则 f 必须包含在 A_m 的局部状态表示当中。

进一步，不仅团队的回报值可以分解，对于任何联合策略的值也可以分解为局部值。局部值 V_m^π 为团队按照策略 π 执行行动时智能体 A_m 的局部回报值的和：

$$V_m^\pi = \mathbb{E}^\pi \left[\sum_{t=0}^{T} r(s_m^t, a_m^t) \right],$$

定理 3.1: 从而，根据定义得到联合策略 π 的（联合）值可以表示为所有局部值的和：

$$V^\pi = \sum_{m=1}^{|\mathcal{M}|} V_m^\pi.$$

证明：根据定义可知：

$$V^{\pi} = \mathbb{E}^{\pi}\left[\sum_{t=0}^{T} r(s^t, a^t)\right] = \mathbb{E}^{\pi}\left[\sum_{t=0}^{T}\sum_{m=1}^{|\mathcal{M}|} r(s_m^t, a_m^t)\right] = \mathbb{E}^{\pi}\left[\sum_{m=1}^{|\mathcal{M}|} r(s_m^t, a_m^t)\right]$$
$$= \sum_{m=1}^{|\mathcal{M}|} V_m^{\pi}.$$

定义联合历史 h 表示所有智能体联合动作和观测的序列。TD-POMDP 的解可以描述为一个局部策略组合成的联合策略 $\pi = \langle \pi_1, \ldots, \pi_{|\mathcal{M}|} \rangle$，其中 π_m （智能体 A_m 的策略）将智能体集合的联合历史映射为 A_m 动作集合的概率分布。联合策略值函数 $V^{\pi}(h)$ 为由当前时刻 t 开始，采用策略 π 的期望累积回报。

与一般的 Dec-POMDP 所不同，TD-POMDP 模型表示出了信息固有的分布式特性。根据这种分布式特性，可以定义局部模型，来表示与某个智能体 A_m 相关的世界的动态性如下：

定义 (局部模型): TD-POMDP 中智能体 A_m 的局部模型表示为一个多元组 $\langle \mathcal{S}_m, \mathcal{A}_m, \mathcal{O}_m, \mathcal{T}_m, \Omega_m, r_m, \bar{m}_m, T \rangle$。

这种局部模型的表示方式使得每个智能体只需要存储必要的少量信息，避免了传统 Dec-POMDP[5]的不必要的大规模的信息存储。

基于此，定义 TD-POMDP 的解的概念如下：

定义 (最优联合策略): TD-POMDP 的最优联合策略 π^* 为局部策略的一个组合，其中每个局部策略根据局部观测历史来分配智能体的局部动作，实现最大化值函数：$\pi^* \in \arg\max_{\pi} V^{\pi}$。

在 TD-POMDP 模型的基础上，下面重点考虑如何在模型中加入约束。对于每个智能体 $A_m \in \mathcal{M}$ 在 T 时间步长的过程，定义一个有限的续航能力预算 β_m。历史 h_m 表示智能体 A_m 的动作和观测序列。对于动作 a_m 和观测 o_m，智能体 A_m 续航能力的消耗表示为 $c_m(a_m, o_m)$。

对于历史 h_{mt}，定义余下续航能力预算为：

$$b_m(h_{mt}) = \beta_m - \sum_{k=1}^{t} c_m(a_{mk}, o_{mk}), \tag{3.1}$$

然后，基于历史 h_{mt}，智能体决策时需要考虑 A_m 如下的续航能力约束：

$$\text{s.t.} \quad \mathbb{E}^{\pi}\left[\sum_{k=t+1}^{T} c_m(a_{mk}, o_{mk})\right] < b_m(h_{mt}). \tag{3.2}$$

[5] 关于 Dec-POMDP 的定义和介绍详见文献 [102]。

因此，论文通过以下方式扩展 TD-POMDP 模型使其具有描述约束的能力。首先，在每个智能体 A_m 的局部状态变量中，添加变量集合 $\bar{b}_m = \langle b_{m_1}, b_{m_2}, \ldots \rangle$，其中 b_{m_i} 跟踪智能体 $A_{m_i} \in \mathcal{M}_m$ 的剩余续航能力预算。b_{m_i} 初始化为总的续航能力预算 β_{m_i}。然后，在状态转移模型中，对于选择联合动作 $a_{\mathcal{M}_m}$，$A_{m_i} \in \mathcal{M}_m$ 需要的消耗将在相应的剩余预算中扣除：$b_{m_i} \leftarrow (b_{m_i} - c_m(a_{m_i}, o_{m_i}))$。然后在每个智能体 A_m 的动作集合 $\mathcal{A}_m$ 中添加“空”。当 $b_{m_i} \leq 0$ 时，智能体 A_{m_i} 只能执行动作“空”并且从智能体集合中删除。余下的智能体继续执行任务，直到智能体集合变为空集或者终端时刻 T 到达。本文将智能体集合 $\mathcal{M}_m$ 在历史 $h_{\mathcal{M}_m}$ 下的剩余预算表示为 $\bar{b}_m(h_{\mathcal{M}_m})$。因此，续航能力约束通过对每个智能体的局部状态不断检测进行了表达，并且基于威胁模型以及智能体的动作和观测进行随机更新。建立这种约束，是为了保证智能体期望消耗的能量低于当前的续航能力预算（公式（3.2））。基于此，为了求解 TD-POMDP 的算法，本文采用蒙特卡洛仿真生成策略的同时，需要检验续航能力约束（算法 3.1的 9-11 行），后面将对此进行详细介绍。

3.3.2 TD-POMDP-HC

根据 TD-POMDP$\langle \mathcal{M}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \Omega, r, T \rangle$ 的模型，本节首先将没有威胁的分散式侦察监视问题形式化表达如下：

- $\mathcal{M} = \{A_1, \ldots, A_{|\mathcal{M}|}\}$ 为智能体的集合。
- $\mathcal{S}$ 为状态的集合，用来建模智能体的位置以及环境中所有顶点的信息和威胁状态。状态的性质可以分解为每个智能体的局部状态，来描述其局部性质。定义智能体 A_m 的局部状态 s_m，由两个集合组成：$s_m = \langle \bar{e}_m, \bar{v}_m \rangle$。不可控性质 $\bar{e}_m = \langle (e^R_{m_1}, e^R_{m_2}, \ldots), (e^I_{m_1}, e^I_{m_2}, \ldots) \rangle$ 包括智能体 A_m 监视区域 g_m 所有顶点的信息和威胁状态。$\bar{e}_m$ 不受任何智能体的控制，但是可以被多个智能体观测[6] 。局部特性 $\bar{v}_m$ 包括集合 $\mathcal{M}_m$ 中所有智能体的位置。$\bar{v}_m$ 被 A_m 和 $\mathcal{M}_m$ 中的其他智能体控制。
- $\mathcal{A}$ 为联合动作的集合。$\mathcal{A} = \times_{1 \leq m \leq |\mathcal{M}|} \mathcal{A}_m$，其中 $\mathcal{A}_m$ 为智能体 A_m 的动作集合。每个联合动作 a 定义为 $\langle a_1, \ldots, a_{|\mathcal{M}|} \rangle$，其中 $a_m \in \mathcal{A}_m$。同时定义 $a_{\mathcal{M}_m}$ 为智能体集合 $\mathcal{M}_m$ 的联合动作。

  任何智能体 A_m 选择自己监视区域内相邻的顶点进行访问或者选择“空”[7]为一个动作 a_m。

[6]其他问题中的不可控特性的例子包括一天中的时刻或者气温。
[7]当某个智能体“死亡”时，只能执行动作“空”。随后将进行详细描述。

- $\mathcal{O}$ 为联合观测的集合。$\mathcal{O}=\times_{1\le m\le|\mathcal{M}|}\mathcal{O}_m$，其中 $\mathcal{O}_m$ 为智能体 A_m 的观测集合。每个联合观测 o 定义为$\langle o_1,\ldots,o_{|\mathcal{M}|}\rangle$，其中 $o_m\in\mathcal{O}_m$。

  智能体 A_m 的观测 o_m，包括智能体 A_m 的位置，以及该位置的信息和威胁状态。o_m 可以看作 s_m 是一部分。

- $\mathcal{T}$ 为联合转移概率集合。满足 $\mathcal{T}(s'|s,a)=\prod_{1\le m\le|\mathcal{M}|}\mathcal{T}_m(s'_m|s_m,a_{\mathcal{M}_m})$，其中 $\mathcal{T}_m(s'_m|s_m,a_{\mathcal{M}_m})=\mathcal{T}_m(e'_m|e_m)\,\mathcal{T}_m(\bar{v}'_m|\bar{v}_m,a_{\mathcal{M}_m})$ 为 A_m 的局部转移函数。

  由之前定义的马尔科夫模型可知，$\mathcal{T}_m(e'_m|e_m)$ 按照离散时间马尔科夫决策过程进行变换。当 s'_m 为目标位置时，$\mathcal{T}_m(\bar{v}'_m|\bar{v}_m,a_{\mathcal{M}_m})=1$，否则，$\mathcal{T}_m(\bar{v}'_m|\bar{v}_m,a_{\mathcal{M}_m})=0$。

- Ω 为联合观测概率集合。$\Omega(o|s,a)=\prod_{1\le m\le|\mathcal{M}|}\Omega_m(o_m|s_m,a_m)$ 为联合状态 s 下执行联合动作 a 获得联合观测 o 的概率。

  由于智能体 A_m 的一个观测 o_m 即为某些局部状态的一部分，如果局部状态 s_m 和观测 o_m 一致，则观测概率为 $\Omega_m(o_m|s_m,a_m)=1$，否则 $\Omega_m(o_m|s_m,a_m)=0$。

- $r:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ 为可分解的回报函数。$r(s,a)=\sum_m r_m(s_m,a_m)$ 为状态 s 下，执行联合动作 a 的回报值。

  定义 $r_m(s_m,a_m)=\frac{\alpha(n_{v_i})}{n_{v_i}}f^i(e_i^I)$ 为智能体 A_m 所获得的信息值，其中 $v_i\in s_m$ 为当前 A_m 的位置，n_{v_i} 为同时访问 v_i 的智能体个数，$\alpha(n_{v_i})$ 表示这些智能体在该位置所获得的信息。然后，可以得到所有智能体的回报函数为 $r(s,a)=\sum_m r_m(s_m,a_m)=\sum_{v_i\in\vec{v}}\alpha(n_{v_i})f^i(e_i^I)$[8]，即所有智能体所获信息值的总和，其中 $\vec{v}$ 为所有智能体的当前位置向量。

- T 为规划的时间长度。

联合历史 h 表示所有智能体联合动作和观测的序列。TD-POMDP 的解可以描述为一个联合策略 $\pi=\langle\pi_1,\ldots,\pi_{|\mathcal{M}|}\rangle$，其中 π_m（智能体 A_m 的策略）为将智能体集合的联合历史映射为 A_m 动作集合的概率分布。联合策略值函数 $V^\pi(h)$，为由当前时刻 t 开始，采用策略 π 往前的期望累积回报。

进一步，对于每个智能体 $A_m\in\mathcal{M}$ 在 T 时间步长的监视过程，定义一个有限的健康预算 β_m。历史 h_m 表示智能体 A_m 的动作和观测序列。对于动作 a_m 和观测 o_m，智能体 A_m 受到与其位置 v_i 的威胁状态 e_i^R 相关的瞬时伤害 $c_m(a_m,o_m)=c^i(e_i^R)$。

从而，对于历史 h_{mt}，定义健康预算为：

$$b_m(h_{mt})=\beta_m-\sum_{k=1}^{t}c_m(a_{mk},o_{mk}),\tag{3.3}$$

[8]在此，定义的回报函数只包含智能体获得的信息值。随后将对威胁代价建模为模型中的约束。

然后，基于历史 h_{mt}，智能体决策时需要考虑如下的健康约束：

$$\text{s.t.} \quad \mathbb{E}^{\pi}\left[\sum_{k=t+1}^{T} c_m(a_{mk}, o_{mk})\right] < b_m(h_{mt}). \tag{3.4}$$

因此，设计了这种基于健康约束的 TD-POMDP（TD-POMDP with Heath Constraints, TD-POMDP-HC）模型，即通过上文中提到方法的一种扩展 TD-POMDP 模型。

3.3.3 可解耦性证明

定理 3.2: TD-POMDP-HC 的值函数分解为：

$$V^{\pi}(h) = \sum_{m=1}^{|\mathcal{M}|} V_m^{\pi_{\mathcal{M}_m}}(h_{\mathcal{M}_m}), \tag{3.5}$$

其中 $\pi_{\mathcal{M}_m}$ 和 $h_{\mathcal{M}_m}$ 分别为智能体集合 $\mathcal{M}_m$ 的联合策略和联合历史，$V_m^{\pi_{\mathcal{M}_m}}(h_{\mathcal{M}_m})$ 为智能体 A_m 的值函数，即智能体集合 $\mathcal{M}_m$ 从联合历史 $h_{\mathcal{M}_m}$ 开始执行联合策略 $\pi_{\mathcal{M}_m}$ 时智能体 A_m 所获得的期望回报值。

证明：本文定义的 TD-POMDP-HC 模型为文献 [105] 中模型的一种特殊形式。本模型中，智能体拥有局部的参数（每个智能体 A_m 的局部状态 $\bar{v}_m$ 和回报值 $r_m(s_m, a_m)$）。然而，$\bar{v}_m$ 不仅依赖于智能体 A_m 的动作，而且依赖于其他智能体的动作。如文献 [105] 中描述的那样，智能体之间的依赖性描述为一个表达影响（Influence）的直接无环图（Directed Acyclic Graph，DAG），图中的每个顶点表示一个智能体。DAG 中的父 - 子关系表示父母对孩子的局部状态可以进行控制，反之不可以。基于某个智能体父母的策略，该智能体能够计算出自己的值函数。这个性质在本文的模型中表示为智能体 A_m 的值函数因子，即包括了 DAG 中 A_m 的所有祖先。从而，值函数明确存在这种可分解的性质。一个智能体对邻近的孩子们的影响可以总结为一个动态贝叶斯网络（Dynamic Bayesian Network, DBN）[105]。因此，基于指定的联合策略，监视环境能够重新划分为智能体之间没有重叠的监视区域集合。直观上看，值函数 $V_m^{\pi_{\mathcal{M}_m}}(h_{\mathcal{M}_m})$ 为智能体 A_m 在其监视区域当中获得的累积效能，总的值函数 $V^{\pi}(h)$ 为所有智能体在这些相互之间没有覆盖的所有区域上监视所获得的累积效能的总和。

以上，对部分可观的非静态环境下基于健康约束分散式侦察监视问题建模为 TD-POMDP-HC[9]。下面将提出可扩展的分散式算法在线求解 TD-POMDP-HC。

[9]值得注意的是，本文更强调求解监视问题，而不是为了解决 TD-POMDPs 的某一类具体模型的问题。本文的模型泛化了 TD-POMDPs 的建设（即允许局部通信），而且引入了额外的组成（即作

3.4 基于树搜索和 max-sum 的 TD-POMCP 算法

本节提出传递函数解耦的部分可观蒙特卡洛规划（Transition-Decoupled Partially Observable Monte-Carlo Planning，TD-POMCP）算法（见算法3.1）来求解一般的 TD-POMDPs。直接将 MCTS 应用于求解多智能体问题的瓶颈在于，联合动作和联合观测的数量随多智能体个数增加呈指数增长，从而使得向前看的树的分支数量巨大，导致无法计算求解。为了突破这一瓶颈，本文利用值函数的可分解的性质，并行构建许多树，其中每个智能体一个树。具体地，本节首先提出 TD-POMCP 中每个智能体的运行过程；然后介绍智能体之间相互协调的几个阶段，通过引入 max-sum 算法使得智能体之间进行消息传递来实现分散式协作；最后，讨论 TD-POMCP 的收敛性。

3.4.1 TD-POMCP

如图3.2所示，与 POMCP 中采用 MCTS 构建一个单独的搜索树来求解单智能体 POMDP 问题 [2] 所不同[10]，本算法利用智能体之间稀疏交互的性质，同时构建 $|\mathcal{M}|$ 个搜索树，用于求解大规模联合动作和联合观测的问题。所有智能体同时并行构建向前看的树，每个智能体一个树，其中每个智能体 A_m 保持跟踪智能体团队 $\mathcal{M}_m$ 的联合历史、动作和观测。在协调选择动作的同时，智能体采用 MCTS 的算法扩展各自的搜索树。这一技术通过利用多智能体问题的结构，使得 MCTS 算法可以扩展至大量智能体的大规模 TD-POMDP 问题。本节介绍单个智能体运行 TD-POMCP 的过程，并在下一节来介绍协作阶段。

通过元组 $T_m(\hat{h}) = \langle \bar{b}_m(\hat{h}), B_m(\hat{h}), N_m(\hat{h}), V_m(\hat{h}) \rangle$，来表示智能体 A_m 的搜索树中的历史 $\hat{h}$ 处的节点[11]，其中 $\bar{b}_m(\hat{h})$ 表示余下续航能力预算，$B_m(\hat{h})$ 为用于表示信念状态的粒子集合[12]，$N_m(\hat{h})$ 用于统计该节点被访问的次数，以及 $V_m(\hat{h})$ 表示存储值。每个节点 $T_m(\hat{h})$ 同时对每个联合动作 $\hat{a}$ 保存一个值 $V_m(\hat{h}\hat{a})$ 以及对该联合动作的访问次数 $N_m(\hat{h}\hat{a})$，且总的访问的统计满足 $N_m(\hat{h}) = \sum_{\hat{a}} N_m(\hat{h}\hat{a})$。

SEARCH 过程 (line 1-7) 从当前的历史 $\hat{h}_t$ 调用开始。首先信念状态 $B_m(\hat{h})$ 中采样得到起始状态，用于蒙特卡洛仿真 (line 3)。对于 SIMULATION 过程中遇到的每个历史 $\hat{h}$，算法检测余下的续航能力值 (line 9-11)，通过 COORDINATE1[13] 来选择一个动作 $\hat{a}^*$ (line 12)，使用仿真器 $\mathcal{G}(s_m, \hat{a}^*)$ 生成新的状态 s'_m、观测 $\hat{o}$ 和回报 r_m。仿真

战能力约束），这两方面均来源于侦察监视问题本身的性质。

[10]关于 POMCP 算法求解 POMDP 问题的介绍详见附录A。

[11]为了简洁，本节中由 $\hat{h}$ 表示 $h_{\mathcal{M}_m}$，$\hat{a}$ 表示 $a_{\mathcal{M}_m}$ 以及 $\hat{o}$ 表示 $o_{\mathcal{M}_m}$。

[12]通过粒子集合 $B_m(\hat{h})$ 来近似表示智能体 m 在历史 $\hat{h}$ 的信念状态，其中每个粒子对应一个采样状态且信念状态为这些粒子构成的集合。

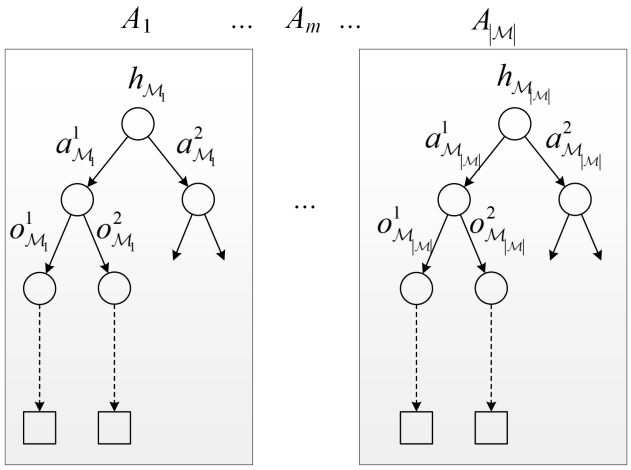


图 3.2　TD-POMCP 中的搜索树

过程中遇到的每个历史 $\hat{h}$，均通过加入对应的仿真状态来更新其对应的信念状态 $B_m(\hat{h})$(line 24)。SEARCH 结束时，智能体从联合动作 $\hat{a}^*$ 中选择自己的动作 a^*_m，从真实世界中接收到一个观测 o_m。

为了处理 TD-POMDP 当中的续航能力约束，采用蒙特卡洛仿真的方式，根据公式 (3.2)，在保证期望的累积损害满足续航能力约束。同时，每个智能体 A_m 通过公式 (3.1) 跟踪余下的续航能力预算。在真实世界执行任务过程中的每个时间步，若续航能力预算用完时，智能体将通知它的邻居在它们的交互列表中将这个智能体删除。类似地，在 SEARCH 的蒙特卡洛仿真过程中，每个智能体通过公式 (3.1) 来跟踪自己的剩余续航能力预算 (line 15)，一旦续航能力预算耗尽 (line 16)，智能体则通知它的邻居们在它们的搜索树中扩展节点时从联合动作中将该智能体的动作移除 (line 18)。

3.4.2　分散式协调

所有的智能体同时运行 TD-POMCP，其中需要两步进行分散式协调：COORDINATE1 (line 12) 用于在 SEARCH 中进行搜索 / 扩展树的时候进行动作的选择；COORDINATE2 (line 6) 用于 SEARCH 结束后选择将要执行的动作。

对于一个 SIMULATE 中的每个联合历史 h，由 $\bar{T}(h) = \langle T_1(h_{\mathcal{M}_1}), \ldots, T_{|\mathcal{M}|}(h_{\mathcal{M}_{|\mathcal{M}|}})\rangle$ 表示所有 $|\mathcal{M}|$ 个树的正在访问的节点。对于每个仿真的每个搜索 / 扩展步，智能体同时并行搜索 / 扩展各自的树，通过最大化置信上限来选择各自的动作：

$$a^* = \arg\max_{a} \sum_{m=1}^{|\mathcal{M}|} U_m(h_{\mathcal{M}_m} a_{\mathcal{M}_m}), \tag{3.6}$$

[13] 与单智能体 MCTS 不同，最优的局部值函数不能保证所有智能体的性能和最优，每个智能体的动作选择需要通过与其他智能体一起协商来得到。稍后将做进一步讨论。

算法 3.1 传递函数解耦的部分可观蒙特卡洛在线规划（TD-POMCP）

procedure SEARCH($\hat{h}$)
  for $i = 1 \rightarrow numSamples$ **do**
  ▷ 从信念向量中采样初始状态：
    $s_m \sim B_m(\hat{h})$
    SIMULATE($s_m, \hat{h}, 0$)
  end for
  ▷ 同其他智能体协调选择一个动作去执行：
  return COORDINATE2($T_m(\hat{h})$)
end procedure

procedure SIMULATE($s_m, \hat{h}, depth$)
  ▷ 检测智能体是否已经死亡：
  if $\bar{b}_m <= 0$ **then**
    return 0
  end if
  ▷ 同其他智能体协调选择一个一个局部联合动作：
  $\hat{a}^* \leftarrow$ COORDINATE1($T_m(\hat{h})$)
  ▷ 使用仿真器 $\mathcal{G}(s_m, \hat{a}^*)$ 生成一个新的状态 s'_m、观测 $\hat{o}$ 和回报 r_m：
  $(s'_m, \hat{o}, r_m) \sim \mathcal{G}(s_m, \hat{a}^*)$
  if $T_m(\hat{h}\hat{a}^*\hat{o}) == null$ **then**
  ▷ 通过公式 *(3.1)* 来更新 $\bar{b}_m$：
    $\bar{b}'_m \leftarrow$ UPDATE($\bar{b}_m$)
    if $\bar{b}'_m <= 0$ **then**
      $\bar{b}'_m \leftarrow 0$
      DELETE(A_m)
    end if
  ▷ 扩展搜索树：
    $T_m(\hat{h}\hat{a}^*\hat{o}) \leftarrow \langle \bar{b}'_m, \emptyset, N_{init}, V_{init} \rangle$
    return ROLLOUT($s'_m, \hat{h}\hat{a}^*\hat{o}, depth + 1$)
  end if
  $R_m \leftarrow r_m +$ SIMULATE($s'_m, \hat{h}\hat{a}^*\hat{o}, depth + 1$)
  $B_m(\hat{h}) \leftarrow B_m(\hat{h}) \cup \{s_m\}$
  $N_m(\hat{h}) \leftarrow N_m(\hat{h}) + 1$
  $N_m(\hat{h}\hat{a}^*) \leftarrow N_m(\hat{h}\hat{a}^*) + 1$
  $V_m(\hat{h}\hat{a}^*) \leftarrow V_m(\hat{h}\hat{a}^*) + (R_m - V_m(\hat{h}\hat{a}^*))/N_m(\hat{h}\hat{a}^*)$
  return R_m
end procedure

procedure ROLLOUT($s_m, \hat{h}, depth$)
  $\hat{a}^* \sim \pi_{rollout}(\hat{h}, \cdot)$
  $(s'_m, \hat{o}, r_m) \sim \mathcal{G}(\hat{s}, \hat{a}^*)$
  return $r +$ ROLLOUT($s'_m, \hat{h}\hat{a}^*\hat{o}, depth + 1$)
end procedure

其中 $U_m(h_{\mathcal{M}_m}a_{\mathcal{M}_m}) = V_m(h_{\mathcal{M}_m}a_{\mathcal{M}_m}) + c\sqrt{\frac{\log N(h_{\mathcal{M}_m})}{N(h_{\mathcal{M}_m}a_{\mathcal{M}_m})}}$，$V_m(h_{\mathcal{M}_m}a_{\mathcal{M}_m})$ 为智能体 A_m 的基于 $h_{\mathcal{M}_m}$ 和 $a_{\mathcal{M}_m}$ 的值。在 SEARCH 之后，通过最大化联合动作的值来选择各自

的动作：

$$a^* = \arg\max_{a} \sum_{m=1}^{|\mathcal{M}|} V_m(h_{\mathcal{M}_m} a_{\mathcal{M}_m}). \tag{3.7}$$

接下来，将介绍如何将求解公式 (3.6) 和 (3.7) 对应的问题看作是 DCOPs，以及怎样使用 max-sum 算法对其进行求解。

3.4.3 动作选择

本节以公式（3.6）为例详细介绍如何将动作选择问题定义为一个典型的 DCOP$\langle \mathcal{M}, \mathcal{X}, \mathcal{D}, \mathcal{V} \rangle$ 并使用 max-sum 进行求解（附录B详细介绍典型 DCOP 模型以及 max-sum 算法）。具体地，max-sum 并不是执行一次优化，而是在协调体系中通过一种分布式消息传播（Message Passing）的方式来运行。这种消息传播的协议允许智能体在任务推进过程中通过计算每个可能的联合策略的效用值，持续不断地进行决策制定。其中的这些效用值，通常表示为因子图中的函数节点。

DCOP 问题的算法已经成功应用在了大规模多智能体问题中 [101]。max-sum[63]，作为其中的一种近似算法，对于大规模 DCOPs 表现出良好的求解性能，而且不随智能体的个数增加指数增长。本文使用 max-sum 进行 TD-POMCP 中对公式 (3.6) 和 (3.7) 的动作选择。每个智能体负责对自己的动作变量的选择，并向邻居智能体传递消息。因此，尽管 max-sum 能够近似得出全局优化问题的解（即公式 (3.7)），但是只涉及到了局部通信与计算。

通过如下 5 个步骤具体使用 max-sum （详细过程参考文献 [106]）：

- 第 1 步 - 定义变量。

  $\mathcal{X} = \{a_1, \ldots, a_{|\mathcal{M}|}\}$ 为动作变量的集合，每个变量由唯一一个智能体控制。$\mathcal{D} = \{\mathcal{A}_1, \ldots, \mathcal{A}_{|\mathcal{M}|}\}$ 为变量的域的集合，且每个变量 a_m 从 $\mathcal{A}_m$ 中进行选择。实际应用中，每个动作可以指定为无人机可能执行的机动动作，并在离散化的运动空间中进行描述。比如，一个固定翼无人机的动作可以描述为其可以遵循的攻角的集合。由于变量的个数及其域的大小将直接影响 max-sum 的性能，设计者需要谨慎设计变量。具体地，一方面，从通讯的角度，从某个变量节点 a_m 发出的一条 max-sum 消息的长度与其变量的域 $\mathcal{A}_m$ 的大小呈线性关系，为 $O(|\mathcal{A}_m|)$；另一方面，从计算的角度，计算出从某个变量节点 a_m 发出一条 max-sum 消息的复杂度为 $O(\prod_{i\in} |\mathcal{A}_k|)$

- 第 2 步 - 定义函数。

  $\mathcal{V} = \{V_1, \ldots, V_{|\mathcal{M}|}\}$ 为函数的集合，并定义函数 $V_m = V_m(h_{\mathcal{M}_m} a_{\mathcal{M}_m})$ 为公式 (3.6) 中的第 m 项且依赖于 $a_{\mathcal{M}_m} \subseteq \{a_1, \ldots, a_{|\mathcal{M}|}\}$。因子图中的每个函数定量

表示其对联合策略对应的全局约束函数的值的影响（公式B.1）。直观上理解，每个无人机对应的效用函数可以看做其在整个任务中累积回报的期望值。

- 第 3 步 - 分配节点。

  为了计算因子图中的函数和变量，需要分别分配相应的计算资源。由于团队中的无人机往往是不同（异构）的，并且计算资源有限，这种计算资源的分配显得相当重要。论文将每个变量分配给其对应决策的智能体，并将每个函数变量分配给其对应效用表示的智能体。

- 第 4 步 - 选择一个消息传播计划。

  需要设计计划，来实现基于接收到的消息来计算新的消息和策略。

- 第 5 步 - 更新邻居。

  无人机团队成员之间按照因子图进行消息传播。通常情况下，每个节点与固定的邻居进行接收和发送消息。然而，由于环境和问题的动态性，这种相互之间的邻居有可能随时间发生变化。所以，在执行任务的过程中，需要对因子图不断进行更新，并确定每个节点的新的邻居。

值得注意的是，算法3.1通过使用 max-sum，每个智能体只需要和相邻的有限几个智能体进行通信，而且其局部的值的计算只与这些邻居的历史和策略相关。基于此，算法3.1通过分散式的方式结合有限的通信进行运算，能够快速收敛到近似的最优解。这种方式比文献 [107] 中的基于全通信的方法更为有效和快速。具体地，文献 [107] 中的算法需要所有智能体共享它们的观测，然而实际问题中这种观测可能会涉及到大量的传感器数据，只能等待接收完所有观测数据才能开始计算智能体策略，这对于带宽有限的情况将会造成严重的延迟。从而，[107] 中的算法很难满足在线需求。然而，本算法由于采用了 max-sum，智能体之间只需要传输非常有限的局部信息。而且，max-sum 是一种任意时间算法[14]，使得本文的算法能够满足实际应用中严格的在线求解条件。此外，这种分散式的方式，不会出现中心点丢失或者通信堵塞的情况，对于智能体数量的增加具有良好的扩展性。

3.5 基于简洁信念表示的 TD-FMOP 算法

本节中，首先定义对环境状态的特征信念表示，然后提出传递函数解耦的基于特征信念的蒙特卡洛在线规划（Transition-Decoupled Factored Belief based Monte-Carlo Online Planning，TD-FMOP）算法来求解论文定义的 TD-POMDP-HC。

[14]任意时间算法是指算法在运算过程中任意时间被打断时都能够返回一个可执行的解。

3.5.1 环境状态的特征信念

每个智能体的局部状态均为部分可观的。具体地，对于局部状态 $s_m = \langle \bar{e}_m, \bar{v}_m, \bar{b}_m \rangle$，$\bar{v}_m$ 和 $\bar{b}_m$ 中的元素均为完全可观，而 $\bar{e}_m$ 中的环境元素不能直接可观。因此，首先介绍环境状态 $\bar{e}_m$ 的直接信念表示，其维数随顶点个数呈指数增长，从而使得 TD-POMDP-HCs 无法求解。替代地，进一步设计 $\bar{e}_m$ 的一个特征信念表示，其维数随顶点个数增加呈线性增长，从而很大程度上降低内存空间的需求[15]。

首先介绍对于智能体 A_m 的环境状态信念的直接表示，即通过定义所有可能状态的联合概率分布。具体地，对于历史 $h_{\mathcal{M}_m}$[16]，智能体对于 $\bar{e}$ 的内部状态可以归结为信念向量 $\Lambda_e(h) = [\lambda_1(h), \ldots, \lambda_{Num\_e}(h)]$，其中 $\lambda_i(h)$ 为向量 $\bar{e}$ 的第 i 个环境状态的条件概率，$Num\_e = \prod_{n=1}^{N} K_R^n K_I^n$ 为所有可能的环境状态的个数。然而，$\Lambda_e(h)$ 的维数随顶点个数 N 的增加呈指数增长，使得 TD-POMDP-HC 无法求解计算。基于 $\Lambda_e(h)$，可以采用第2.3.2节中的方法定义的维数随顶点个数 N 的增加呈线性增长的信念表示方式 $\Psi(h)$。下面，基于这种简洁的信念表示方式，进行分散式在线规划算法的设计。

3.5.2 TD-FMOP

基于特征信念表示和可分解的值函数（公式（3.5）），本节设计基于 MCTS 的传递函数解耦的基于特征信念的蒙特卡洛在线规划（Transition-Decoupled Factored Belief based Monte-Carlo Online Planning，TD-FMOP）（见算法3.2）用于求解多智能体分散式在线规划。与 TD-POMCP 中采用粒子滤波器来表达信念状态所不同，TD-FMOP 将根据 TD-POMDP-HC 的这类问题中的环境特性建立真实的信念状态并将该信念状态用于在线规划过程中。

与 TD-POMCP 相同，TD-FMOP 也同时构建 $|\mathcal{M}|$ 个搜索树。本节只介绍单个智能体运行 TD-FMOP 的过程，多智能体协调进行策略选择的方式与 TD-POMCP 相同（详见第3.4.2和3.4.3节）。

通过 $T_m(\hat{h}) = \langle \bar{v}_m(\hat{h}), \bar{b}_m(\hat{h}), \Psi_m(\hat{h}), N_m(\hat{h}), V_m(\hat{h}) \rangle$ 表示智能体 A_m 的搜索树中的历史 $\hat{h}$ 处的节点[17]，其中 $\bar{v}_m(\hat{h})$，$\bar{b}_m(\hat{h})$ 和 $\Psi_m(\hat{h})$ 分别为智能体集合 $\mathcal{M}_m$ 的位置集合、余下作战能力预算和特征信念向量，$N_m(\hat{h})$ 用于统计该节点被访问的次数，以及 $V_m(\hat{h})$ 表示存储值。每个节点 $T_m(\hat{h})$ 同时对每个联合动作 $\hat{a}$，保存一个值 $V_m(\hat{h}\hat{a})$ 以及对该联合动作的访问次数 $N_m(\hat{h}\hat{a})$，总的访问次数的统计满足

[15] 根据文献 [108] 中的结论可知，对于设计最优策略，只需要对状态的不直接可观部分定义信念表示即可。因此，本算法只需要对 s_m 中的 $\bar{e}_m$ 定义信念表示即可。

[16] 不失一般性，为了简化符号化表示，本节中假设 $\mathcal{M}_m = \mathcal{M}$，从而 $h_{\mathcal{M}_m} = h$。

算法 3.2 传递函数解耦的基于特征信念的蒙特卡洛在线规划（TD-FMOP）

1: **procedure** SEARCH($\hat{h}$)
2: **for** $i = 1 \rightarrow numSamples$ **do**
▷ 从特征信念向量中采样得到环境状态的一个样本：
3: $\bar{e}_m \sim \Psi_m(\hat{h})$
4: $s_m \leftarrow (\bar{v}_m(\hat{h}), \bar{b}(\hat{h}), \bar{e}_m)$
5: SIMULATE($s_m, \hat{h}, 0$)
6: **end for**
▷ 同其他智能体协调选择一个动作去执行：
7: **return** COORDINATE2($T_m(\hat{h})$)
8: **end procedure**

9: **procedure** SIMULATE($s_m, \hat{h}, depth$)
▷ 检测智能体是否已经死亡：
10: **if** $\bar{b}_m <= 0$ **then**
11: **return** 0
12: **end if**
▷ 同其他智能体协调选择一个一个局部联合动作：
13: $\hat{a}^* \leftarrow$COORDINATE1($T_m(\hat{h})$)
▷ 使用仿真器 $\mathcal{G}(s_m, \hat{a}^*)$ 生成一个新的状态 s'_m、观测 $\hat{o}$ 和回报 r_m：
14: $(s'_m, \hat{o}, r_m) \sim \mathcal{G}(s_m, \hat{a}^*)$
15: **if** $T_m(\hat{h}\hat{a}^*\hat{o}) == null$ **then**
▷ 通过公式 *(3.3)* 来更新 $\bar{b}_m$：
16: $\bar{b}'_m \leftarrow$ UPDATE($\bar{b}_m$)
17: **if** $\bar{b}'_m <= 0$ **then**
18: $\bar{b}'_m \leftarrow 0$
19: DELETE(A_m)
20: **end if**
▷ 通过公式 *(2.5)* 更新 $\Psi_m(\hat{h})$：
21: $\Psi_m(\hat{h}\hat{a}^*\hat{o}) \leftarrow$ UPDATE($\Psi(\hat{h}), \hat{a}^*, \hat{o}$)
▷ 扩展搜索树：
22: $T_m(\hat{h}\hat{a}^*\hat{o}) \leftarrow \langle \bar{v}'_m, \bar{b}'_m, \Psi_m(\hat{h}\hat{a}^*\hat{o}), N_{init}, V_{init} \rangle$
23: **return** ROLLOUT($s'_m, \hat{h}\hat{a}^*\hat{o}, depth + 1$)
24: **end if**
25: $R_m \leftarrow r_m +$ SIMULATE($s'_m, \hat{h}\hat{a}^*\hat{o}, depth + 1$)
26: $N_m(\hat{h}) \leftarrow N_m(\hat{h}) + 1$
27: $N_m(\hat{h}\hat{a}^*) \leftarrow N_m(\hat{h}\hat{a}^*) + 1$
28: $V_m(\hat{h}\hat{a}^*) \leftarrow V_m(\hat{h}\hat{a}^*) + (R_m - V_m(\hat{h}\hat{a}^*))/N_m(\hat{h}\hat{a}^*)$
29: **return** R_m
30: **end procedure**

31: **procedure** ROLLOUT($s_m, \hat{h}, depth$)
32: $\hat{a}^* \sim \pi_{rollout}(\hat{h}, \cdot)$
33: $(s'_m, \hat{o}, r_m) \sim \mathcal{G}(\hat{s}, \hat{a}^*)$
34: **return** $r +$ ROLLOUT($s'_m, \hat{h}\hat{a}^*\hat{o}, depth + 1$)
35: **end procedure**

$N_m(\hat{h}) = \sum_{\hat{a}} N_m(\hat{h}\hat{a})$。

SEARCH 过程 (line 1-8) 从当前的历史 $\hat{h}_t$ 调用。由于 $\bar{v}_m(\hat{h}_t)$ 和 $\bar{b}_m(\hat{h}_t)$ 直接可

[17]为了简洁，本节中由 $\hat{h}$ 表示 $h_{\mathcal{M}_m}$，$\hat{a}$ 表示 $a_{\mathcal{M}_m}$ 以及 $\hat{o}$ 表示 $o_{\mathcal{M}_m}$。

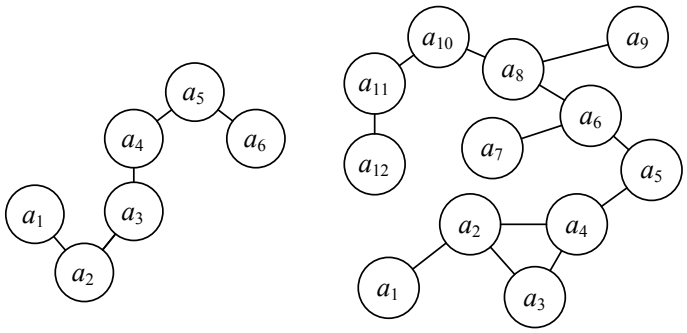


图 3.3　6 智能体侦察和 12 智能体侦察问题的约束图

观，智能体只需基于特征信念向量 $\Psi_m(\hat{h}_t)$ 采样得到环境状态，用于蒙特卡洛仿真 (line 3)。对于 SIMULATION 过程中遇到的每个历史 $\hat{h}$，算法检测余下的作战能力值 (line 10-12)，通过 COORDINATE1 来选择一个动作 $\hat{a}^*$ (line 13)，使用仿真器 $\mathcal{G}(s_m, \hat{a}^*)$ 生成新的状态 s'_m、观测 $\hat{o}$ 和回报 r_m，并且通过公式 (2.5) 更新特征信念向量 $\Psi_m(\hat{h})$ (line 21)。SEARCH 结束时，智能体从联合动作 $\hat{a}^*$ 中选择自己的动作 a^*_m，从真实世界中接收到一个观测 o_m。

为了处理 TD-POMDP-HC 当中的作战能力约束，采用蒙特卡洛仿真的方式，根据公式 (3.4)，在保证期望的累积损害满足作战能力约束的同时，生成监视策略。具体地，每个智能体 A_m 通过公式 (3.3) 跟踪余下的作战能力预算。在真实世界监视过程中，在每个时间步，若作战能力预算用完时，智能体将通知它的邻居将其在它们的交互列表中将这个智能体删除。类似地，在算法3.2的 SEARCH 过程的蒙特卡洛仿真过程中，每个智能体通过公式 (3.3) 来跟踪自己的剩余作战能力预算 (line 16)。一旦作战能力预算耗尽 (line 17)，智能体应该通知它的邻居们在它们的搜索树中扩展节点时从联合动作中将该智能体的动作移除 (line 19)。

所有的智能体同时运行 TD-FMOP，其中需要两步进行分散式协调：COORDINATE1 (line 13) 用于在 SEARCH 中进行搜索 / 扩展树的时候进行动作的选择；COORDINATE2 (line 7) 用于 SEARCH 结束后选择将要执行的动作。这种分散式协调的方式与 TD-POMCP 中相同（详见第3.4.2和3.4.3节）。对于图3.1中的 6 智能体侦察监视和 12 智能体侦察监视问题，区域交叠的智能体之间存在约束，需要相互协调进行动作的选择；图3.3 给出了相应的交互关系图, 其中每两个智能体之间的连线表示相互之间的侦察区域有交叠，动作选择相互约束，从而需要相互协调进行动作选择。

表 3.1 算法和模型

	模型	算法
集中式	MPOMDP-HC	FMOP, FMOP*, POMCP
分散式	TD-POMDP-HC	TD-FMOP, TD-POMCP

3.6 算法比较与性能分析

本节首先对论文提出的模型以及算法之间的关系进行比较和分析。然后对问题的求解复杂度以及论文提出的算法性能进行理论证明与分析。由于 TD-POMCP 和 TD-FMOP 中多智能体分散式协调采用的是近似算法 max-sum，而 max-sum 求解有环因子图问题的最优性和收敛性均没有理论保证，如果不使用 max-sum，而是使用一些精确的算法（本节简称 Exact-DCOP 算法），如 OptAPO [29]、ADOPT [33] 和 DPOP [36] 等，TD-FMOP 及 TD-POMCP 将能够达到很好的最优性和收敛性，然而计算代价将随问题的规模的扩大呈指数增长。与精确算法相比，max-sum 的这种分布式求解方式，使得问题复杂度与智能体个数为多项式关系，能够满足实时性要求。本节将分别从理论上证明基于精确算法的 TD-POMCP 和 TD-FMOP 的收敛性和最优性，并以此为参考分析基于 max-sum 的 TD-POMCP 和 TD-FMOP 的最优性和收敛性。

3.6.1 算法比较

首先，TD-POMCP 可以看作扩展 POMCP 以分散式交互的方式求解 TD-POMDP 模型的问题。关于 TD-POMDP 和 TD-FMOP 的区别主要表现在采用两种不同的方式表示信念状态：TD-POMCP 采用粒子滤波器而 TD-FMOP 采用具体问题的特征信念表示。与 POMCP 类似，TD-POMCP 为一类通用的算法，并且可以用于问题的信念状态很难直接表示或者表示很复杂的情况。

然后，允许智能体之间全通信的 POMDP-HC，可以看作集中式的 TD-POMDP-HC。由于全通信的 MPOMDP 可以简化为一种采用集中式控制性来执行联合动作和接收联合观测的 POMDP，MPOMDP-HC 同样可以看作一种集中式 POMDP 的实例。此外，全通信的 TD-FMOP 可以集中选择联合策略，并记这种算法为 FMOP。

接下来，定义一种 FMOP 的扩展算法，FMOP*，来平衡考虑长时间维度的健康约束与短时间可能收集的信息的关系。FMOP 中定义的参数 τ 确定 MCTS 中向前看的步数，同时问题余下的健康预算需要维持的步数为 $T-t$，其中 t 为当前时间。从而，在每个智能体使用 FMOP* 进行规划时，定义并使用一个 τ 时间段的启发式预算 $\tilde{b}_m(h_{mt}) = b_m(h_{mt})\frac{\tau(t)}{T-t}$。

最后，如表3.1给出以上涉及的模型和算法之间的对照关系[18]。下一节将对这些算法的性能进行仿真实验比较。

3.6.2 复杂度

首先简单介绍一般 Dec-POMDP 的求解复杂度。文献 [102] 中证明了有限视野（Finite-horizon）Dec-POMDP 为 NEXP 复杂度（即便是两个智能体的 Dec-POMDP 问题仍然是 NEXP 难）。这意味着采用任何算法计算最优联合策略的时间，对于最差的情况，与问题描述的规模[19]的增加呈双指数增长，将很难进行求解。

然后，分析求解一般 TD-POMDP-HC 问题最优解的最差情况的复杂度。由于 TD-POMDP-HC 问题为一般 Dec-POMDP 问题的子集，所以其计算复杂度不超过 Dec-POMDP 问题的计算复杂度。虽然 TD-POMDP-HC 在 Dec-POMDP 的基础上利用了一些问题的结构特性，但是其计算复杂度仍然没有降低。基于其他的相关工作 [103, 109, 110]，可以推导 TD-POMDP-HC 问题的最差情况的复杂度。

定理 3.3: TD-POMDP-HC 的复杂度为 NEXP 完备。

证明：一方面，根据文献 [103] 中的方法，可以通过将一种事件驱动交互的分散式 MDP（Decentralized MDP with Event Driven Interaction, EDI-DEC-MDP）[109] 转化成为 TD-POMDP-HC（转化过程参见文献 [103] 中的附录 A），而 EDI-DEC-MDP 的下界已经证明为 NEXP 完备。另一方面，TD-POMDP-HC 可以看做是 Dec-POMDP 的子类。因此，TD-POMDP-HC 的复杂度为 NEXP 完备。

从而可知，TD-POMDP-HC 随问题描述的规模的扩大呈双指数增长。

然而，论文设计的基于 max-sum 的 TD-POMCP 和 TD-FMOP 算法通过迭代的分散式消息传播，大大降低了计算时间（详见第3.4节）。

3.6.3 收敛性和最优性

本节依次分析 FMOP、TD-FMOP 和 TD-POMCP 的收敛性和最优性。

定理 3.4: FMOP 能够最优求解 POMDP-HC。且访问次数 $N(h)$ 接近无穷时，其与最优值函数的偏差 $E[V(h)/V^*(h)]$ 为 $O(logN(h)=N(h))$。

[18]值得注意的是，虽然论文提出的一些算法面向求解基于健康约束的问题，但是这些算法均可以应用于求解不带约束的问题。

[19]问题描述的规模是指存储完整的具体问题模型所需空间大小，比如 MDP 的多元组 $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r \rangle$，以及 POMDP 的多元组 $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r, \mathcal{O}, \Omega, T \rangle$。

证明：Silver 和 Veness[2] 建立了 MCTS 方法在线求解 POMDP 问题的最优性条件：其搜索过程中能够从真实信念状态中进行采样。由定理2.1可知，FMOP 的信念状态向量等价于问题的真实信念状态，从而 FMOP 能够最优求解 POMDP-HC。与文献 [8] 中分析 UCT 应用到 PO-UCT 类似，访问次数 $N(h)$ 接近无穷时，FMOP 的解的值与最优值函数的偏差 $E[V(h)/V^*(h)]$ 为 $O(logN(h) = N(h))$。

定理 3.5：访问次数 $N(h)$ 接近无穷时，基于 Exact-DCOP 算法的 TD-FMOP 能够收敛到 TD-POMDP-HC 的最优解。

证明：基于 TD-POMDP-HC 值函数的可解耦特性（定理3.2），只要求解算法能够得到满足最大化 TD-POMDP-HC 值函数的解，也即能够得到最优解。基于 TD-FMOP 中的搜索树结构（图3.2），Exact-DCOP 能够实现智能体之间动作的最优选择，并且类似于 FMOP 能够保证蒙特卡洛树搜索（定理3.4）收敛到局部最优值函数，从而访问次数 $N(h)$ 接近无穷时，TD-FMOP 解的值能够收敛到值函数的最大值。收敛性的分析参照文献 [107] 中 FV-POMCP 对 MPOMDP 求解的分析。

类似可知，基于 Exact-DCOP 算法的 TD-POMCP 和 TD-FMOP 能够最优求解一般的 TD-POMDP。

然后，max-sum 使用 DCOPs 中的因子图表示，并且存在环，以至于收敛性遭到破坏 [111]。众所周知，对于无环的因子图，max-sum 能够保证收敛到最优解，但是有环存在的因子图不能保证最优。然而，大量的实验证明 DCOPs 的一系列算法均能得到很好的近似解 [63, 111, 112]。总之，TD-POMCP 和 TD-FMOP 在理论上具有很好的收敛性，将在下面的实验过程中对其进行验证。

3.7 仿真实验验证

基于第1.2节的侦察监视的任务流程，指挥员首先指定出一些需要使用无人机进行侦察监视的位置。然后，无人机连续飞行于这些位置之间进行侦察监视来实时更新一些具体的信息，来用于指挥员来进一步安排救援资源的分配与使用（比如支援我方某进攻力量、摧毁打击某个地方目标等）。为了对论文中规划方法进行评价，我们假设指挥员制订了如图3.1中的监视图。基于此，下面比较集中式的算法 FMOP 和 POMCP，以及分散式算法 TD-FMOP 和 TD-POMCP，其中 POMCP 是当前最好的通用 POMDP 在线求解算法。对于侦察监视的想定，定义一些不同的针对各个顶点的 3 状态马尔科夫信息模型以及 2 状态马尔科夫威胁模型，并将这些模型以及不同的信息和伤害值函数分布于每个想定的各个顶点中。对于每个实验想定以及每个算法，运行 100 次仿真并通过平均总回报以及运行时间来评价每

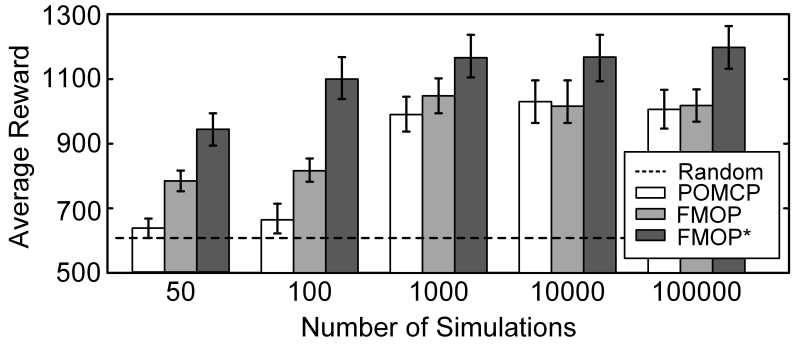


图 3.4　2 个智能体侦察监视问题的获得的平均回报值

个算法的性能。这些算法的运行环境为 2.6 GHz Intel dual core CPU 和 1GB RAM 的计算机。

3.7.1　特征信念与启发式预算的实验评价

为了评价特征信念的性能，将 POMCP、FMOP 和 FMOP* 应用于如下的一个 POMDP-HC：2 个智能体在一个 12 个顶点 (3×4) 的矩形网格上进行侦察监视，且每个智能体移动于所有顶点之间执行任务。FMOP 可以看作是组合了 POMCP 和特征信念表示的方法，而 FMOP* 为通过启发式预算对 FMOP 进行扩展而来的方法。两个智能体的健康预算分别设置为 100 和 150。对于这个两个智能体侦察监视的问题，包含了约 5^2 个联合动作，6^2 个联合观测和约 6^{12} 个环境状态。智能体在这个不断随机变化的环境中进行连续 200 个时间步长的侦察监视任务，并且每个时刻通过考虑向前 10 步的任务来决策当前 1 步的行为策略。

对于采用不同算法进行不同次数仿真，如图3.4给出了所获得的平均总回报值，且表3.2给出了其运行时间。首先比较 FMOP 和 POMCP，可以看出对于每个仿真个数，FMOP 表现均强于 POMCP，这是得益于其精确的信念表示。对于小量仿真的情况，进行 50 次仿真时 FMOP 的表现超过 POMCP 达 21.19%，对于 100 次仿真时超过 19.03%。FMOP 和 POMCP 均为基于 MCTS 的算法，从而其性能随仿真次数的增长而增长。然后对 FMOP* 和 FMOP 与 POMCP 进行比较。对于这个 200 时间步长的问题，智能体采用 FOMP* 能够覆盖更多的环境区域。实验结果证明，对于 FMOP* 仅仅使用 100 次仿真所获得的性能，均优于实验中各个次数仿真的 FMOP 和 POMCP 所获得的性能。

3.7.2　可扩展性的实验评价

对于 TD-POMDP-HC 的求解，论文实验验证 TD-FMOP、TD-POMCP 及 FMOP 的有效性和可扩展性，并以 POMCP 作为基准算法进行对比验证[20]。 构建如下两

[20]在使用 FMOP 和 POMCP 对 TD-POMDP-HC 进行求解时假设智能体之间能够进行全局通信。

表 3.2　2 个智能体侦察监视问题的仿真运行时间 (s)

	50	100	1000	10000	100000
FMOP*	0.11	**0.14**	1.18	9.22	74.98
FMOP	0.11	0.18	**1.73**	16.27	223.5
POMCP	0.05	0.09	**3.46**	20.56	217.7

个想定：

- **想定 A**：如图3.1 (a) 所示，6 个智能体在一个具有 12 个顶点 15 条边的环境中进行侦察监视，其中指定了 6 个具体的监视区域且每个智能体约有 2 个邻居。对于 FMOP 和 POMCP，6 个智能体的联合动作和联合观测约为 3^6 和 6^6，而对于 TD-FMOP 和 TD-POMCP，局部联合动作和局部联合观测的个数分别约为 6×3^3 和 6×6^3。

- **想定 B**：如图3.1 (b) 所示，12 个智能体在一个具有 38 个顶点 59 条边的环境中进行侦察监视，其中指定了 12 个具体的监视区域且每个智能体约有 3 个邻居。对于 FMOP 和 POMCP，12 个智能体的联合动作和联合观测约为 3^{12} 和 6^{12}，而对于 TD-FMOP 和 TD-POMCP，局部联合动作和局部联合观测的个数分别约为 12×3^4 和 12×6^4。

- **想定 C**：24 个智能体在一个具有 76 个顶点 119 条边的环境中进行侦察监视，其中指定了 24 个具体的监视区域且每个智能体约有 3 个邻居。对于 FMOP 和 POMCP，24 个智能体的联合动作和联合观测约为 3^{24} 和 6^{24}，而对于 TD-FMOP 和 TD-POMCP，局部联合动作和局部联合观测的个数分别约为 24×3^4 和 24×6^4。

对于所有的想定，智能体在每个图中进行 10 个时间步长的侦察监视。对于想定 A，主要比较 4 个算法求解这个 6 智能体侦察监视问题，并且将不同算法采用不同次数仿真求得的平均回报值在表3.3中给出。对于某个仿真个数，分布式算法 TD-FMOP 和 TD-POMCP 的性能分别明显优于 FMOP 和 POMCP，且 TD-FMOP 表现最好。对于小量次数仿真的情况，在进行 50 次仿真时，TD-FMOP 的表现超过 TD-POMCP 达 46.48%，对于 100 次仿真时超过 30.24%，而且对于 100,000 仿真时的性能超过所有其他算法 90%。对于仿真次数少于等于 100 次时，TD-FMOP 超过 POMCP 达 56.72%。

论文通过想定 B 和 C 进行实验，来评价 TD-FMOP 的可扩展性。对于这些想定，采用每个算法进行不同次数仿真实验所获得的平均回报和运行时间如表3.4和3.5所示。对于 24 个智能体的想定 C，随机算法所获得的平均回报为 520.4。

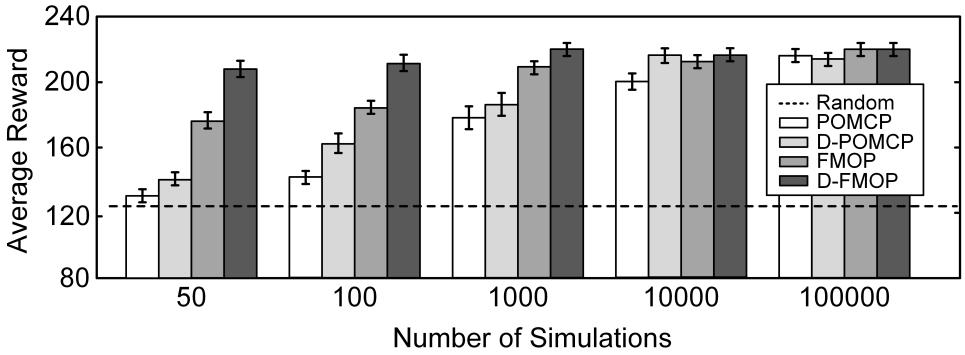


图 3.5　6 个智能体侦察监视问题的获得的平均回报值

表 3.3　6 个智能体侦察监视问题的仿真运行时间 (s)

	50	100	1000	10000	100000
TD-FMOP	0.41	**0.88**	9.02	93.26	1059.44
FMOP	0.15	0.27	**3.39**	45.12	468.60
TD-POMCP	0.13	0.39	6.87	**93.38**	1146.23
POMCP	0.09	0.17	3.34	64.04	**588.46**

表 3.4　不同仿真次数的 12 个智能体的运算结果

	50		100		1000	
	值	时间 (s)	值	时间 (s)	值	时间 (s)
TD-FMOP	**408.2**	**18.06**	412.8	39.78	438.1	578.6
FMOP		–		–		–
TD-POMCP	254.9	3.31	257.5	6.12	**263.2**	**72.51**
POMCP		–		–		–

超过运算时间限制 (30 分钟) 的实验记为 '–'。

表 3.5　不同仿真次数的 24 个智能体的运算结果

	50		100		1000	
	值	时间 (s)	值	时间 (s)	值	时间 (s)
TD-FMOP	**807.8**	**40.15**	826.7	78.67	843.3	1371.7
FMOP		–		–		–
TD-POMCP	522.5	7.94	524.3	16.08	519.8	172.9
POMCP		–		–		–

超过运算时间限制 (30 分钟) 的实验记为 '–'。

由于 FMOP 和 POMCP 为集中式的算法，其联合动作和联合观测过于巨大，以至于两种算法在 30 分钟内无法求得一个解。相反地，TD-FMOP 和 TD-POMCP 的运算时间远远小于 FMOP 和 POMCP，且结果显得比较合理，这是因为对于联合动

作的选择采用了分布式的近似计算方式（即 max-sum）。然而，对于大的状态空间且状态随机频繁变化的情况，TD-POMCP 的粒子滤波器变得不再有效。从而，TD-POMCP 的计算结果几乎与随机算法的结果相当。由于其精确的信念更新方式，TD-FMOP 超过 TD-POMCP 达 58.65%。

因此，论文通过比较算法 POMCP、FMOP 和 FMOP*，对特征信念和启发式预算的性能进行了评价，并验证了 TD-FMOP 具有良好的可扩展性，即可有效求解大规模问题（即 24 个智能体的侦察监视问题）。

3.8 本章小结

本章首先提出了新的模型用来描述在不确定和健康约束下的部分可观和非静态环境中的分散式侦察监视问题。由于其问题复杂度过高，因此当前已有的算法不能有效进行求解。所以设计了在线规划算法，提供能够扩展到大量智能体问题的分布式解。特别地，论文提出的算法组合了健康约束、特征信念表示、蒙特卡洛树搜索和 max-sum 进行在线规划。最后对本章的方法进行仿真验证，通过小量智能体问题的实验验证其有效性，并验证其可以扩展到有效求解具有稀疏交互性质的大量智能体的问题。

在实际应用中，采用无人机对环境进行连续的侦察监视是国防任务中非常重要的部分。论文中模型的设计，主要来源于国防专家根据在威胁环境中部署和使用无人机的经验与需求。这些模型也涉及到在危险环境中部署人来执行一些危险任务的情形。对于派出贵重的资源（或人）于危险环境执行任务所面临的困难，论文的工作同样提供了重要的理论支撑。

鉴于本章方法针对的是非常通用的模型，这种方法同样能够有效应用到描述其他领域问题，如可以对突发事件所在现场的其他环境物理特征进行建模。具体地，虽然论文假设已知环境中的图（由地点和可通行的路径构成），但是其中每个地点的状态并不可知，所以环境中的其他物理特征（如障碍物）可以建模为每个顶点的状态变量之一。

第四章　人辅助下基于潘多拉规则的搜索方法

4.1　引言

机器人（包括无人机和自主车等）[1] 广泛应用于执行搜索任务，特别是部署于一些难以到达的极端环境中。同时，这些环境的自然属性常常具有不确定性，使得人很难直接进入环境中执行任务 [113]。例如，一辆探测车在某个星球地表进行搜索并挑选出一块某类的矿石，而矿石的品质在挖出之前是不确定的。又如，一辆无人机在灾难环境中搜寻某架解体飞机的某块重要部分的碎片。在这些想定中，机器人需要付出一定的代价（如挖开一块矿石消耗的能量 / 时间，或访问某个位置的风险）来获知某个选择的准确回报值（如某块矿石的质量，或能否在某个位置找到期望的那块碎片），而且机器人在有些时候并不足够高效（需要花费较多时间挖开某块矿石，或冒较大风险访问某个位置）。特别地，人可以辅助机器人来更加有效地减小这些回报值的不确定性 [80]（如通过提供关于环境中的一些信息的快照和众包报告等），以进一步提高机器人执行任务的效率。另一方面，人并不能总有空，而且寻求人的辅助同样可能耗费一定的代价 [79, 114]。

基于这两种不确定性（环境和人的辅助的可利用性），这类机器人搜索问题的挑战表现在两个方面。首先，在每个时间步，机器人需要决定是否继续进行搜索来获取关于环境更多的知识，或者停止搜索来选择某个已知选项以完成任务。然后，对于在某时刻对某个未知选项回报值的探索，机器人需要决定是否寻求人的辅助或者自己来进行探索。因此，对于最优决策制定，机器人需要平衡所有可能行为的值。

传统的人与机器人的研究领域主要考虑人与机器人如何交互的体系和机制 [83, 115–117]。近些年，当前先进的规划方法（包括确定性和不确定性条件下），成功应用于人与机器人构成团队的决策制定当中 [80, 118–120]。然而，这些通用的规划算法很难解决本章的这种每个动作都可能产生大量（可能为无穷多）不同值的搜索问题。

为解决这一挑战，本章关注一类搜索问题，其中访问不同位置所需代价值之间相互独立。与文献 [80, 120] 中的更通用模型相比，这种独立性的假设使得论文模型的表达能力变窄，但是仍然能够覆盖机器人搜索问题中的一大类重要应用。比如，如果一个自主探测车需要探测的物品之间距离很近，那么这种物理移动耗费的能量或时间远远小于挖开这些矿石所消耗的；又如，当无人机搜索某个坠落

[1] 由于无人机可以看做一种类型的机器人，且本章的方法不局限于无人机并可扩展至通用的机器人领域，因此，本章以机器人为对象介绍论文的方法。

直升机的碎片时，更为重要的代价为环境中威胁，而且不同位置之间的这种代价或威胁可以看作是相互独立的。

基于以上背景，本章提出一种新的模型，机器人和人的搜索问题（Robot-Human Search, RHS），并提出一种新颖的最优算法来求解这类问题。具体地，与其他搜索或规划方法相似 [80, 89, 120, 121]，论文的机器人决策制定问题也可以形式化描述为一个动态规划（Dynamic Programming）问题。然而，本文并不采用那些典型的近似算法进行求解，而是设计了一个基于指标的算法。更具体地，论文对每个可能的动作定义了一个指标，并设计一个搜索规则来总是选择具有最大指标值的动作来执行。特别地，本文证明基于这种代价的独立性假设，这种基于指标的搜索算法为多项式时间复杂且满足最优。与当前的研究现状相比，本章的创新包括以下几个方面：1）首次提出一种新的形式化模型，描述一个机器人在不确定知识和人辅助下的搜索问题；这一模型不仅考虑在搜索中机器人主动寻求人的辅助，而且考虑到物品回报值和人的可利用的不确定性。2）设计一个多项式时间算法来求解这种搜索问题，并通过理论其最优性；与文献中 [80, 122] 更为通用的形式化模型相比，本章的方法能够显著提高计算效率。3）通过仿真实验表明，本文的搜索方法显著优于一些相关的测试集方法。

本章余下各节安排如下：首先对问题进行描述并建立基于动态规划的形式化表达。然后设计搜索算法，并证明其多项式时间复杂度和最优性。最后通过仿真实验对本章的方法进行进一步的验证。

4.2 问题建模

本节将首先对 RHS 问题进行建模，然后给出一个例子来解释这类问题。

4.2.1 RHS 模型

考虑机器人在环境中指定的 n 个位置之间进行搜索。对于每个位置 i（$1 \leq i \leq n$）均放置有一个物品。单个物品的可能回报值 x_i 服从概率分布函数 $F_i(x_i)$，且不同物品的回报值之间是相互独立的。每个物品的回报值（事先）为不确定的，而机器人能够通过自己的工具（通过一些物理方式）去揭示，或者可以请求人辅助来检查这一回报值。揭示代价表示为 c_i^{reveal}，而寻求人的辅助需要机器人同样扣除一个代价 c_i^{ask}。论文定义 p 表示机器人寻求辅助时该人能够提供所期望信息的概率。机器人在人的辅助下不断地在这些物品之间进行探索，然后选择某个已知的物品来进行收集。因此，机器人的目标是最大化所获物品的回报值，同时最小化搜索代价的和。

下面将具体介绍机器人、人与环境之间的交互关系（如图4.1（a）所示）。

首先，定义物品的可能状态（如图4.1（b）所示）。单个物品的所有状态表示

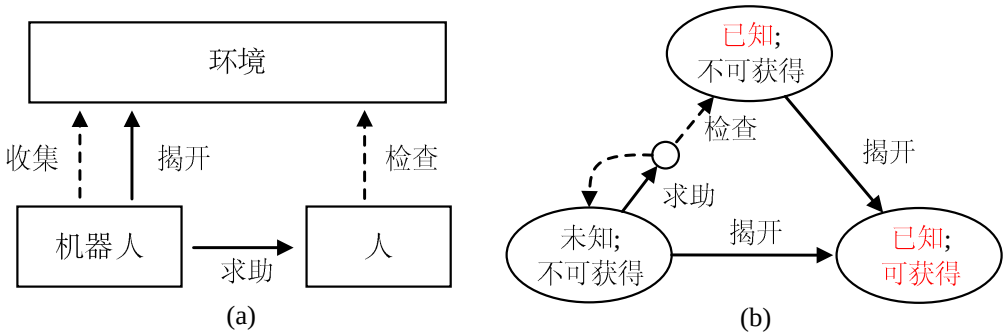


图 4.1 （a）机器人、人与环境之间的交互关系（b）物品的可能状态之间的转移关系

为“未知, 不可获得”，“已知, 不可获得”，和“已知, 可获得”，其中已知 / 未知表示机器人知道 / 不知道该物品的回报值，而可获得 / 不可获得表示机器人是否已经可以收集该物品了。初始时物品的状态为“未知, 不可获得”，即其回报值未知且仍然不能被收集带走，除非移除掉一定的物理障碍[2]。

机器人的动作包括：揭开某个未知物品和收集带走某个已知物品。具体地，机器人需要耗费代价 c_i^{reveal} 揭开某个物品来使得其已知并可获得，即“未知, 不可获得”$\rightarrow$“已知, 可获得”。例如，机器人需要消耗一些能量 / 时间来掀开表面的土壤以使得一个潜在的矿石并变得可获得。更进一步，机器人可以在已知物品的集合中选择一件收集带走[3]，这也意味着机器人的搜索任务完成。

人能够向机器人提供关于某个物品回报值的一些信息。实际当中，人并非永远可利用或者可被打扰 [114, 123]。当人不在时或者无法提供机器人所需的信息，人必然无法辅助机器人。如果某个人正忙于辅助其他机器人时，人同样无法进行响应。进一步，机器人需要付出代价 c^{ask} 来寻求人的辅助。比如，从月球传输信号至地球需要 2.7 秒的时间，从火星传输信号至地球需要 14 分钟的时间。如果当前人可被利用，她将检查物品的回报值，即“未知, 不可获得”$\rightarrow$“已知, 不可获得”；否则该物品的状态保持不变。此外，对于已经检查过的位于位置 i 的物品，机器人仍然需要付出代价 c_i^{reveal} 来使其可获得，即“已知, 不可获得”$\rightarrow$“已知, 可获得”。

基于此，下面将提出 RHS 的形式化描述。具体地，定义三个二元决策变量的集合：$q_i^{\text{c}}, \forall i \in I$，当检查物品 i 为解的一部分时 $q_i^{\text{c}} = 1$，否则 $q_i^{\text{c}} = 0$；$q_i^{\text{r}}, \forall i \in I$，当揭开物品 i 为解的一部分时 $q_i^{\text{r}} = 1$，否则 $q_i^{\text{r}} = 0$；及 $l_i, \forall i \in I$，当收集带走物品 i 时 $l_i = 1$，否则 $l_i = 0$。RHS 的形式化描述如下：

[2]补充说明，论文中物品的“可获得”是指该物品可随时被机器人收集带走，而人的“可利用”是指在某个时刻人可以向机器人提供辅助。

[3] 为了简化，论文省略了人对某个进行检查所需的代价。然而，论文的方法可以很容易扩展到求解具有这类代价的问题。

RHS:

$$\text{Max} \quad \mathbb{E}\left[\sum_{i\in I}\left(-n_i^{\text{a}}c_i^{\text{ask}} - q_i^{\text{r}}c_i^{\text{reveal}} + l_i x_i - l_i q_i^{\text{c}} c_i^{\text{reveal}}\right)\right] \tag{4.1}$$

$$\text{S. t.}$$

$$q_i^{\text{c}} + q_i^{\text{r}} \leq 1, \forall i \in I, \tag{4.2}$$

$$q_i^{\text{c}} + q_i^{\text{r}} \geq l_i, \forall i \in I, \tag{4.3}$$

$$\sum_{i\in I} l_i = 1, \tag{4.4}$$

$$q_i^{\text{c}}, q_i^{\text{r}}, l_i \in \{0,1\}, \forall i \in I. \tag{4.5}$$

其中 $n_i^{\text{a}} \in \mathbb{N}, \forall i \in I$ 为机器人尝试求助人检查物品 i 的次数。

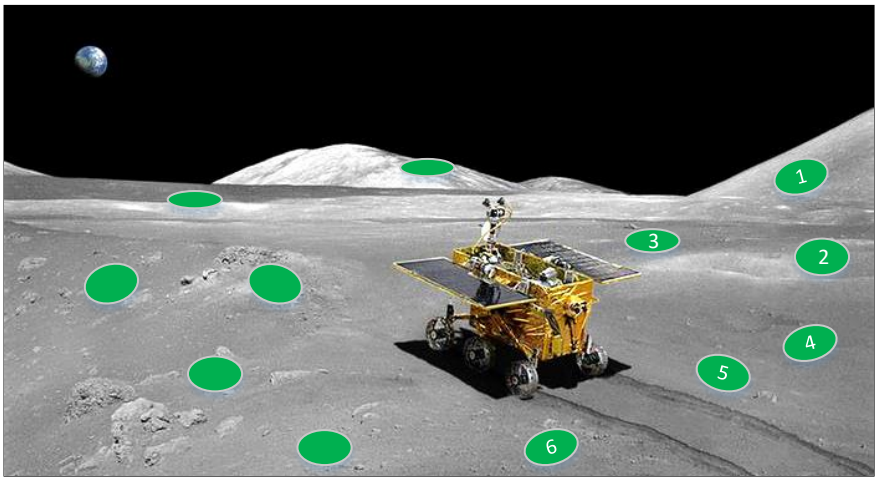
目标函数 (4.1) 为最大化所获得物品回报值减去搜索代价之和的期望值。约束 (4.2) 保证未知物品或者被检查或者被揭开或者什么都不操作。约束 (4.3) 指出一个物品在收集带走之前需要被揭开或检查。约束 (4.4) 保证只有一个物品被收集带走。约束 (4.5) 强制使所有决策变量为二元变量。

4.2.2 想定举例

这里给出一个简单的想定例子来解释机器人如何在其任务想定中运行的。如图4.2所示，一个自主车在月球表面搜索某块矿石以带回地球，地球上有一名工作人员可以向机器人提供一些可能的辅助，其中绿色圆盘表示指定的待自主车进行搜索的位置。基于一些先验知识确定了一些潜在的挖掘位置，而每个位置中可能的矿石均具有一些基于其大小和材质的效能值的不确定性。为了评估某个位置 i 的矿石的效能值 x_i，机器人可以花费代价 c^{ask} 向人寻求辅助或者花费代价 c_i^{reveal} 亲自挖开该位置覆盖的土壤来测量矿石。这些不同位置矿石的参数列出如图4.2所示。通过对其进行分析，能够发现对于不同矿石的不同操作，需要不同对待。比如，能够推理得出如下一些结论：

- 矿石 1 和矿石 2 具有相同的期望效能值，当机器人揭开 x_1 并发现其为 0.9 时（高于 x_2 的上确界），将没有必要再去访问矿石 2 了。然而，这对于先揭开矿石 2 的选择并不成立。因此，即使两个物品拥有相等的期望效能值，采用不同的揭开顺序将可能带来不同的结果。

- 与矿石 2 相比，矿石 3 具有更高的期望效能值和更高的揭开代价。如果机器人求助于人来检查 x_3 并发现其为 0.6，机器人将放弃矿石 3 并继续进行搜



$c^{\text{check}} = 0.02$

$x_1 \in [0.1, 0.9],\quad c_1^{\text{reveal}} = 0.10$
$x_2 \in [0.4, 0.6],\quad c_2^{\text{reveal}} = 0.10$
$x_3 \in [0.3, 1.0],\quad c_3^{\text{reveal}} = 0.20$
$x_4 \in [0.2, 0.4],\quad c_4^{\text{reveal}} = 0.01$

…

图 4.2　自主车探索矿石想定

索。（因为这将会造成 $x_3 - c_3^{\text{reveal}} \leq x_2 - c_2^{\text{reveal}}$，即 $\forall x_2 \in [0.4, 0.6], 0.6 - 0.2 \leq x_2 - 0.1$）。从而，将可能避免揭开矿石 3 所需要耗费的代价 0.2。因此，在揭开某个矿石之前先对它进行检查将可能降低整个搜索耗费的搜索代价。

- 若某个矿石的揭开代价小于 $c^{\text{ask}} = 0.02$，比如矿石 4 满足 $c_4^{\text{reveal}} = 0.01$，揭开该矿石的动作将优先于求助于人来检查该效能值。因此，对于小的揭开代价的矿石，将没有必要请求人来进行辅助检查。

从而，机器人的回报值将同时与矿石的效能值以及揭开和求助动作的代价相关。因此，同时考虑搜索代价和矿石的效能值，机器人的策略需要最大化整个搜索过程所能导致的最终利益收获，定义为最终选择的矿石的效用值减去搜索过程中累积的代价，而不仅仅是为了找到具有最大效用值的矿石。

以上定义了 RHS 的问题模型，现在需要一个方法，来基于物品的当前知识的状态来规划搜索动作。因此，接下来，将提出一种动态规划的形式化方法用于描述机器人的决策并设计算法来进行求解。

4.2.3 基于动态规划的形式化描述

基于论文提出的问题模型，本节将 RHS 问题中机器人的决策问题形式化为一个动态规划问题，并定义其最优解。动态规划的形式可以用来描述很多规划和决策问题，如文献 [80, 89, 120, 121] 中同样采用动态规划对各种搜索问题进行的形式化建模。然而，不同动态规划问题的求解难度差异可能很大。下面将具体介绍如何将论文的 RHS 问题描述为动态规划问题以及其最优解。并在后面介绍并验证其求解算法。

将 n 个物品的集合表示为 $I=\{1,2,\ldots,n\}$，并将其分为两个集合：一个为不断增长的已知物品的集合 $S\subseteq I$；另一个为其补集，即未知物品的集合 $\bar{S}$。

每个时刻，机器人可以选择是否从集合 $\bar{S}$ 中选择一个未知物品求助于人来检查，或者自主地从 $\bar{S}$ 中选择并揭开一个物品，或者停止搜索并从已知物品 S 中选择一个进行收集带走。

首先，将收集带走某个已知物品 $i\in S$ 的收集回报值表示如下：

$$r_i=\begin{cases} x_i & \text{如果该物品被机器人揭开过,} \\ -c_i^{\text{reveal}}+x_i & \text{如果如果该物品被人检查过.} \end{cases} \tag{4.6}$$

当机器人决定尝试收集带走某个物品时，其最优策略为：直接选择具有最大的当前已知的收集回报值：

$$y=\max_{i\in S} r_i. \tag{4.7}$$

将任意时刻的状态表示为 $(\bar{S},y)$，并定义 $\Psi(\bar{S},y)$ 表示：当未知位置集合为 $\bar{S}$ 且最大已知回报为 y 时，从这一时刻起按照最优策略能够获得的最终的期望值。值得注意的是，论文并不明确将人的可利用性在定义的状态中进行表达。这是因为对于每个时刻，在执行求助之前，机器人并不能知道人是否能够提供辅助，即需要花费代价 c^{ask} 来尝试与人取得联系。相反，论文稍后将在各个状态之间的递归关系中考虑人的辅助的不确定性。

对于每个 $\bar{S}$ 和 y，值函数需要满足基本的递归关系：

$$\Psi(\bar{S},y)=\max\left\{y,\Psi^{\text{reveal}}(\bar{S},y),\Psi^{\text{ask}}(\bar{S},y)\right\},$$

其中

$$\Psi^{\text{reveal}}(\bar{S},y)=\max_{i\in\bar{S}}\left\{-c_i^{\text{reveal}}+\Psi(\bar{S}-\{i\},y)\int_{-\infty}^{y}\mathrm{d}F_i(x)+\int_{y}^{\infty}\Psi(\bar{S}-\{i\},x)\mathrm{d}F_i(x)\right\},$$

和

$$\begin{aligned}\Psi^{\text{ask}}(\bar{S},y) &= p\hat{\Psi}^{\text{ask}}(\bar{S},y)+(1-p)\left(\Psi^{\text{ask}}(\bar{S},y)-c^{\text{ask}}\right)\\ &= \hat{\Psi}^{\text{ask}}(\bar{S},y)-\frac{(1-p)c^{\text{ask}}}{p}\\ &= \max_{i\in\bar{S}}\left\{-c^{\text{ask}}+\Psi(\bar{S}-\{i\},y)\int_{-\infty}^{y}\mathrm{d}F_i^{\text{ask}}(x)\right.\\ &\left.+\int_{y}^{\infty}\Psi(\bar{S}-\{i\},x)\mathrm{d}F_i^{\text{ask}}(x)\right\}-\frac{(1-p)c^{\text{ask}}}{p}\\ &= \max_{i\in\bar{S}}\left\{-\frac{c^{\text{ask}}}{p}+\Psi(\bar{S}-\{i\},y)\int_{-\infty}^{y}\mathrm{d}F_i^{\text{ask}}(x)\right.\\ &\left.+\int_{y}^{\infty}\Psi(\bar{S}-\{i\},x)\mathrm{d}F_i^{\text{ask}}(x)\right\},\end{aligned}$$

其中 $\Psi^{\text{reveal}}(\bar{S},y)$ 和 $\Psi^{\text{ask}}(\bar{S},y)$ 分别为动作揭开和求助的值，且 $F_i^{\text{ask}}(x)=F_i(x+c_i^{\text{reveal}})$ 为 $x=x_i-c_i^{\text{reveal}}$ 的累积概率分布函数，即代表着机器人如果收集带走位置 i 处的物品将能获得的回报值。更具体地，对于每个状态 $(\bar{S},y)$，机器人需要比较不同动作可能引起的值。对于揭示某个物品这个动作的值，需要考虑如下两类可能出现的结果，具体如下：

- 如果回报值 $x\le y$，最高的当前已知回报值将不变，且期望效用值为 $-c_i^{\text{reveal}}+\Psi(\bar{S}-\{i\},y)$；
- 否则，最高的当前已知回报值将更新为 x 且期望效用值为 $-c_i^{\text{reveal}}+\Psi(\bar{S}-\{i\},x)$。

当向人求助来检查某个物品时，将考虑人的可用性可能出现的两种不同状况，具体如下：

- 如果无法获得辅助（概率为 $1-p$），状态将不会发生变化且期望效用值为 $-c_i^{\text{ask}}+\Psi(\bar{S},y)$；
- 否则，当人能够提供辅助时的值表示为 $\hat{\Psi}^{\text{ask}}(\bar{S},y)$，其递归关系与上面对 $\Psi^{\text{reveal}}(\bar{S},y)$ 的分析类似。

从而，根据以上对每类动作（包括机器人收集带走某个物品、揭开某个物品，以及向人求助检查某个物品）的不同结果对应的值递归关系的具体分析，可以将这种机器人在不确定知识和人的辅助下的搜索问题（RHS）总结如下：

RHS 的最优解:

对于 RHS 的当前状态 $(\bar{S}, y)$，能够最大化值 $\Psi(\bar{S}, y)$ 的一个最优解可以计算如下：

$$\Psi(\bar{S}, y) = \max\left\{y, \Psi^{\text{reveal}}(\bar{S}, y), \Psi^{\text{ask}}(\bar{S}, y)\right\}, \tag{4.8}$$

其中

$$\begin{aligned}\Psi^{\text{reveal}}(\bar{S}, y) = \max_{i\in\bar{S}}\Bigg\{&-c_i^{\text{reveal}} + \Psi(\bar{S}-\{i\}, y)\int_{-\infty}^{y} \mathrm{d}F_i(x)\\ &+\int_{y}^{\infty}\Psi(\bar{S}-\{i\}, x)\mathrm{d}F_i(x)\Bigg\},\\ \Psi^{\text{ask}}(\bar{S}, y) = \max_{i\in\bar{S}}\Bigg\{&-\frac{c^{\text{ask}}}{p} + \Psi(\bar{S}-\{i\}, y)\int_{-\infty}^{y} \mathrm{d}F_i^{\text{ask}}(x)\\ &+\int_{y}^{\infty}\Psi(\bar{S}-\{i\}, x)\mathrm{d}F_i^{\text{ask}}(x)\Bigg\}.\end{aligned}$$

因此，以上将机器人的决策问题形式化描述成了一个动态规划问题。然而，这一形式当中，递归的值函数将无法直接进行求解计算。具体地，这一动态规划问题的计算时间和存储需求与传统的 n 个访问节点的旅行商问题相同，即复杂度为 $O(n^2 2^n)$ [124]，这对于大的 n 的问题将无法计算求解。因此，论文在下一节将设计一种基于指标的策略，能够在多项式时间内对问题进行最优求解。

4.3 基于潘多拉规则的求解算法

4.3.1 搜索策略

基于动态规划的形式化描述，本节首先提出一个多项式时间的搜索策略，一种通过定义所有可能动作的指标的基于指标的策略。然后证明这种策略的最优性。最后进一步分析这一最优策略的相关性质。

受经典潘多拉问题中潘多拉规则的启发，定义每个揭开和求助动作的指标。具体地，对于每个位置 i，定义两个指标，分别针对每个揭开和求助动作，表示为揭开指标 z_i^{reveal} 和求助指标 z_i^{ask}，分别可计算得出：

$$\begin{aligned}z_i^{\text{reveal}} &= -c_i^{\text{reveal}} + z_i^{\text{reveal}}\int_{-\infty}^{z_i^{\text{reveal}}}\mathrm{d}F_i(x) + \int_{z_i^{\text{reveal}}}^{\infty} x\mathrm{d}F_i(x),\\ z_i^{\text{ask}} &= -\frac{c^{\text{ask}}}{p} + z_i^{\text{ask}}\int_{-\infty}^{z_i^{\text{ask}}}\mathrm{d}F_i^{\text{ask}}(x) + \int_{z_i^{\text{ask}}}^{\infty} x\mathrm{d}F_i^{\text{ask}}(x),\end{aligned}$$

即：

$$c_i^{\text{reveal}} = \int_{z_i^{\text{reveal}}}^{\infty} (x - z_i^{\text{reveal}}) \mathrm{d}F_i(x), \tag{4.9}$$

$$\frac{c^{\text{ask}}}{p} = \int_{z_i^{\text{ask}}}^{\infty} (x - z_i^{\text{ask}}) \mathrm{d}F_i^{\text{ask}}(x). \tag{4.10}$$

基于状态 $(\bar{S}, y)$ 和指标集合 $\{z_i^{\text{reveal}}, z_i^{\text{ask}} | i \in \bar{S}\}$，可以设计一种简单（但是最优）策略，称为搜索规则，如下：

搜索规则:

求助 / 揭开规则：如果将要向人求助来检查（或机器人揭开）某个位置的物品，那么应该选择具有最大求助指标（或揭开指标）的未知物品。关于是选择向人求助还是机器人自己揭开，由哪个最大的指标更大来决定；

收集规则：当已知的最大收集回报值同时大于所有未知物品的求助指标和揭开指标时，停止搜索并选择具有最大收集回报的物品进行收集带走。

4.3.2 搜索算法

接下来，将设计执行搜索策略的算法（见算法4.1）。首先，计算已知物品的最高收集回报值（line 2-3）并计算所有未知物品的揭开指标和求助指标（line 4-7）。然后，比较这些指标（line 8）并选出一个物品进行收集（line 9-10）、揭开（line 11-12）或求助（line 13-14）。当寻求辅助的时候，如果人当前不可利用，那么重复执行求助的动作也是没有意义的，这是因为人的是否可利用的状态通常不会立刻改变。因此，若人在当前时刻不可利用，机器人将选择一个已知的物品收集带走，或者选择一个未知的物品揭开（line 21-27）。

基于此，针对论文设计的 RHS 问题，设计了基于指标的搜索算法，下面将对算法的一些性能进行分析。

4.4 算法性能分析

基于以上设计的搜索策略，接下来将分析其性能，包括复杂度和最优性，以及其他的一些性质。

4.4.1 复杂度和最优性证明

本节首先证明算法的多项式时间计算复杂定，然后证明其最优性。

算法 4.1 基于潘多拉规则的 RHS 问题求解算法

procedure SEARCH($S, \bar{S}$)
▷ 找到已知物品中的最大收集指标：
    $y = \max_{i \in S} r_i$
    $\hat{i} = \arg\max_{i \in S} r_i$
▷ 计算揭开指标和求助指标：
    for $i \in \bar{S}$ **do**
        $z_i^{\text{reveal}} \leftarrow$ Solve $c_i^{\text{reveal}} = \int_{z_i^{\text{reveal}}}^{\infty} (x - z_i^{\text{reveal}}) \mathrm{d}F_i(x)$
        $z_i^{\text{ask}} \leftarrow$ Solve $\frac{c^{\text{ask}}}{p} = \int_{z_i^{\text{ask}}}^{\infty} (x - z_i^{\text{ask}}) \mathrm{d}F_i^{\text{ask}}(x)$
    end for
▷ 比较所有的指标并执行最优动作：
    $i^* = \arg\max_{i \in S} \max\{z_i^{\text{reveal}}, z_i^{\text{ask}}\}$
    if $y \geq \max\{z_{i^*}^{\text{reveal}}, z_{i^*}^{\text{ask}}\}$ **then**
        return COLLECT($S, \bar{S}, \hat{i}$)
    else if $z_{i^*}^{\text{reveal}} \geq z_{i^*}^{\text{ask}}$ **then**
        return REVEAL($S, \bar{S}, i^*$)
    else
        return ASK($S, \bar{S}, i^*$)
    end if
end procedure

procedure ASK($S, \bar{S}, i$)
    if $humanAvailable = True$ **then**
        return CHECK($S, \bar{S}, i$)
    else
        $i' = \arg\max_{i \in S} \max\{z_i^{\text{reveal}}\}$
        if $\hat{y} \geq z_{i'}^{\text{reveal}}$ **then**
            return COLLECT($S, \bar{S}, \hat{i}$)
        else
            return REVEAL($S, \bar{S}, i'$)
        end if
    end if
end procedure

定理 4.1: 搜索规则的复杂度为 $O(n \log n)$。

证明：在算法4.1中，当机器人基于所有物品指标值的顺序来选择执行的动作，而且这个顺序在搜索过程中并不会发生变化，论文搜索策略的复杂度只依赖于计算这些指标的顺序，从而复杂度为 $O(n \log n)$。

定理 4.2: 搜索规则为 RHS 的最优策略。

证明：为了证明搜索规则为 RHS 的最优策略，论文解释如何将 RHS 映射成为潘多拉问题 [125]，一种关于打开盒子的经济学搜索模型。潘多拉问题中，每个关闭的盒子装有服从一个概率分布函数的潜在回报值，而且每个盒子都对应一个代价来衡量打开并学习其内容所耗费的代价。首先，集合 S 中每个已知物品和其位置，可以看作为一个含有回报值 r_i 的打开的盒子。然后，如图4.3所示，认为每个未知的位置包含两个盒子 i^{reveal} 和 i^{ask}: i^{reveal} 含有概率分布为 $F_i(x_i)$ 的潜在回报值 x_i 且其打开代价为 c_i^{reveal}；i^{ask} 含有概率分布为 $F_i(x_i)$ 的潜在回报值 x_i 且其打开代价为 $\frac{c^{\text{ask}}}{p}$。一旦盒子打开，将其从关闭盒子的集合 $\bar{S}$ 中移到打开的盒子的集合 S 中，并将同位置的另外一个盒子删除。基于指标的策略如下：1）如果要打开某个盒子，那么打开具有最高指标的盒子。2）当最大的采样回报值超过每个关闭盒子的指标，停止搜索。文献 [125] 证明了这类策略能够得到最优期望回报值。因此，RHS 便映射成为了一个盒子集合为 $\{i^{\text{reveal}}, i^{\text{ask}} \mid i \in I\}$ 的潘多拉问题，且论文的搜索策略能够得到最优期望回报值（方程（4.8））。

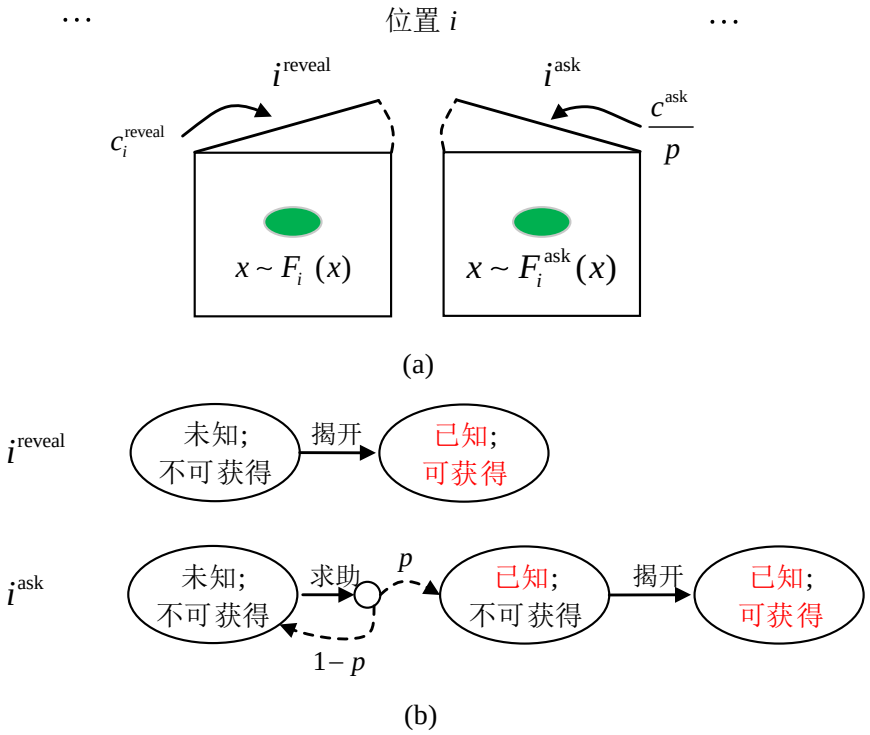


图 4.3 （a）将揭开或求助动作映射为潘多拉问题中打开一个盒子（b）物品各个状态之间的转换关系

4.4.2 其他性质分析

基于论文设计的搜索规则和算法4.1，能够得到一些关于最优解的理想的性质。首先，如果人具有相对高的可利用性 p，那么向人求助的动作将具有相对高的指标。

性质 4.1: 对任何未知的物品 $i \in \bar{S}$，其求助指标 z_i^{ask} 随人的可利用性 p 增长，且其揭开指标 z_i^{reveal} 与 p 独立。

然后，能够得到揭开指标和求助指标满足如下的性质：

性质 4.2: 对于任何未知的物品 $i \in \bar{S}$，如果 $c_i^{\text{reveal}} < \frac{c^{\text{ask}}}{p}$，那么 $z_i^{\text{reveal}} > z_i^{\text{ask}}$，即求助人去检查 i 的动作被揭开 i 的动作占优[4]。

证明：通过反证法来证明这个性质，即：若 $z_i^{\text{reveal}} \leq z_i^{\text{ask}}$，则 $c_i^{\text{reveal}} \geq \frac{c^{\text{ask}}}{p}$。具体地，首先注意到，对于任何服从概率分布 $F(x)$ 的潜在回报值为 x 的物品，函数 $\int_z^{\infty}(x-z)\mathrm{d}F(x)$ 随 z 递减（因为其导数为 $-(x-z)\dot{F}(x) \leq 0$，其中 $x \in \{z, \infty\}$）。接下来，若 $z_i^{\text{reveal}} \leq z_i^{\text{ask}}$，则 $z_i^{\text{reveal}} \leq z_i^{\text{ask}} + c_i^{\text{reveal}}$。然后满足：

$$
\begin{aligned}
&\int_{z_i^{\text{reveal}}}^{\infty}(x - z_i^{\text{reveal}})\mathrm{d}F_i(x) - \int_{z_i^{\text{ask}}}^{\infty}(x - z_i^{\text{ask}})\mathrm{d}F_i(x + c_i^{\text{reveal}}) \\
=&\int_{z_i^{\text{reveal}}}^{\infty}(x - z_i^{\text{reveal}})\mathrm{d}F_i(x) - \int_{z_i^{\text{ask}}+c_i^{\text{reveal}}}^{\infty}\left(x - (z_i^{\text{ask}} + c_i^{\text{reveal}})\right)\mathrm{d}F_i(x) \\
\geq& 0.
\end{aligned}
$$

基于此，由方程（4.9）和（4.10）可得 $c_i^{\text{reveal}} \geq \frac{c^{\text{ask}}}{p}$。因此，可以得出：如果 $c_i^{\text{reveal}} < \frac{c^{\text{ask}}}{p}$，那么 $z_i^{\text{reveal}} > z_i^{\text{ask}}$。

第三，如果求助动作一直不需要代价且人永远可利用，那么一个最优解为检查所有未知的物品然后选择最好的收集带走。类似地，如果揭开某个物品的动作不需要代价，那么一个最优解可能包含揭开那个物品的动作。

性质 4.3: 如果 $c^{\text{ask}} = 0$，那么一个最优解为检查所有的未知物品并选择最好的收集带走。对于任何未知的物品 $i \in \bar{S}$，如果 $c_i^{\text{reveal}} = 0$，那么一个最优解包含揭开物品 i 的动作。

[4]值得注意的是，性质4.2的逆命题（即对任何未知的物品 $i \in \bar{S}$，如果 $c_i^{\text{reveal}} > \frac{c^{\text{ask}}}{p}$ 那么 $z_i^{\text{reveal}} < z_i^{\text{ask}}$）不成立。这意味着有时 $c_i^{\text{reveal}} > \frac{c^{\text{ask}}}{p}$ 且 $z_i^{\text{reveal}} > z_i^{\text{ask}}$，即对于一个物品，如果其揭开代价高于 $\frac{c^{\text{ask}}}{p}$，那么揭开这个物品也有可能优于求助人来检查它。

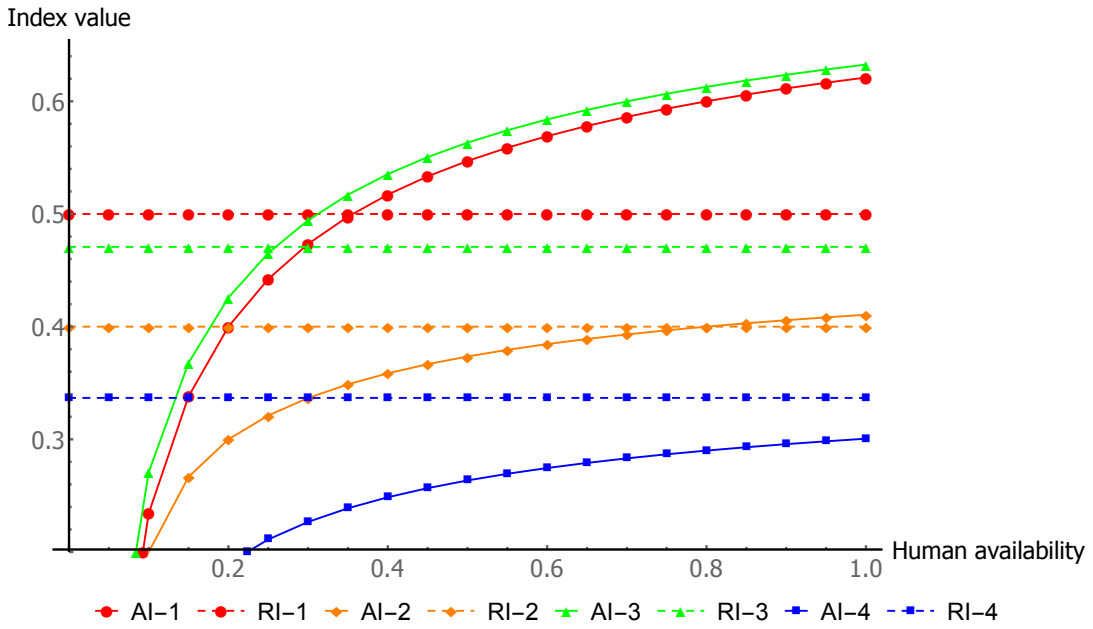


图 4.4　求助 / 揭开指标和人的可利用性关系实例

下面，基于4.2.2节中给出的例子中物品的参数，通过计算每个物品揭开指标和求助指标在不同人的可用性下的值，来进一步分析性质 4.1和性质 4.2。图4.4给出计算出的其中 4 个物品的在人的不同可用性下的求助指标和揭开指标，其中 AI-i 和 RI-i 分别表示第 i 个物品的求助指标和揭开指标。对于低的可利用性，所有 4 个物品的求助指标均低于其揭开指标。然后求助指标随人的可利用性快速增长（性质 4.1）并逐渐收敛到一定的值，其中物品 1 和物品 3 的求助指标变得大于其揭开指标。此外，由图4.4可以看出，对于任何人的可利用性 p 满足 $c_i^{\text{reveal}} < \frac{c^{\text{ask}}}{p}$（例如对于 $p < 0.2$，$c_1^{\text{reveal}} < \frac{c^{\text{ask}}}{p}$），可以得到 $z_i^{\text{reveal}} > z_i^{\text{ask}}$，其中 $i \in \{1,2,3,4\}$（性质 4.2）。

4.5　仿真实验验证

本节进行仿真实验，来评价用于求解 RHS 的搜索规则（本节中称作“Optimal”）的性能。由于已经证明论文的搜索规则为多项式时间最优，那么将不需要与其他更高复杂度的相关算法 [89, 121] 的求解性能进行比较。取而代之，论文设计四个基准测试算法策略来与论文的搜索规则进行比较。具体通过运行这些算法生成的平均效用值来进行比较。

4.5.1　实验设置

为了评价论文算法的最优性以及分析机器人、人与环境之间的交互，定义如下的统计：

- 平均效用值：基于方程4.1中定义的目标函数，一次仿真的效用值为所获得物品的回报值减去累计的代价；
- n_Ask：机器人求助的平均次数；
- n_Check：机器人得到辅助的平均次数；
- n_Reveal：机器人揭开物品的平均个数；
- n_Known：已知物品的平均个数，这些已知物品由机器人揭开或者由人检查其回报值。容易得到，满足 n_Check $\leq n$_Ask 及 n_Known $= n$_Check$+n$_Reveal。

其中不同算法生成的平均效用值用于评价这些算法的性能，其他的四个统计是用于分析搜索过程中的交互。

然后分别针对不同的人的可利用性和揭开代价来设计两个实验，且这两个实验设置如下：

- **实验 A**：首先基于第4.2.2节中给出的例子，设计在四个物品中间进行搜索的想定，并按照其中的参数构建这些想定，即：回报值的概率分布设为一致分布 $U_1(0.1,0.9), U_2(0.4,0.6), U_3(0.3,1), U_4(0.2,0.4)$，揭开代价为 $c_1^{\text{reveal}}=0.1, c_2^{\text{reveal}}=0.1, c_3^{\text{reveal}}=0.2, c_4^{\text{reveal}}=0.01$，且求助代价为 $c^{\text{ask}}=0.02$。通过变化人的可利用性 p 作为实验的参数，选择范围为 0 至 1 并以 0.05 为步长递增。

- **实验 B**：然后设计更为通用的实验想定，评价在不同揭开代价时各个算法的性能。特别地，构建 $|I|=n=10$ 个物品的 400 个想定。在每个想定中，每个物品回报值的概率分布设置为 $U(a,b)$，且 $a<b$ 并由 $U(0,1)$ 中进行一致采样得到。人的可利用性的参数和求助代价设置为：$p=0.75, c^{\text{ask}}=0.02$。通过变化人的揭开代价 c_i^{reveal} 作为实验的参数，选择范围为 0 至 0.2 并以 0.02 为步长递增。

4.5.2 基准测试算法

设计如下五个基准搜索策略来与论文的最优搜索策略进行比较：

- **Random**：机器人在每个时刻随机选择一个物品揭开、检查或收集。具体地，首先，在集合 I 中随机选择一个物品 i。如果 $i\in\bar{S}$（即物品 i 的回报值未知），然后随机揭开或者向人求助来进行检查（如果当前时刻人不可利用，那么机器人自己来揭开它），然后返回并继续执行 Random；否则，机器人收集物品 i 并结束搜索过程，最终获得这个物品的回报值。

- **All:** 在收集任何物品之前，机器人对所有的物品进行检查或者揭开。对于一个未知的物品 i，如果 $c_i^{\text{reveal}} \leq c^{\text{ask}}$，则机器人将决定揭开它；否则将向人求助来对其进行检查。一旦所有的物品变为已知，机器人将选择具有最佳效用值的物品进行收集。直观上看，这个算法在低求助代价或揭开代价时能够取得好的效果，并在满足性质4.3中的条件时达到最优。

- **Highest expected value:** 与论文中采用定义指标评价未知物品类似，另一种合理的策略为挑选具有最高期望值（即 $\max_{i\in\bar{S}}\{\mathbb{E}(x_i) - c_i^{\text{reveal}}\}$）的物品来揭开。如果某个已知物品的回报值高于所有未知物品中的最高期望值，那么收集带走那个物品并结束搜索[5]。

- **Optimal without human:** 当不考虑人的辅助时，采用论文的搜索策略需要从动作空间中删除所有向人求助的动作。一方面，这个策略可用来与 “Highest expected value” 比较两者的最优性，因为这两个算法均不需要人的辅助。另一方面，通过将其和 “Optimal” 进行比较，可以用来评价在这种机器人搜索问题中，引入人与机器人交互机制的效果。

- **Upper bound:** 定义一个上界，其表示的情境为：机器人知道关于人和所有物品的信息，包括每个时刻人是否可利用，以及每个物品的真实效用值。基于此，机器人根据这些完全的信息进行决策，例如，机器人直接选择具有最高回报值 $\arg\max_{i\in I}\{x_i - c_i^{\text{reveal}}\}$ 的物品收集带走。值得注意的是，因为假设了机器人拥有了超过问题基本模型之外的信息，所以这并不是论文问题解空间的严格的上确界。

4.5.3 实验结果与讨论

实验 A 中，对每个 p 的取值，进行 1000 次仿真。不同搜索策略所获得的平均收益如图4.5所示，且图4.6给出了相应的统计参数包括 n_Ask、n_Reveal、n_Reveal 和 n_Known。这些图中的误差条表示均值周围 95% 的置信区间且非重叠的误差条表示 $\alpha = 0.05$ 的空假设。由图4.5中可见，论文的最优搜索策略（“Optimal”）显著优于其他所有的策略。具体地，“Optimal” 和 “All” 所获得的平均效用值随 p 逐渐增加并且当 $p = 1$ 时接近 “Upper Bound” 的结果，而对于较低的 p，“Optimal” 的表现明显优于 “All”。这是因为，对于较高的 p，消耗一定的代价 c^{ask} 时机器人具有较高的概率获得人的辅助（如图4.6（a）和（b）所示）。进一步，对于较高的可利用性（$p > 0.6$），“Optimal” 和 “All” 的表现好于所有其他的算法，这是由于

[5] 值得注意的是，这个算法并不考虑人的辅助，因为对于未知的物品，如果揭开它之前进行求助检查将降低其期望值。

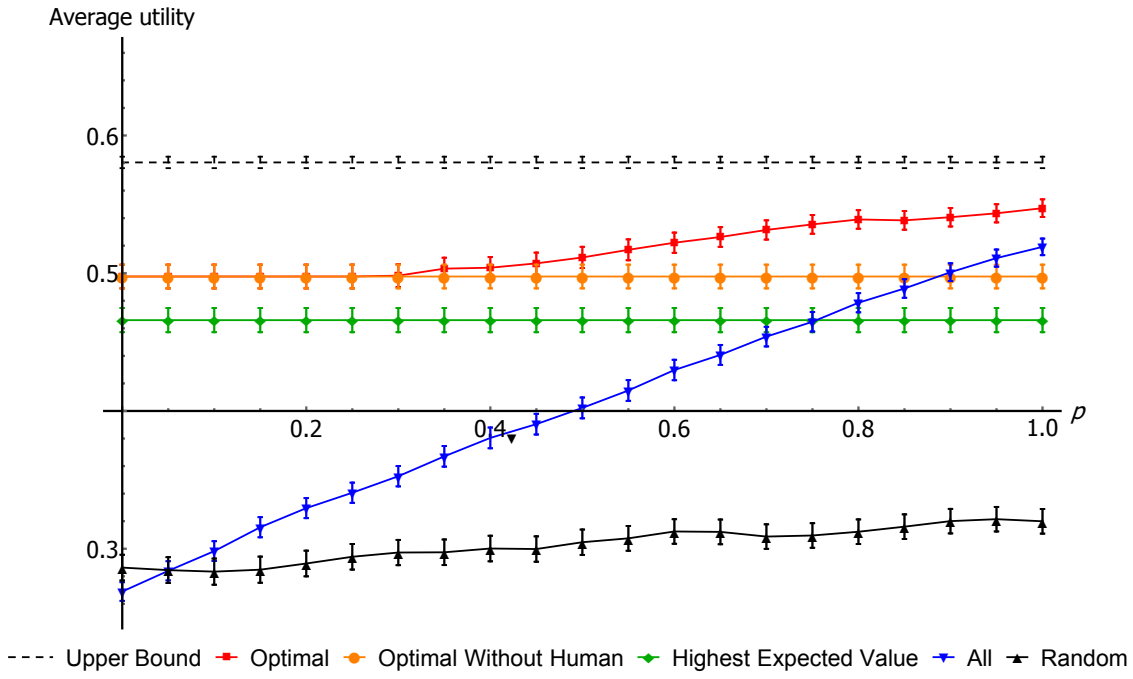


图 4.5　实验 A 中不同的人的可利用性取值下获得的平均效用值

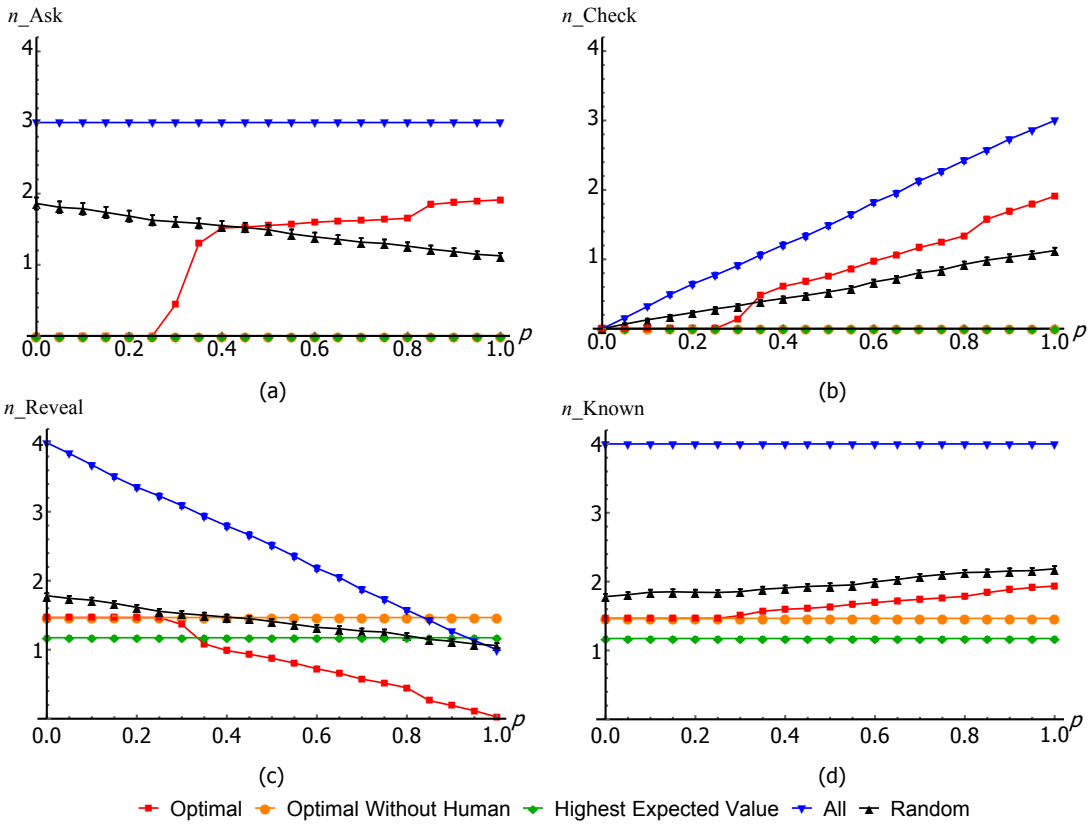


图 4.6　实验 A 中不同的人的可利用性取值下的平均求助次数、检查次数、揭开次数和已知物品次数

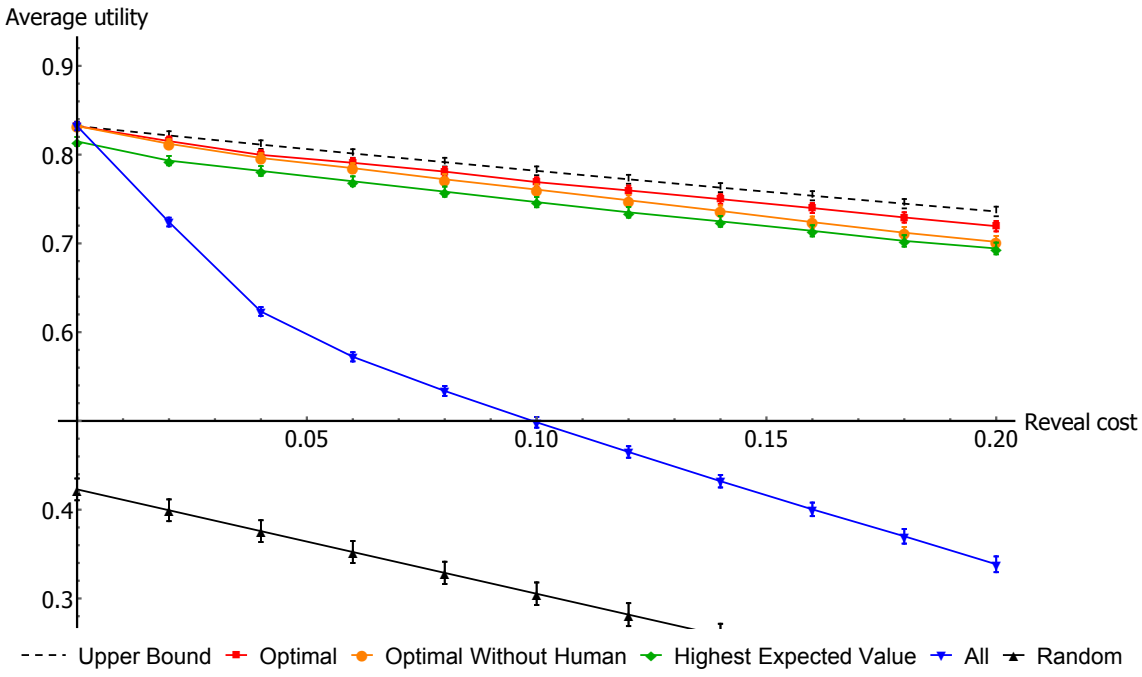


图 4.7　实验 B 中不同的揭开代价取值下获得的平均效用值

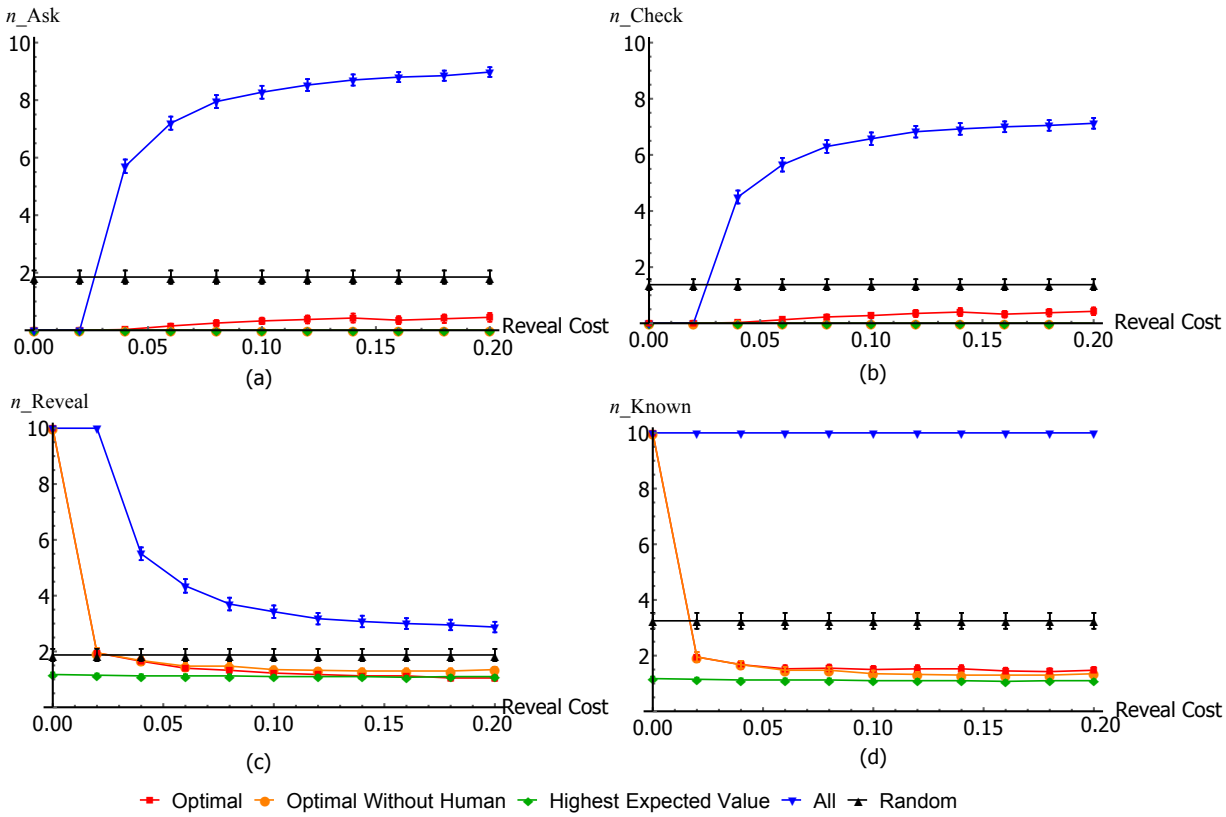


图 4.8　实验 B 中不同的揭开代价取值下的平均求助次数、检查次数、揭开次数和已知物品次数

有些物品具有高的揭开代价而这两个算法均能够求助于人来辅助检查这些物品的回报值（在图4.6（b）和（c）中可以看出，对于 “Optimal” 和 “All”，随着 p 的增加，n_Check 逐渐增加且 n_Reveal 逐渐减小）。然后，对于不考虑人的辅助的算法，“Optimal without human” 的表现优于 “Highest expected value”，这是因为论文的搜索策略考虑的是整个搜索过程的期望值，而不仅仅是当前动作的期望效用值。此外，由图4.6（d）可以看出 “Optimal” 能够在掌握了比较合适数量的物品的时候停止搜索，这个数量多余 “Optimal without human” 和 “Highest expected value” 而少于 “All” 和 “Random”，即 “Optimal” 表现优异源于能够 “有效使用资源来搜索关键的物品”。最后，“Random” 的表现远优所有其他的策略，这是由于所有的搜索动作都是有代价的且有些物品的回报值也很低，从而 “Random” 不仅在平均效用值上表现很差，而且可能随机得到的很差的结果。

实验 B 中，对每个揭开揭开代价的取值，进行 1000 次仿真。不同搜索策略所获得的平均收益如图4.7所示，且图4.8给出了相应的统计参数包括 n_Ask、n_Reveal、n_Reveal 和 n_Known。虽然所有搜索策略得到的平均效用值随揭开代价递减，论文的 “Optimal” 策略能够表现优于其他的算法且与 “Upper bound” 的结果很相近。具体地，当揭开代价为 0 时，“Optimal”、“Optimal without human” 和 “All” 的结果与 “Upper Bound” 很接近，这是因为所有这三个策略可以消耗零代价来揭开所有的物品然后选择具有最大回报值的物品收集带走（如图4.8（c）所示）。然而，当某个已知物品的收集回报值高于所有未知物品的期望值时，“Highest expected value” 将停止搜索（其中的一些仍可能拥有更好的回报值）。进一步，如图4.8（c）所示，“Highest expected value” 和 “Upper Bound” 的 n_Reveal 与揭开代价 $c_i^{\text{reveal}}, \forall i \in I$ 独立，‘Highest expected value” 和 “Upper Bound” 生成的平均效用值随揭开代价 $c_i^{\text{reveal}}, \forall i \in I$ 线性递减（如图4.7），这一结论容易由其动作选择得到（即分别基于 $\mathbb{E}(x_i) - c_i^{\text{reveal}}$ 和 $x_i - c_i^{\text{reveal}}$）[6]。此外，“Optimal” 和 “Optimal without human” 生成的平均效用值随揭开代价递减的速度超过 “Highest expected value”。这是由于揭开代价递减时，前面两种策略将揭开更少的物品且这些策略中的动作顺序之间的差异将降低。

最后，基于这些实验结果，可以总结得出：论文的策略 “Optimal” 显著优于相关的基准测试算法，而且通过良好的人与机器人交互，机器人执行任务的性能得到了明显提升。

[6] 以 “Highest expected value” 为例，随着揭开代价的降低，所有物品的期望值（即 $\mathbb{E}(x_i) - c_i^{\text{reveal}}, \forall i \in \bar{S}$）降低同样的额度，因此这些物品的揭开顺序并不改变且不同 c_i^{reveal} 的搜索顺序保持不变，即不同揭开代价物品构成的相同集合被揭开而且同一个物品被手机带走。因此，不同揭开代价的同一个想定下最终获得的效用值随揭开代价递减。

4.6 本章小结

本章建立了一种新的模型，描述机器人在不确定的知识和人辅助下的环境中进行搜索的问题。特别的是，论文考虑的想定为：一个机器人搜索一个物品并收集带走，且这些待搜索物品的效用值在机器人揭开或人检查之前是未知的。论文提出了一种多项式时间最优搜索策略并且通过实验证明论文的方法能够有效求解这类搜索问题，且性能优于其他的基准测试算法。

此外，论文的方法能够扩展到更为通用的问题，不是考虑获取最佳的物品为目标，而是考虑的目标为最大化所有发现物品集合的一个一般性的函数。具体地，本章的搜索规则可以根据文献 [126] 中技术将潘多拉规则扩展至求解这种更通用的问题。幸运的是，通过这种扩展，算法的复杂度和最优性仍然能够保持不变。

论文下一步的工作将考虑在揭开某个物品之后的切换代价（比如不同位置之间的路径）。由于不同物品之间的这些代价并不独立，论文的搜索策略将无法继续保证最优性，需要设计新的方法来求解这类具有耦合代价的问题。然而，可以设计基于指标的近似策略算法并将其和其他现有的规划方法（比如定向越野问题、图搜索问题及马尔科夫决策过程的相关方法）进行比较。

然后，希望考虑在线搜索的问题，其中的物品可能依次出现，然后机器人将需要考虑选择某个已经出现的物品进行搜索，或者等待新的物品出现后再综合考虑后继续进行搜索。再次，希望考虑在一些模型的初始参数未知时的学习方法。比如，人的可利用性可能很难进行精确表达，或者在搜索过程中可能会发生变化；在这种情况下，机器人的策略需要平衡这种探索和采用（Exploration 和 Exploitation）。最后，希望能在实际平台的真实机器人中应用和测试论文的搜索方法。

第五章　总结与展望

5.1　全文工作总结

无人机集群作战，将成为未来战争的重要作战手段之一。如何根据指挥信息系统提供的作战命令和交战规则，结合相关数据和模型，对这种大规模无人机团队部署于复杂环境中进行任务规划，成为当前科学研究的一项重要挑战。论文将无人机集群系统抽象看作多智能体系统，针对侦察监视任务规划问题，开展了相关理论与方法的研究。在多智能体系统中，多个可以独立行动的智能体，参与同环境的交互。而多智能体规划问题，可以看做由智能体团队成员共同协调为各自行为进行规划。在完全合作的多智能体系统中，所有智能体只拥有一个共同的目标。本论文针对侦察监视问题，设计一系列的规划算法并对其进行理论分析和仿真实验验证。论文的研究成果将对未来无人机集群任务规划系统的建设提供非常重要的理论支撑和方法保证。具体地，论文的主要研究工作和取得的成果如下：

（1）基于预测性启发式的集中式侦察监视规划方法。提出了在不确定和威胁下的集中式多智能体侦察监视的问题模型，这种模型不仅考虑了动态环境的部分可观和非静态的特性，而且考虑了侦察监视智能体的健康状态；设计了一种预测性启发式，来评估智能体当前时刻所有可能计划的值，并基于此设计了一种单智能体的在线规划算法；进一步，提出了一种基于顺次分配技术的多智能体算法，并证明了这种多智能体算法比当前的其他算法能够扩展到更大规模的问题中。

（2）基于树搜索和 max-sum 的分散式侦察监视规划方法。提出了一种不确定和威胁下的分散式侦察监视问题模型，这种模型不仅考虑了动态环境中的部分可观和非静态特性以及智能体的健康约束，而且也明确考虑了分散式交互的运行模式；基于 MCTS 和 max-sum 设计了具有可扩展性的分散式在线规划算法，这种方法的新颖性在于每个智能体构建并扩展一个向前看的搜索树，且相互之间通过分散式的消息传播方式不断更新各自的搜索树；通过仿真实验证明，对于 6 个智能体的问题，论文的方法求得的结果超过基准测试算法 56%，并且能够扩展至超过 24 个智能体的问题中。

（3）人辅助下基于潘多拉规则的搜索方法。提出了一种新的形式化模型，描述一个机器人在不确定知识和人辅助下的搜索问题。这一模型不仅考虑在搜索中机器人主动寻求人的辅助，而且考虑到物品回报值和人的可利用的不确定性；设计了一个多项式时间算法来求解这种搜索问题，并通过理论证明其最优性。与其他更为通用的求解方法相比，本章的方法能够显著提高计算效率；通过仿真实验表明，该方法显著优于一些相关的测试集方法。

5.2 研究展望

对于无人机集群系统任务规划的进一步研究，主要考虑在人与智能体集合的任务规划系统，以及非完全合作多智能体系统中的决策问题。

（1）人与智能体集合的任务规划。人的行为和操作将越来越多的被任务规划系统依赖并交叉于任务规划系统当中。随着这一趋势的发展，任务规划系统将明显涉及到人与软件智能体在系统中共生交错的现象。从而，这种人与智能体的紧密关系的出现，将对任务规划系统带来深远的影响 [78]。具体地，与之前的系统中只是配置被动的机器（保持等待状态直到有人干预）相比，未来将研究如何自主地和智能地将这些高度交互的计算单元串联起来。这种变化，需要具备处理所获得的信息和服务的规模、变化和速度的能力。系统中的个体（包括人和智能体）不可能完全了解相互之间的联系并手动将其连接起来。计算机需要基于用户的偏好和约束，更主动来引导用户并与其进行交互操作。所以，如何平衡人和机器的控制 / 操作，需要引起更多的关注。在一些想定任务中，人担任主管角色，智能体主要担任支持的角色，并提供建议和操作选项。然而，在另外一些想定任务中，智能体主要担任控制的角色，人担任支持的角色（如自主泊车系统和股票市场中的自动交易系统）。而且，在任务执行的过程当中，这些人和智能体的关系可能发生改变（如某个任务的操作人员可能被更紧急的任务中断，使得这个任务只能由“不需手动干预”的方式来进行处理；或者某个智能体可能遇到了意外的情形，需要寻求人的辅助来执行这项原本计划自主完成的任务）。这类新的规划系统称为人与智能体集合的任务规划系统，即人、软件智能体及硬件智能体能够在紧密的关系中进行灵活交互的任务规划系统。这种规划系统，一方面展现出更强的自主性，另一方面具有着开放性和社会性，这种开放性是指参与者需要不断地、灵活地建立和管理一系列的协作关系。因此，基于某个手头上的任务，不同的人、资源和数据需要汇集到一起，协作操作运行，然后解散。这种开放性，以及参与者拥有不同的利益模式（每个都有自己的资源和目标）的存在，意味着个体的参与是源于一种激励，这种激励包括外在的（如工作或奖励的要求）和内在的（如热衷于参与公众或国防事业），而不仅仅是强制委派。

（2）非完全合作多智能体系统中的决策问题。在真实世界的很多应用当中，智能体往往是自私的或者不是完全合作的，其规划目标是最大化自己的利益。对于这种自私型智能体构成的团队，需要设计有效的机制，使得每个智能体参与信息收集的过程中，最大化自身目标的同时，保证全局的态势感知性能。一方面，机制设计（Mechanism Design），作为博弈论研究领域的一个分支，就是研究这种规则设计的问题，其中每个参与者拥有未知的和私有的效用，规则设计者的目标就是

设计这样的机制，能够实现其独立于每个智能体的效用值的均衡（Equilibrium）。另一方面，未来的信息来源各种各样，比如移动手持设备大量分布于人们的社会生活当中。这些设备往往具有日益增强的功能，包括获取、分类和传输图像，声音，位置以及其他数据。参与式感知（Participatory Sensing），研究如何通过公共用户或者专业用户，利用这些移动设备，进行收集、分析和共享其局部的信息。论文进一步将研究基于用户参与的经济学计算模型，考虑具体信息感知的时间空间特性，设计激励用户参与的激励性机制和诚实性机制。

致　谢

在攻读博士学位期间，我所从事的学习和课题研究工作，都是在导师和实验室其他老师和同学的帮助下开展的。在论文即将收尾之际，请容许我向他们致以最诚挚的谢意！

首先感谢我的导师沈林成教授，感谢沈老师一直以来给予我悉心的指导和教诲。博士论文研究工作的点点滴滴，凝聚了导师大量的心血、不断的鼓励和支持。沈老师宽广的胸怀、敏锐的学术洞察力和高瞻远瞩的科研眼光深深影响着我，定将使我受益终生。沈老师严谨的治学态度和忘我的工作精神激励着我，使我能够保持沉着冷静、耐心钻研，攻克论文研究中遇到的一个又一个难题。谨在此向沈老师致以崇高的敬意和衷心的感谢！

感谢陈璟教授，感谢陈老师在我的学习、工作和生活的点点滴滴当中给予的指导和帮助。陈老师分析和解决问题的大局观以及追求创新的学术精神深深影响着我。在整个研究生学习期间，感谢陈老师不断教导和敦促我要追求真理，坚持科学精神和科学方法，使我收获了超越知识本身的科研法宝！

感谢潘亮副研究员，以及张万鹏、王楠、张煜、刘鸿福、谷学强、肖明、谢愈、相晓嘉、周晗、贾圣德和项凤涛等师兄师姐们的指点和照顾，与你们的讨论开阔了我的研究思路；感谢实验室的师弟师妹们，谢冰、任保安、崔璨、陈浩、刘扬、姜山、刘培根、周文宏、贾凡、曹欣芹、翁郁、王雪莹、方志勇和宋耀波等，和你们在空天任务规划实验室一起学习，共同走过了这段愉快的时光；并感谢陈浩和周文宏对论文校对工作作出的贡献。

感谢 Sarvapali D. Ramchurn 教授在我访问英国南安普顿大学 AIC 课题组期间的关怀和帮助。感谢 AIC 课题组的 Nick Jennings 教授、吴锋副教授、赵登吉、Tim Baarslag 和 Ruben Stranders 对论文的指导和贡献。同时感谢 AIC 课题组共同工作和学习过的尚贝宁、Sherry Yang、Chetan S. Mehra、Henry N.C. Truong、Alexandros Zenonos、Evangelos Tolias、Long Tran-Thanh、Zoltan Beck 等。

感谢国防科大的所有朋友们，特别是一起踢足球的伙伴们，和你们一起努力拼搏、赢取比赛的日子，使我感受到团队协作带来的快乐。

感谢我的父母，感谢你们的养育之恩！感谢你们一直对我的鼓励和支持！

感谢所有的老师、亲人、朋友和同学！祝大家幸福、平安！

陈少飞

二零一六年三月于长沙

参考文献

[1] Barca J C, Sekercioglu Y A. Swarm robotics reviewed [J]. Robotica. 2013, 31 (03): 345–359.

[2] Silver D, Veness J. Monte-Carlo Planning in Large POMDPs [C]. In NIPS. 2010: 2164–2172.

[3] Wooldridge M. An introduction to multiagent systems [M]. John Wiley & Sons, 2009.

[4] 沈林成, 陈璟, 王楠. 飞行器任务规划综述 [J]. 航空学报. 2014, 35 (3): 593–606.

[5] 陈璟, 张万鹏, 任敏, et al. 飞行任务规划 [M]. 国防科技大学出版社, 2015.

[6] 张万鹏. XXXX 分层任务网络规划建模及优化技术 [D]. [S. l.]: 国防科学技术大学, 2012.

[7] 王楠. XXXX 综合任务规划技术 [D]. [S. l.]: 国防科学技术大学, 2012.

[8] 张煜. XXXX 空对地攻击轨迹规划技术研究 [D]. [S. l.]: 国防科学技术大学, 2012.

[9] 刘鸿福. XXXX 隐身轨迹规划技术研究 [D]. [S. l.]: 国防科学技术大学, 2013.

[10] 谷学强. 多机协同 XXXX 轨迹规划技术研究 [D]. [S. l.]: 国防科学技术大学, 2014.

[11] 陈少飞. 隐身飞行器突防问题特性分析与航迹规划方法研究 [D]. [S. l.]: 国防科学技术大学, 2011.

[12] Stranders R, de Cote E M, Rogers A, et al. Near-optimal continuous patrolling with teams of mobile information gathering agents [J]. Artif. Intell. 2013, 195: 63–105.

[13] Farinelli A, Rogers A, Petcu A, et al. Decentralised coordination of low-power embedded devices using the max-sum algorithm [C]. In AAMAS. 2008: 639–646.

[14] Singh A, Krause A, Kaiser W J. Nonmyopic Adaptive Informative Path Planning for Multiple Robots [C]. In IJCAI. 2009: 1843–1850.

[15] Ny J L, Dahleh M, Feron E. Multi-UAV dynamic routing with partial observations using restless bandit allocation indices [C]. In American Control Conference, 2008. 2008: 4220–4225.

[16] Agmon N, Kraus S, Kaminka G A, et al. Adversarial Uncertainty in Multi-Robot Patrol [C]. In IJCAI. 2009: 1811–1817.

[17] Basilico N, Gatti N. Automated Abstractions for Patrolling Security Games [C]. In AAAI. 2011: 1096–1101.

[18] Qian Y, Haskell W B, Jiang A X, et al. Online planning for optimal protector strategies in resource conservation games [C]. In AAMAS. 2014: 733–740.

[19] Yost K A, Washburn A R. The LP/POMDP marriage: Optimization with imperfect information [R]. 2000.

[20] Whittle P. Restless bandits: Activity allocation in a changing world [J]. Journal of Applied Probability. 1988, 25: 287–298.

[21] Castanón D A. Stochastic control bounds on sensor network performance [C]. In IEEE Conference on Decision and Control. 2005: 4939–4944.

[22] Zois D-S, Levorato M, Mitra U. Energy-efficient, heterogeneous sensor selection for physical activity detection in wireless body area networks [J]. IEEE Transactions on Signal Processing. 2013, 61 (5-8): 1581–1594.

[23] Zhao Q, Tong L, Swami A, et al. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework [J]. IEEE Journal on Selected Areas in Communications. 2007, 25 (3): 589–600.

[24] Ouyang Y, Teneketzis D. On the Optimality of Myopic Sensing in Multi-State Channels [J]. IEEE Transactions on Information Theory. 2014, 60 (1): 681–696.

[25] Singh A, Krause A, Guestrin C, et al. Efficient planning of informative paths for multiple robots [C]. In IJCAI. 2007: 2204–2211.

[26] Singh A, Krause A, Guestrin C, et al. Efficient informative sensing using multiple robots [J]. Journal of Artificial Intelligence Research. 2009: 707–755.

[27] Meliou A, Krause A, Guestrin C, et al. Nonmyopic informative path planning in spatio-temporal models [C]. In AAAI. 2007: 16–7.

[28] Paruchuri P, Pearce J P, Tambe M, et al. An efficient heuristic approach for security against multiple adversaries [C]. In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. 2007: 181.

[29] Tsai J, Yin Z, Kwak J-y, et al. Urban security: Game-theoretic resource allocation in networked physical domains [C]. In AAAI. 2010.

[30] Basilico N, Gatti N, Amigoni F. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies [C]. In AAMAS. 2009: 57–64.

[31] Agmon N, Kraus S, Kaminka G, et al. Multi-robot perimeter patrol in adversarial settings [C]. In Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. 2008: 2339–2345.

[32] Elmaliach Y, Agmon N, Kaminka G A. Multi-robot area patrol under frequency constraints [J]. Annals of Mathematics and Artificial Intelligence. 2009, 57 (3-4): 293–320.

[33] Grocholsky B. Information-theoretic control of multiple sensor platforms [D]. [S. l.]: University of Sydney. School of Aerospace, Mechanical and Mechatronic Engineering, 2002.

[34] Fiorelli E, Leonard N E, Bhatta P, et al. Multi-AUV control and adaptive sampling in Monterey Bay [J]. Oceanic Engineering, IEEE Journal of. 2006, 31 (4): 935–948.

[35] Martinez-Cantin R, de Freitas N, Doucet A, et al. Active Policy Learning for Robot Planning and Exploration under Uncertainty. [C]. In Robotics: Science and Systems. 2007: 321–328.

[36] Ahmadi M, Stone P. A multi-robot system for continuous area sweeping tasks [C]. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on. 2006: 1724–1729.

[37] Stranders R, Farinelli A, Rogers A, et al. Decentralised coordination of mobile sensors using the max-sum algorithm [C]. In IJCAI. 2009: 299–304.

[38] Stranders R, Fave F M D, Rogers A, et al. A Decentralised Coordination Algorithm for Mobile Sensors [C]. In AAAI. 2010.

[39] Rasmussen C E. Gaussian processes for machine learning [J]. 2006.

[40] Guestrin C, Krause A, Singh A P. Near-optimal sensor placements in gaussian processes [C]. In Proceedings of the 22nd international conference on Machine learning. 2005: 265–272.

[41] Ko C-W, Lee J, Queyranne M. An exact algorithm for maximum entropy sampling [J]. Operations Research. 1995, 43 (4): 684–691.

[42] Bai X, Kumar S, Xuan D, et al. Deploying wireless sensors to achieve both coverage and connectivity [C]. In Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. 2006: 131–142.

[43] Stranders R, Rogers A, Jennings N R. A Decentralised Coordination Algorithm for Minimising Conflict and Maximising Coverage in Sensor Networks [C]. In AAMAS. Richland, SC, 2010: 1165–1172.

[44] Bellman R. Dynamic programming and Lagrange multipliers [J]. Proceedings of the National Academy of Sciences of the United States of America. 1956, 42 (10): 767.

[45] Howard R. Dynamic Programming and Markov Processes [M]. MIT Press, 1960.

[46] Feinberg E A, Shwartz A. Handbook of Markov decision processes: methods and applications [M]. Springer Science & Business Media, 2012.

[47] Boutilier C, Dean T, Hanks S. Decision-theoretic planning: Structural assumptions and computational leverage [J]. Journal of Artificial Intelligence Research. 1999, 11 (1): 94.

[48] Dean T L, Kaelbling L P, Kirman J, et al. Planning With Deadlines in Stochastic Domains. [C]. In AAAI. 1993: 574–579.

[49] Dearden R, Boutilier C. Integrating planning and execution in stochastic domains [C]. In Proceedings of the Tenth international conference on Uncertainty in artificial intelligence. 1994: 162–169.

[50] Koenig S. Optimal probabilistic and decision-theoretic planning using Markovian decision theory [M]. University of California, Berkeley, Computer Science Division, 1992.

[51] Sutton R S, Barto A G. Reinforcement learning: An introduction [M]. MIT press Cambridge, 1998.

[52] Balas E. The prize collecting traveling salesman problem [J]. Networks. 1989, 19 (6): 621–636.

[53] Jaillet P. Probabilistic travelling salesman problem [D]. [S. l.]: Massachusetts Institute of Technology, 1984.

[54] Garg N. Saving an epsilon: a 2-approximation for the k-MST problem in graphs [C]. In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing. 2005: 396–402.

[55] Tsiligirides T. Heuristic methods applied to orienteering [J]. Journal of the Operational Research Society. 1984: 797–809.

[56] Ilhan T, Iravani S M, Daskin M S. The orienteering problem with stochastic profits [J]. Iie Transactions. 2008, 40 (4): 406–421.

[57] Dantzig G B, Ramser J H. The truck dispatching problem [J]. Management science. 1959, 6 (1): 80–91.

[58] Gendreau M, Laporte G, Séguin R. Stochastic vehicle routing [J]. European Journal of Operational Research. 1996, 88 (1): 3–12.

[59] Gendreau M, Laporte G, Séguin R. An exact algorithm for the vehicle routing problem with stochastic demands and customers [J]. Transportation science. 1995, 29 (2): 143–155.

[60] Jennings N R. On agent-based software engineering [J]. Artificial intelligence. 2000, 117 (2): 277–296.

[61] Dias M, Zlot R, Kalra N, et al. Market-Based Multirobot Coordination: A Survey and Analysis [J]. Proceedings of the IEEE. 2006, 94 (7): 1257–1270.

[62] Cole D T. A Cooperative Unmanned Aircraft System Architecture for Information-Theoretic Search and Track. [D]. [S. l.]: University Of Sydney, 2009.

[63] Farinelli A, Rogers A, Jennings N R. Agent-based decentralised coordination for sensor networks using the max-sum algorithm [J]. Autonomous agents and multi-agent systems. 2014, 28 (3): 337–380.

[64] Scerri P, Liao E, Xu Y, et al. Coordinating very large groups of wide area search munitions [J]. Theory and Algorithms for Cooperative Systems. 2005.

[65] Sultanik E, Modi P J, Regli W C. On Modeling Multiagent Task Scheduling as a Distributed Constraint Optimization Problem. [C]. In IJCAI. 2007: 1531–1536.

[66] Waharte S, Trigoni N. Supporting search and rescue operations with UAVs [C]. In Emerging Security Technologies (EST), 2010 International Conference on. 2010: 142–147.

[67] Furukawa T, Bourgault F, Lavis B, et al. Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets [C]. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on. 2006: 2521–2526.

[68] Thrun S, Burgard W, Fox D. Probabilistic robotics [M]. MIT press, 2005.

[69] 叶媛媛. 多 UCAV 协同任务规划方法研究 [D]. [S. l.]: 国防科学技术大学, 2005.

[70] 霍霄华. 多 UCAV 动态协同任务规划建模与滚动优化方法研究 [D]. [S. l.]: 国防科学技术大学, 2007.

[71] 田菁. 多无人机协同侦察任务问题建模与优化技术研究 [D]. [S. l.]: 国防科学技术大学, 2007.

[72] 彭辉. 分布式多无人机协同区域搜索中的关键问题研究 [D]. [S. l.]: 国防科学技术大学, 2009.

[73] 李远. 多 UAV 协同任务资源分配与编队轨迹优化方法研究 [D]. [S. l.]: 国防科学技术大学, 2010.

[74] 张庆杰. 基于一致性理论的多 UAV 分布式协同控制与状态估计方法 [D]. [S. l.]: 国防科学技术大学, 2011.

[75] 苏菲. 动态环境下多 UCAV 分布式在线协同任务规划技术研究 [D]. [S. l.]: 国防科学技术大学, 2013.

[76] Antoniades A, Kim H J, Sastry S. Pursuit-evasion strategies for teams of multiple agents with incomplete information [C]. In Decision and Control, 2003. Proceedings. 42nd IEEE Conference on. 2003: 756–761.

[77] Bethke B, Valenti M, How J P. UAV task assignment [J]. Robotics & Automation Magazine, IEEE. 2008, 15 (1): 39–44.

[78] Jennings N R, Moreau L, Nicholson D, et al. Human-agent Collectives [J]. Commun. ACM. 2014, 57 (12): 80–88.

[79] Schmidt-Rohr S R, Knoop S, Losch M, et al. Reasoning for a multi-modal service robot considering uncertainty in human-robot interaction [C]. In Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on. 2008: 249–254.

[80] Rosenthal S, Veloso M. Modeling humans as observation providers using pomdps [C]. In RO-MAN, 2011 IEEE. 2011: 53–58.

[81] Fong T, Thorpe C, Baur C. Robot, asker of questions [J]. Robotics and Autonomous systems. 2003, 42 (3): 235–243.

[82] Stone P, Kaminka G A, Kraus S, et al. Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination. [C]. In AAAI. 2010.

[83] Gao F, Cummings M, Solovey E. Modeling Teamwork in Supervisory Control of Multiple Robots [J]. Human-Machine Systems, IEEE Transactions on. 2014, 44 (4): 441–453.

[84] Weiss A, Igelsböck J, Tscheligi M, et al. Robots asking for directions: the willingness of passers-by to support robots [C]. In Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction. 2010: 23–30.

[85] Mekonnen A A, Lerasle F, Herbulot A. Cooperative passers-by tracking with a mobile robot and external cameras [J]. Computer Vision and Image Understanding. 2013, 117 (10): 1229–1244.

[86] Kaelbling L P, Littman M L, Cassandra A R. Planning and acting in partially observable stochastic domains [J]. Artificial intelligence. 1998, 101 (1): 99–134.

[87] Vansteenwegen P, Souffriau W, Van Oudheusden D. The orienteering problem: A survey [J]. European Journal of Operational Research. 2011, 209 (1): 1–10.

[88] Shasha D, Wang J T, Giugno R. Algorithmics and applications of tree and graph searching [C]. In Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. 2002: 39–52.

[89] Kang S, Ouyang Y. The traveling purchaser problem with stochastic prices: Exact and approximate algorithms [J]. European Journal of Operational Research. 2011: 265–272.

[90] Baarslag T, Gerding E H. Optimal incremental preference elicitation during negotiation [C]. In IJCAI. 2015.

[91] Shiryaev A N. Optimal stopping rules [M]. Springer Science & Business Media, 2007.

[92] Freeman P. The secretary problem and its extensions: A review [J]. International Statistical Review/Revue Internationale de Statistique. 1983: 189–206.

[93] Stone L D．Theory of optimal search / Lawrence D. Stone [M]．Academic Press New York，1975．

[94] Amin K I, Lee C. Option Trading, Price Discovery, and Earnings News Dissemination* [J]. Contemporary Accounting Research. 1997, 14 (2): 153–192.

[95] Baarslag T, Hindriks K V. Accepting optimally in automated negotiation with incomplete information [C]. In AAMAS. 2013: 715–722.

[96] Schut M, Wooldridge M, Parsons S. On partially observable MDPs and BDI models [M] // Schut M, Wooldridge M, Parsons S. Foundations and Applications of Multi-Agent Systems. Springer, 2002: 2002: 243–259.

[97] Smallwood R D, Sondik E J. The optimal control of partially observable Markov processes over a finite horizon [J]. Operations Research. 1973, 21 (5): 1071–1088.

[98] Sandholm W H. Orders of limits for stationary distributions, stochastic dominance, and stochastic stability [J]. Theoretical Economics. 2010, 5 (1): 1–26.

[99] Keilson J, Kester A. Monotone matrices and monotone Markov processes [J]. Stochastic Processes and their Applications. 1977, 5 (3): 231–241.

[100] Pynadath D V, Tambe M. The Communicative Multiagent Team Decision Problem: Analyzing Teamwork Theories and Models [J]. Journal of Artificial Intelligence Research. 2002, 16: 389–423.

[101] Yokoo M. Distributed constraint satisfaction: foundations of cooperation in multi-agent systems [M]. Springer Publishing Company, Incorporated, 2012.

[102] Bernstein D S, Givan R, Immerman N, et al. The complexity of decentralized control of Markov decision processes [J]. Mathematics of operations research. 2002, 27 (4): 819–840.

[103] Witwicki S J. Abstracting influences for efficient multiagent coordination under uncertainty [D]. [S. l.]: University of Michigan, 2011.

[104] Guestrin C, Koller D, Parr R, et al. Efficient solution algorithms for factored MDPs [J]. Journal of Artificial Intelligence Research. 2003: 399–468.

[105] Witwicki S J, Durfee E H. Influence-Based Policy Abstraction for Weakly-Coupled Dec-POMDPs. [C]. In ICAPS. 2010: 185–192.

[106] Fave F M D. Theory and practice of decentralised coordination algorithms exploiting the generalised distributive law [D]. [S. l.]: University of Southampton, 2012.

[107] Amato C, Oliehoek F A. Scalable Planning and Learning for Multiagent POMDPs [C]. In AAAI. 2015.

[108] Ong S C, Png S W, Hsu D, et al. Planning under uncertainty for robotic tasks with mixed observability [J]. The International Journal of Robotics Research. 2010, 29 (8): 1053–1068.

[109] Becker R, Zilberstein S, Lesser V. Decentralized Markov decision processes with event-driven interactions [C]. In AAMAS. 2004: 302–309.

[110] Allen M W. Agent interactions in decentralized environments [D]. [S. l.]: University of Massachusetts Amherst, 2009.

[111] Kschischang F R, Frey B J, Loeliger H-A. Factor graphs and the sum-product algorithm [J]. IEEE Transactions on Information Theory. 2001, 47 (2): 498–519.

[112] MacKay D J C. Information Theory, Inference & Learning Algorithms [M]. New York, NY, USA: Cambridge University Press, 2002.

[113] Liu Y, Nejat G. Robotic urban search and rescue: A survey from the control perspective [J]. Journal of Intelligent & Robotic Systems. 2013, 72 (2): 147–165.

[114] Fogarty J, Hudson S E, Atkeson C G, et al. Predicting human interruptibility with sensors [J]. ACM Transactions on Computer-Human Interaction (TOCHI). 2005, 12 (1): 119–146.

[115] Murphy R R. Human-robot interaction in rescue robotics [J]. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on. 2004, 34 (2): 138–153.

[116] Nourbakhsh I R, Sycara K, Koes M, et al. Human-robot teaming for search and rescue [J]. Pervasive Computing, IEEE. 2005, 4 (1): 72–79.

[117] Goodrich M A, Schultz A C. Human-robot interaction: a survey [J]. Foundations and trends in human-computer interaction. 2007, 1 (3): 203–275.

[118] Bresina J L, Morris P H. Mixed-initiative planning in space mission operations [J]. AI magazine. 2007, 28 (2): 75.

[119] Talamadupula K, Benton J, Kambhampati S, et al. Planning for human-robot teaming in open worlds [J]. ACM Transactions on Intelligent Systems and Technology (TIST). 2010, 1 (2): 14.

[120] Rosenthal S, Veloso M M. Mobile Robot Planning to Seek Help with Spatially-Situated Tasks. [C]. In AAAI. 2012: 1.

[121] Hazon N, Aumann Y, Kraus S, et al. Physical search problems with probabilistic knowledge [J]. Artificial Intelligence. 2013, 196: 26–52.

[122] Rosenthal S, Veloso M M, Dey A K. Learning Accuracy and Availability of Humans Who Help Mobile Robots. [C]. In AAAI. 2011.

[123] Rosenthal S, Veloso M, Dey A K. Is someone in this office available to help me? [J]. Journal of Intelligent & Robotic Systems. 2012, 66 (1-2): 205–221.

[124] Bellman R. Dynamic programming treatment of the travelling salesman problem [J]. Journal of ACM. 1962, 9 (1): 61–63.

[125] Weitzman M L. Optimal search for the best alternative [J]. Econometrica: Journal of the Econometric Society. 1979: 641–654.

[126] Olszewski W, Weber R. A more general Pandora rule? [J]. Journal of Economic Theory. 2015, 160: 429 – 437.

[127] Browne C B, Powley E, Whitehouse D, et al. A survey of monte carlo tree search methods [J]. Computational Intelligence and AI in Games, IEEE Transactions on. 2012, 4 (1): 1–43.

[128] Gelly S, Silver D. Achieving Master Level Play in 9 x 9 Computer Go. [C]. In AAAI. 2008: 1537–1540.

[129] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search [J]. Nature. 2016, 529 (7587): 484–489.

[130] Lorentz R J. Amazons discover monte-carlo [M] // Lorentz R J. Computers and games. Springer, 2008: 2008: 13–24.

[131] Finnsson H, Björnsson Y. Simulation-Based Approach to General Game Playing. [C]. In AAAI. 2008: 259–264.

[132] Baier H, Drake P D. The power of forgetting: Improving the last-good-reply policy in Monte Carlo Go [J]. Computational Intelligence and AI in Games, IEEE Transactions on. 2010, 2 (4): 303–309.

[133] Aji S M, McEliece R J. The generalized distributive law [J]. Information Theory, IEEE Transactions on. 2000, 46 (2): 325–343.

[134] Bishop C M, et al. Pattern recognition and machine learning [M]. springer New York, 2006.

作者在学期间取得的学术成果

发表的学术论文

[1] **Shaofei Chen**, Tim Baarslag, Dengji Zhao, Jing Chen, Lincheng Shen. A Polynomial Time Optimal Algorithm for Robot-Human Search under Uncertainty. *In Proceedings of the 25th International Joint Conference on Artificial Intelligence*, (IJCAI), New York, USA, 2016. (人工智能顶级会议)

[2] **Shaofei Chen**, Feng Wu, Lincheng Shen, Jing Chen and Sarvapali D. Ramchurn. Decentralised Patrolling under Constraints in Dynamic Environments. *IEEE Transactions on Cybernetics*, 2016. (SCI 收录, 影响影子：3.469)

[3] **Shaofei Chen**, Feng Wu, Lincheng Shen, Jing Chen and Sarvapali D. Ramchurn. Multi-Agent Patrolling under Uncertainty and Threats. *PLoS ONE*, 10(6): e0130154, 2015. (SCI 收录，影响影子：3.234)

[4] **Shaofei Chen**, Hongfu Liu, Jing Chen, Lincheng Shen. Penetration Trajectory Planning based on Radar Tracking Features for UAV. *Aircraft Engineering and Aerospace Technology*, 2013, 85(1):62-71. (SCI 收录, 影响影子：0.48)

[5] Yu Zhang, Jingzhao Yang, **Shaofei Chen**, Jing Chen. Decentralized Cooperative Trajectory Planning for Multiple UAVs in Dynamic and Uncertain Environments. *IEEE Seventh International Conference on Intelligence Computing and Information Systems*, 2016. (EI 收录)

[6] Hongfu Liu, **Shaofei Chen**, Lincheng Shen, Jing Chen. An Integrated Multi-criterion hp-Adaptive Pseudospectral Method for Direct Optimal Control Problems Solving. *Mathematical Problems in Engineering*, 2012. (SCI 收录)

[7] Hongfu Liu, **Shaofei Chen**, Lincheng Shen, Jing Chen. Tactical Trajectory Planning for Stealth UAV to Win the Radar Game. *Defense Science Journal*, 2012, 62(6):375-381. (SCI 收录)

[8] Hongfu Liu, Lincheng Shen, Jing Chen, **Shaofei Chen**. Low observability trajectory planning for stealth aircraft to evade radars tracking. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2012, 228(3):398-410. (SCI 收录)

[9] 陈璟, **陈少飞**, 刘鸿福. 基于 RCS 的三维低可探测性轨迹优化方法. 国防科技大学学报, 2012, 34(3):89-93. (EI 收录)

获得的学术奖励

[1] 国家建设高水平大学公派研究生资助项目（No. 201306110066）

[2] 湖南省研究生科研创新项目（No. CX2013B013）

[3] 国防科技大学优秀研究生创新资助（No. B130302）

参与的科研项目

[1] 国家自然科学基金（No. 61403411）：高动态环境下低可探测性飞行器自主任务规划方法研究

附录 A　蒙特卡洛树搜索在线求解 POMDP

论文第3章设计基于蒙特卡洛树搜索（MCTS）的在线求解算法，是因为在许多具有挑战性的问题中，MCTS 的表现均优于其他规划算法（详见文献 [127] 中的综述）。其中的问题包括求解围棋 [128, 129]、Amazons[130] 和一般的游戏 [131] 的策略。如图A.1所示，MCTS 的基本过程在概念上很简单（详见文献 [132]）。通过不断增长和非对称的方式来构建一个树。算法的每次迭代过程中，通过一个树策略用来找到当前的树最紧急的节点。这个树策略尝试平衡对于探索（Exploration，朝向尚未很好采样过的区域）和（Exploitation，朝向具有良好前景的区域）。然后，从所选择节点开始运行一次仿真并根据运行结果来更新这个搜索树。这其中主要涉及对于与所选节点执行动作后如何添加子节点，以及如何更新其祖先节点的统计。这个仿真过程中根据一些默认的策略进行移动，其中最简单的情况为进行一致随机移动。MTCS 的巨大优势在于不需要对于一些中间状态进行估值或评价，如对于深度有限的最小最大搜索，将在很大程度上降低所需的领域知识。只需要知道每次仿真最终状态的值。

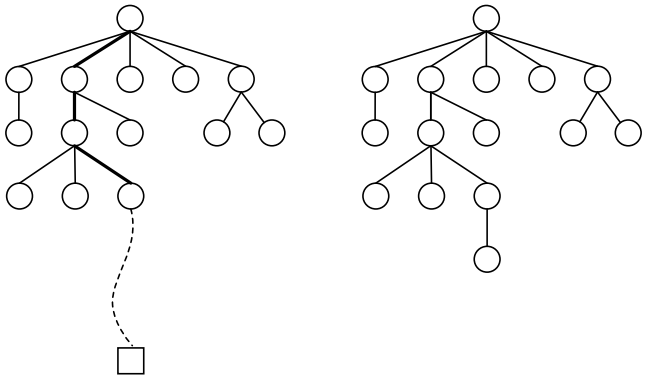


图 A.1　MCTS 的基本执行过程

特别地，部分可观蒙特卡洛规划 （Partially-Observable Monte Carlo Planning, POMCP），一种基于 MCTS 的算法，在大规模 POMDPs 问题中进行单智能体规划得到了成功应用 [2]。POMCP 的运行，通过由当前信念反复执行仿真试验来递增地构建一个向前看的搜索树。一个无权重的粒子滤波器（Particle Filter）用来对这个信念状态进行近似。通过部分可观的 UCT 算法 [2] 具体执行仿真。对于一次仿真中遇到的每个历史 h（动作和观测的历史），通过最大化置信区间的上界（Upper Confidence Bounds）来选择动作。而且，POMCP 在很多大的问题中只需要很少数

量的仿真就可以表现很好，且其收敛性的满足条件为 POMCP 由真实的信念状态中进行采样 [2]。此外，文献 [107] 将 POMCP 扩展到了求解多智能体 POMDP 中，其中假设所有智能体之间能够进行全通讯并对他们对世界的观察能够进行分享。然而，论文第3章中的智能体之间只允许进行少量智能体之间的局部通信。

附录 B　DCOPs 问题和 max-sum 算法

首先介绍分布式约束优化问题（DCOPs）的形式化模型，然后介绍如何使用 max-sum 算法来通过分散式的方式来对 DCOPs 进行求解。

B.1　DCOP 模型

一个标准的，可用于描述多智能体协调的 DCOP，为一个多元组 $\langle \mathcal{M}, \mathcal{X}, \mathcal{D}, \mathcal{V} \rangle$，其中 $\mathcal{M} = \{A_1, \ldots, A_{|\mathcal{M}|}\}$ 为智能体的集合，$\mathcal{X} = \{x_1, \ldots, x_{|\mathcal{M}|}\}$ 为变量的集合，并且每个变量被且仅被某一个智能体控制。智能体 A_i 负责对其所控制的变量 x_i 分配选取的值，其中 $i \in \mathcal{M}$。$\mathcal{D} = \{D_1, \ldots, D_{|\mathcal{M}|}\}$ 为一个离散的、有限个数的变量域，且每个变量 x_i 可以从其所属域 D_i 中选取值。然后，$\mathcal{V} = \{V_1, \ldots, V_n\}$ 为一组函数的集合，来表示变量之间的约束关系。每个函数 $V_j : D_{j_1} \times \cdots \times D_{j_{k_j}} \to \mathbb{R} \cup \{-\infty\}$ 依赖于变量集合 $\mathbf{x_j} \in \mathcal{X}$，其中 $k_i = |\mathbf{x_j}|$ 函数的参数数量且 $-\infty$ 用于表示严格的约束。每个函数分配给依赖其变量的每个可能的分配的一组取值。问题的目标是找到一组分配能够最大化约束函数的和：

$$\arg\max_{\mathbf{x}} \sum_{i} V_i(\mathbf{x_i}). \tag{B.1}$$

DCOPs 通常可以通过一个交互图来进行表示，其中圆圈表示变量，变量之间的变量之间的边表示其所连接的变量存在于某个约束当中。

对于求解 DCOPs 的算法已经成功应用在了大量的多智能体协调问题 [101] 当中。其中的一种近似算法，max-sum [63]，在求解大规模 DCOPs 当中表现出了很好的有效性。下面将对 max-sum 做详细的介绍。

B.2　Max-sum 算法

Max-sum 为一种通用消息传播算法（General Message Passing, GDL），基于 GDL 来分解分布式约束优化问题，即将问题表示成为简单因子的和或积，来降低其复杂的计算 [133]。使用时，DCOPs 需要表示成为一类特殊的双向图，称作因子图。一个因子图包括两类节点：变量节点（通常表示为圆圈）和函数节点（通常表示为方块）[13, 134]。无向边连接每个函数到其变量。一个因子图明确表示变量节点和函数节点之间的关系。例如：如图B.1（a）3 个智能体在环境中进行侦察监视，其中这些监视区域相互交叠且黑色圆圈为图中的顶点，（b）智能体的交互图，（c）问题转化的基于效能的因子图。考虑图B.1中表示的情形并以智能体 A_2 为例，将其值函数 V_2 表示的函数节点和其控制的变量 x_2（智能体 A_2 的动作）以及其他变量 x_1 和 x_3（邻居智能体的动作）进行连接。由这个因子图表示的整合函

数为 $V = V_1(x_1, x_2) + V_2(x_1, x_2, x_3) + V_3(x_2, x_3)$，即全局效能（Social Welfare）函数。已知 max-sum 在无环因子图所示的问题中表现出最优性，但是这种最优性对于优化的一般问题不能保证。然而，大量实例的应用证明这类算法能够达到很好的近似最优解。

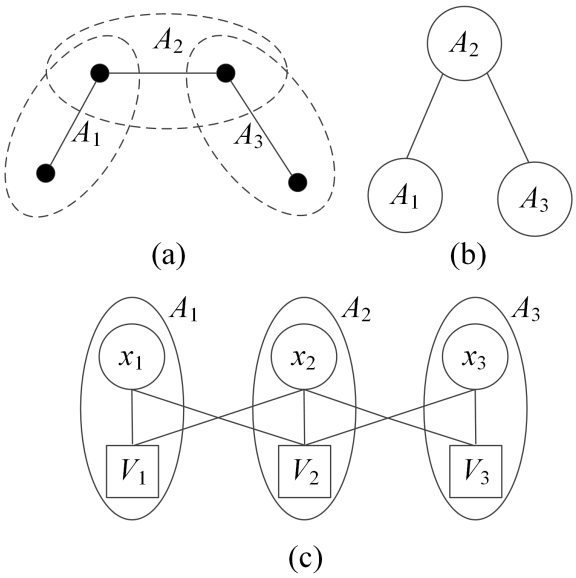


图 B.1 （a）3 个智能体在环境中进行侦察监视，其中这些监视区域相互交叠且黑色圆圈为图中的顶点（b）智能体的交互图（c）问题转化的基于效能的因子图

值得注意的是，这种形式化描述明确表示了一种将变量和函数明确分配给各个智能体的关系。换而言之，每个智能体负责自己所控制变量的取值，接收由其函数和变量节点传递过来的消息以及更新有其函数和变量传递出去的消息。通过这种方式，智能体之间连续协商可能执行的最佳动作，因此可以更加有效的对环境中可能的变化作出反应。

具体地，max-sum 通过变量到函数、函数到变量，不断地迭代进行消息传播。定义的变量和函数之间交换的两类消息如下：

- 由变量 x_i 到函数 V_j：

$$q_{i\to j}(x_i) = \alpha_{ij} + \sum_{k\in\mathcal{M}_i\backslash j} r_{k\to i}(x_i), \tag{B.2}$$

其中 $\mathcal{M}_i$ 表示与变量 x_i 连接的函数节点下标的集合，α_{ij} 为满足 $\sum_{x_i} q_{i\to j}(x_i) = 0$ 的常量。

- 由函数 V_j 到变量 x_i：

$$r_{j\to i}(x_i)=\max_{\mathbf{x_j}\setminus x_i}\left[V_j(\mathbf{x_j})+\sum_{k\in\mathcal{N}_j\setminus i}q_{k\to j}(x_k)\right], \tag{B.3}$$

其中 $\mathcal{N}_j$ 表示连接 V_j 的变量节点下标集合且 $\mathbf{x_j}$ 为变量向量 $\langle x_{x_1},\cdots,x_{j_{k_j}}\rangle$。

因此，每个变量 a^i 的边界函数可以由下式计算得出：

$$z_i(x_i)=\sum_{j\in\mathcal{M}_i}r_{j\to i}(x_i)\approx\max_{\mathbf{x}\setminus x_i}\sum_{j=1}^{m}V_j(\mathbf{x_j}), \tag{B.4}$$

并进一步计算出变量 x_i 所分配的值：

$$x_i^*=\arg\max_{x_i\in D_i}z_i(x_i). \tag{B.5}$$