文章编号:1001-2486(2011)01-0053-06

一种基于多 Agent 强化学习的多星协同任务规划算法*

王 冲,景 宁,李 军,王 钧,陈 浩
(国防科技大学 电子科学与工程学院,湖南 长沙 410073)

摘 要:在分析任务特点和卫星约束的基础上给出了多星协同任务规划问题的数学模型。引入约束惩罚算子和多星联合惩罚算子对卫星 Agent 原始的效用值增益函数进行改进,在此基础上提出了一种多卫星 Agent 强化学习算法以求解多星协同任务分配策略,设计了基于黑板结构的多星交互方式以降低学习交互过程中的通信代价。通过仿真实验及分析证明该方法能够有效解决多星协同任务规划问题。

关键词:卫星任务规划;协同规划;多智能体强化学习;黑板结构
中图分类号:TP391 **文献标识码**:A

An Algorithm of Cooperative Multiple Satellites Mission Planning Based on Multi-agent Reinforcement Learning

WANG Chong, JING Ning, LI Jun, WANG Jun, CHEN Hao
(College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A multi-satellite cooperative planning problem model was given considering the characteristics of the task requests and satellite constraints. Then the original performance function of each satellite agent was modified by introducing both the constraint punishing operator and the multi-satellite joint punishing operator. Next, a multi-satellite reinforcement learning algorithm (MUSARLA) was proposed to derive the coordinated task allocation strategy. Furthermore, the interaction among multiple satellites was designed based on blackboard architecture to reduce the communication cost while learning. Finally, simulated experiments are carried out which verified the effectiveness of the proposed algorithm.

Key words: satellite mission planning; cooperative planning; multi-agent reinforcement learning; blackboard architecture

对地观测卫星(Earth Observing Satellite, EOS)通过星载传感器从太空获取地面影像数据,已经成为勘测和研究地球资源的重要手段。如何高效利用多颗卫星的观测资源完成目标观测任务,是目前卫星任务规划领域的热点。当前,按照卫星任务规划模式可分为集中式[1]和分布式[2-5]协同规划方法。虽然集中式任务规划系统可从全局的角度对问题进行求解和优化,但在实际规划过程中存在求解复杂度高,鲁棒性差、可扩展性不足等局限,难以保证遥感需求的质量和时效性。

在卫星自主能力不断提高的前提下,分布式协同规划是卫星任务规划的重要发展方向。多颗具有自主规划能力的卫星通过星际链路组网进行协同任务规划可以通过自主控制对局部事态变化做出快速反应,能够充分发挥卫星资源的自治能力。与以往集中式任务规划相比,具有更好的灵活性、鲁棒性、容错性和可扩展性等优势。

目前,求解多星协同任务规划问题通常集中于两个方面:一种是基于层次化结构的多星任务协商分配方法[2-3]。通过卫星之间的关系划分为星座、星群和单星三个层次。将粗粒度的规划目标通过层次间协商分解分配至各颗卫星,协商过程中采用领域知识迭代修复的方法。另一种思路是根据不同类型的卫星抽象其约束信息,将每颗卫星映射为具有自主规划能力的 Agent,组成多智能体系统(Multi-Agent System, MAS)。结合卫星任务规划的特点采用基于 MAS 合同网协议[6](Contract Net Protocol, CNP)的思想进行求解。高黎[4]建立了基于心智模型的合同网任务分配模型,根据卫星完成任务情况的历史信息对卫星 Agent 心智参数进行评价,以指导后续任务分配。陈浩[5]在合同协议的基础上引入外包机制和免责原则,克服了多星协同任务规划方法中调度结果对任务招投标顺序依赖较大的问题。

* 收稿日期:2010-06-25
基金项目:国家自然科学基金资助项目(60604035);国家 863 高技术资助项目(2007AA12020203)
作者简介:王冲(1982—),男,博士生。

以上两种求解思路都建立在理想通信环境的基础上,在实际协同规划过程中产生大量通信代价。而且两种方式都存在规划管理节点,若节点失效系统则面临崩溃。若采用传统基于启发式算法(如:遗传算法等)求解多星任务规划问题则面临“建模难”(Curse of Modeling)的问题,模型参数的知识难以完全获取。为了有效解决多星协同任务规划问题,本文将每颗卫星作为一个自主学习 Agent,设计了模型无关的多卫星 Agent 强化学习算法,解决多星之间的规划决策问题。

1 多星协同规划模型建立

1.1 问题描述

对地观测卫星绕地球飞行,在飞抵待观测目标上空时通过星载传感器收集目标信息完成一次观测任务。多颗具有自主规划能力的对地观测卫星通过星际链路连接组成卫星星群,相对于大量的观测任务请求,每颗卫星的观测资源有限,不存在一个中心节点能够获取并统一规划其它卫星的观测资源。卫星之间需要通过交互协商完成规划任务,以期望将观测目标合理分配至各颗卫星,使得总体的观测收益最大。在协同规划过程中涉及以下变量:

(1)给定的规划时段 $w_{\text{schedule}} = [t_S, t_E]$。$t_S$ 表示规划起始时间,t_E 表示规划结束时间。

(2)系统中存在 N_S 能力异构的卫星 Agent,表示为 $SAT = \{sat_1, sat_2, \cdots, sat_{N_S}\}$。$\forall sat_k \in SAT$,$sat_k = \langle R^k_{Vst}, R^k_{Mem}, R^k_{Eng}\rangle$,其中 R^k_{Vst} 表示在规划时段 sat_k 卫星可提供的目标访问时间窗口资源,R^k_{Mem} 为卫星可提供的存储资源,R^k_{Eng} 表示卫星当前可用的能量。

(3) $T = \{t_1, t_2, \cdots, t_{N_T}\}$ 为观测目标集合,$|T| = N_T$。$\forall t_i \in T$ 可表示为 $t_i = \langle u_i, A_i(k)\rangle$,$sat_k \in SAT$,$u_i$ 表示完成 t_i 获得的效用值,$A_i(k) = (A_{i,Vst}(k), A_{i,Mem}(k), A_{i,Eng}(k))$,表示 t_i 对卫星 sat_k 的资源需求向量,由于卫星能力异构,不同卫星对于同一目标 t_i 的资源需求向量不相等。$A_{i,Vst}(k)$、$A_{i,Mem}(k)$、$A_{i,Eng}(k)$分别代表卫星 sat_k 对 t_i 进行观测所要占用的时间窗口资源、存储器容量和能量消耗,目标 t_i 被 sat_k 观测的必要条件是 $A_{i,Vst}(k) \leqslant R^k_{Vst} \wedge A_{i,Mem}(k) \leqslant R^k_{Mem} \wedge A_{i,Eng}(k) \leqslant R^k_{Eng}$。

1.2 建立模型

基于上述问题描述,多星协同任务规划就是卫星进行自主协商,确定每颗卫星的观测目标集合 ST_k,其中 $ST = \bigcup_k ST_k$, $ST \subseteq T$。规划的目标是在各卫星满足其载荷约束的前提下,使得收益最大化。多星协同任务规划的收益通过特征函数 $V(T)$给出,如式(1)所示,其中,$x_i(k)$满足式(2)。

$$V(T) = \sum_{k\in S}\sum_{i\in ST_k} u_i \cdot x_i(k) \tag{1}$$

$$x_i(k) = \begin{cases} 1 & \text{卫星 } sat_k \text{ 观测目标 } t_i \\ 0 & \text{其他} \end{cases} \tag{2}$$

为获得最大的收益,确定最终的目标子集 T^*,即:

$$V(T^*) = \text{Max}(V(T)) \tag{3}$$

约束条件为:

$$\sum_{k\in SAT}\sum_{i\in ST_k} A_{i,Eng}(k) \leqslant R^k_{Eng} \tag{4}$$

$$\sum_{k\in SAT}\sum_{i\in ST_k} A_{i,Mem}(k) \leqslant R^k_{Mem} \tag{5}$$

$\forall t_i, t_j \in T, i \neq j$:

$$r^i_{Vst}(k) \wedge r^j_{Vst}(k) = \varnothing \tag{6}$$

其中,$r^i_{Vst}(k) = [s^k_i, e^k_i]$, $r^i_{Vst}(k) \in R^k_{Vst}$ 表示目标 t_i 占用卫星 s_k 的时间窗口资源,s^k_i 和 e^k_i 分别表示任务的开始、结束时间。

不等式(4)表示观测过程不能违反卫星的能量约束,式(5)限制了卫星观测任务不能超过卫星存储器容量,式(6)保证同一颗卫星的不同任务的观测时间窗口资源不冲突。

2 效用值增益函数定义

分析第 1 节模型可知,卫星集合 SAT 针对任务集的 T 协同规划求解过程是一个分布式约束优化问题(Distributed Constrained Optimization Problem, DCOP)[7]。在多星协同任务规划求解过程中,在卫星集合 SAT 与任务集合 T 组成的环境下,每颗卫星根据当前自身状态以及与环境中其他卫星交互信息,不断迭代“试错”选择最优的规划方案,即是一个多星强化学习的过程。多星强化学习的关键是要结合具体问题设计合理的效用值增益函数。在多星协同任务规划问题中效用的增益函数不但要考虑各卫星所选择的任务集合不能满足约束,而且还需要考虑多星之间任务重复访问。

首先给出多卫星 Agent 强化学习的相关定义。

定义 1 卫星 Agent 的状态 s^k_t:用二元组 $\langle CR^k_t, CT^k_t\rangle$表示 t 时刻卫星 Agent 所处的状态,其中,$CR^k_t = (cr^k_{t,Vst}, cr^k_{t,Mem}, cr^k_{t,Eng})$ 为卫星 Agent 当前

可提供的能力向量,即剩余能力向量; $CT_t^k = \{(t_i, A_{i,Eng}(k)) \mid t_i \in TA_t^k, A_{i,Eng}(k) \in TE_t^k\}$,表示卫星 Agentt 时刻的学习策略片段,其中 $TA_t^k \subseteq T$ 表示卫星 Agent 在 t 时刻受理的任务集合, $TE_t^k = \{A_{i,Eng}(k) \mid t_i \in TA_t^k\}$ 为与 TA_t^k 对应的能量消耗向量。卫星 Agent 所有可能处于的状态 s_t^k 组成的集合称为状态集,表示为

$$S_k = \{s_t^k \mid t = 1, 2, \cdots\}$$

定义 2　卫星的积累效用值 μ_t^k 是指 sat_k 在状态 s_t^k 时已接受的任务效用值之和,表示为:

$$\mu_t^k = \sum_{t_i \in TA_t^k} u_i$$

定义 3　卫星 sat_k 的动作集合 $ACT_k = \{a_t^k \mid a_t^k \in P(T)\}$,其中 $P(T)$表示目标集合 T 的幂集。

定义 4　卫星的更新效用值 ω_t^k 为卫星执行 a_t^k 所包含的任务效用值之和,表示为:

$$\omega_t^k = \sum_{t_i \in a_t^k} u_i$$

卫星 sat_k 在状态 s_t^k 执行规划化动作 $a_t^k \in ACT_k$,转移到下一个状态 s_{t+1}^k, s_{t+1}^k 与 s_t^k 的关系可以表示为:

$$s_{t+1}^k = \langle CR_{t+1}^k, CT_{t+1}^k \rangle \tag{7}$$

$$CR_{t+1}^k = \sum_{t_i \in a_t^k} A_i(k) \tag{8}$$

$$TA_{t+1}^k = a_t^k \tag{9}$$

$$TE_{t+1}^k = \{A_{i,Eng}(k) \mid t_i \in TA_{t+1}^k\} \tag{10}$$

首先考虑单卫星条件下的学习策略,记 $f(s_t^k, a_t^k, s_{t+1}^k)$为卫星 Agent 在状态 s_t^k 下采取行动 a_t^k 转移到下一状态 s_{t+1}^k的效用值增益,根据下式计算:

$$f_k(s_t^k, a_t^k, s_{t+1}^k) = \omega_t^k - \mu_t^k - \eta_t^k \tag{11}$$

其中, η_t^k 为约束惩罚算子,取值为 a_t^k 选择的任务违反卫星约束(4)~(6)的任务效用值乘以惩罚系数 λ 作为惩罚项。如果 a_t^k 未违反任何约束则 η_t^k 为零,显然 η_t^k 总是大于零。学习优化的目标就是在策略空间中搜索一个最优策略,使得效用值增益最大。

考虑多星协同规划条件下的卫星 Agent 学习策略,每颗卫星仍可按照式(11)的效用值增益与自身状态构建独立的卫星学习算法和决策机制。虽然上述的单卫星 Agent 学习算法无需依赖环境模型,但是在每一步迭代学习时都要在动作空间内搜索,当任务规模很大时,将带来“维数灾难”(Curse of Dimension),导致收敛缓慢。更为重要的是,多颗卫星同时在同一动作空间进行搜索,星与星之间的选择动作中很可能会包含重复的任务,违反了目标观测的唯一性约束,造成观测资源的浪费。

针对以上问题,我们基于 Tan[8] 交互学习的思想采取交换学习策略片段的方法。卫星 Agent 在每一步中交互当前学习策略片段,每个 Agent 根据其它卫星的学习策略片段可以对自身的规划策略空间进行扩展,并通过其它卫星的规划策略片段对冲突的任务进行消解。我们设计了多星联合惩罚算子 δ_t^k,具体计算过程如图 1 所示,于是对式(11)进行改进,定义联合效用值增益为

$$f_k(s_t^k, a_t^k, s_{t+1}^k) = \omega_t^k - \mu_t^k - \eta_t^k - \delta_t^k \tag{12}$$

图 1 中, $Epd_{other}^t = \{CT_t^j \mid sat_j \in SAT, j \neq k\}$表示其它卫星当前的学习策略片段集合。

算法名称:联合惩罚算子计算
输入: CT_{t+1}^k, Epd_{other}
输出: δ_t^k 联合惩罚算子
① for each $(t_i, A_{i,Eng}(k)) \in CT_{t+1}^k$
②　　count = 0
③　　do for each $CT_t^j \in Epd_{other}$
④　　　　for each $(t_l, A_{l,Eng}(j)) \in CT_t^j$
⑤　　　　　　do if $t_i = t_l \wedge A_{i,Eng}(k) > A_{l,Eng}(j)$
⑥　　　　　　count←count + 1
⑦　　if count≥⌈(|S| − 1)/4⌉
⑧　　　　$\delta_i^k \leftarrow \delta_i^k - \beta u_i$
⑨　　else if count < ⌈(|S| − 1)/4⌉
⑩　　　　$\delta_i^k \leftarrow \delta_i^k + \beta u_i$
⑪ return δ_t^k

图 1　联合惩罚算子计算流程

Fig.1　Flowchart of computing joint punishment operator

3　多卫星 Agent 强化学习算法

在多星协同规划问题中,各颗卫星之间是完全合作的关系,也就是说在观测目标未被重复访问的前提下,提高任何一颗卫星的规划收益都会使得系统的总收益增加。但是,每颗卫星的能力是有限的且存在差异,需要每个卫星 Agent 通过对环境的学习和交互达到最优的规划效果。

多 Agent-Q 学习算法(Multi-Agent Q-Learning MAQL)是一种模型无关的多智能体强化学习算法。每个 Agent 每步在有限动作集合中按照策略选取一个动作,接受该动作后状态发生转移,通过感知环境的变化给出评价值,从而不断学习逼近

状态 – 行动对的值函数进行问题的求解。而在多星任务规划问题中任务规模较大,若采用 MAQL 求解,由于每个 Agent 的策略不但考虑自身状态,而且需要参考其它 agent 的信息,使得每个 Agent 的状态空间巨大,导致 MAQL 算法将面临“维数灾”。

Wolf-PHC 算法[9]是 MAQL 的一种改进算法,其 Q 值更新过程不直接依赖其他 Agent 的信息,减少了每颗卫星动作空间,并结合多星联合惩罚算子,提高多星的协作能力,改善学习优化性能。学习中,每个卫星 Agent 采用混合策略(Mixed policy)且只保存自身的 Q 值表。所以,它避免了一般 MAQL 中需要解决的探索和利用(Exploration vs. exploitation)这一矛盾问题[10]。另外,与 Nash-Q[11]等算法相比,Wolf-PHC 算法只需保存较少的信息,可相对降低多 Agent 问题求解的空间复杂度。算法中卫星 Agent 的随机策略只与自身状态有关,记为 π_k。初始时每个状态下行动的选择概率都相同,即对 $\forall s_t^k$, a_t^k, $\pi_k = 1/|ACT_k|$, $|ACT_k|$表示卫星可选的动作数。第 t 步学习中,sat_k 的决策更新公式为:

$$\pi_k(s_t^k, a_t^k) = \pi_k(s_t^k, a_t^k) + \begin{cases} \varepsilon_k & a_t^k = \arg\max_{a' \in A_k} Q_k(s_t^k, a') \\ \dfrac{-\varepsilon_k}{|ACT_k| - 1} & \text{其他} \end{cases} \tag{13}$$

其中,ε_k 为学习增量,随着学习的进行呈下降趋势。$Q_k(s_t^k, a')$表示卫星 sat_k 的状态行动对(s_t^k, a')的 Q 值。在 Wolf-PHC 算法中,策略更新采用更理性的方法,即“输快赢慢”(Learning Quickly while losing and slowly while winning)原则。结合 Q 值表,可以判断学习的输或赢,由此决定 ε_k 的取值,即

$$\varepsilon_k = \begin{cases} \varepsilon_{lose} & \text{若“输”} \\ \varepsilon_{win} & \text{其他} \end{cases} \tag{14}$$

其中,当 $\sum_{a_t^k \in A_k} \pi_k(s_t^k, a_t^k) Q_k(s_t^k, a_t^k) > \sum_{a_t^k \in A_k} \bar{\pi}_k(s_t^k, a_t^k) Q_k(s_t^k, a_t^k)$ 时,为“输”。一般情况下,ε_{lose} 比 ε_{win}大若干倍,且均随着学习进行逐步衰减。上式中,$\bar{\pi}_k(s, a)$称为平均策略,其更新公式为:

$$\bar{\pi}_k(s_t^k, a_t^k) = \bar{\pi}_k(s_t^k, a_t^k) + \frac{\pi_k(s_t^k, a_t^k) - \bar{\pi}_k(s_t^k, a_t^k)}{c_k(s_t^k)} \tag{15}$$

式中 $c_k(s_t^k)$为卫星 sat_k 达到状态 s_t^k 的次数。

由前面分析,得到卫星 sat_k 的 Q 学习公式如下:

$$\begin{aligned} &Q_k(s_t^k, a_t^k) \\ &= Q_k(s_t^k, a_t^k) + \alpha[f_k(s_t^k, a_t^k, s_{t+1}^k) \\ &\quad + \gamma \max_{a' \in ACT_k} Q_i(s_{t+1}^k, a') - Q_k(s_t^k, a_t^k)] \end{aligned} \tag{16}$$

其中,$\alpha > 0$ 为学习率,$\gamma(0 \leqslant \gamma < 1)$表示折扣因子。

基于以上分析,图 2 给出多卫星 Agent 强化学习算法的具体步骤:

算法名称:基于多 Agent 强化学习的多星任务规划算法
输入:参与卫星集合 SAT,观测目标集合 T
输出:每颗卫星的策略 π_k,学习策略片段 Epd_k^t
① 根据观测目标集合 T 计算每颗可访问任务集合,令每颗卫星的 Q_k 值为零,随机初始化每颗星的任务选择策略 π_k,令 $t = 0$
② 对于每一步决策时刻 t,令 $k = 0$
③ sat_k 获取其他卫星的学习策略片段 $Epdt_{other}^t$
④ sat_k 根据混合策略 π_k 在 s_t^k 状态下选择规划动作 a_t^k
⑤ 根据式(7) ~ (10)计算状态 s_{t+1}^k,得到观测样本数据 $\langle s_t^k, a_t^k, s_{t+1}^k, \omega_t^k \rangle$
⑥ 根据式(12)计算联合增益效用 $f_k(s_t^k, a_t^k, s_{t+1}^k)$,并根据式(16)更新 $Q_k(s_t^k, a_t^k)$
⑦ 根据式(13)更新 π_k 在状态 s_t^k 时的动作选择概率
⑧ 更新 sat_k 的学习策略片段 Epd_k^t
⑨ $k = k + 1$,若 $k = |SAT| + 1$,则转⑩,否则转③
⑩ 若满足算法终止条件,学习结束。否则 $t = t + 1$,转②

图 2 基于多 Agent 强化学习的多星任务规划算法流程
Fig. 2 Flowchart of multiple satellites task planning algorithm based on multi-satellite reinforcement learning

4 基于黑板结构的多星通信方式

由于基于多 Agent 强化学习的多星协同任务规划过程中需要根据其它卫星 Agent 的学习片段选择规划动作,在实际的多星协同规划运行环境中不同的交互策略对算法的性能有着较大的影响,所以需要设计高效的多星通信方式。通常基于 MAS 的协商通信方式有两种,消息传递机制(Message Passing)和基于黑板结构(Blackboard Architecture)的通信方式。在多星协同规划过程中,为了保证规划结果的优化性,在每一步学习中每个卫星 Agent 根据本地策略进行规划,在此基础上通过彼此交换规划片段信息改进自身的学习策略,是一个反复迭代的过程,若卫星 Agent 以消息传递方式交互信息将会带来很大的通信代价。

而在黑板通信结构中,信息的交流通过中间结构——黑板进行,卫星 Agent 之间无需直接交互通信,只需向黑板更新学习策略片段。其通信结构如图 3 所示,卫星 Agent 从黑板中获取其它卫星的学习策略片段,作为本次规划的参考,并将最新的学习策略片段更新至黑板。

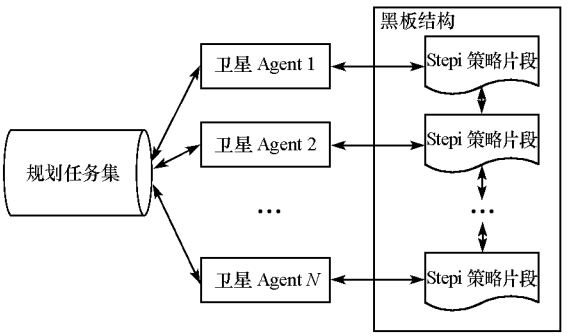

图 3　基于黑板结构的多星通信方式
Fig.3　Communication mode of multiple satellites based on blackboard architecture

5　实验对比及分析

目前卫星规划调度领域尚没有公认的 Benchmark 测试问题集或公开可比较的算法运行结果。为了验证本文算法的适用性和可行性,针对所考虑的卫星,基于卫星实际的不同观测数据进行实验,生成了不同规模、不同任务冲突程度的测试数据集,具体见表 1。在仿真实验中模拟 4 颗卫星参与协同规划,规划时段为 24h。计算平台为 Pentium 2.6GHz CPU,2G RAM,采用 Windows Visual Studio 2005 C # 编码。MUSARLA 算法对各算例进行 20 万步的训练,统计结果为各算例独立测试 30 次的均值。

图 4 给出了 MUSARLA 算法对算例 PH121 的收益优化曲线图,其中 MUSARLA-COP 表示了卫星 Agent 根据式(12)进行合作学习的结果,MUSARLA-SIG 则为卫星 Agent 采用式(11)进行独立学习的结果。从图 4 中可以看出,两种算法都能够得到较优化的结果,与 MUSARLA-SIG 相比,MUSARLA-COP 在卫星 Agent 学习的过程中彼此交互当前的学习策略片段,一方面丰富了本身的学习知识,加快了算法收敛速度;另一方面消解了卫星之间任务冲突,跳出局部最优,提高资源利用率,所以获得了更多的收益。

表 1　测试数据集

Tab.1　Test data set

规划批号	目标数量	任务数量	冲突程度	规划批号	目标数量	任务数量	冲突程度
PH120	30	157	高	PH125	180	955	中
PH121	60	325	中	PH126	210	1266	高
PH122	90	495	中	PH127	240	1244	高
PH123	120	542	低	PH128	270	1613	高
PH124	150	751	中	PH129	300	1573	中

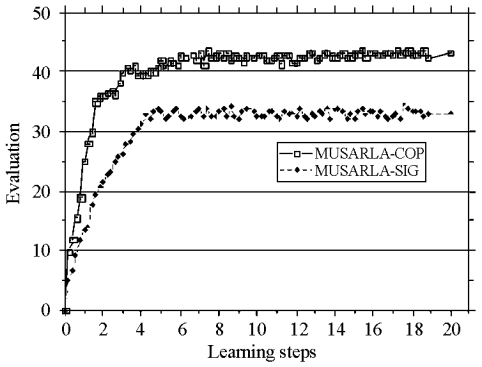

图 4　MUSARLA 收益曲线
Fig.4　Evaluation curve of MUSARLA

为证明算法的有效性,选择基于合同网络协议的多星协同规划算法与 MUSARLA-COP 进行对比。MUSARLA-COP 与 CNP 的算法收益对比如图 5 所示。从图 5 中可以看出,在小规模、低冲突任务条件下,两种算法收益相差较小,随着算例规模和任务冲突程度的增加 MUSARLA-COP 较 CNP 取得了更好的规划收益。主要由于 CNP 算法基于贪婪策略选择任务,而 MUSARLA-COP 算法通过多星之间的交互使得卫星 Agent 能够在更大的范围内搜索问题的优化解,使得各颗卫星能够完成更多的观测目标。

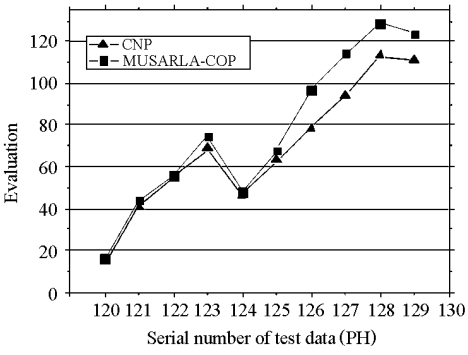

图 5　MUSARLA-COP 与 CNP 规划结果对比
Fig.5　Comparison of planning result

图 6 记录了不同实验算例下 CNP 与 MUSARLA-COP 交互次数比较结果,由于 CNP 算法以招投标的方式分配每一个任务,这导致在协

同规划过程中卫星 Agent 的交互次数对任务规模、卫星数量和任务的冲突程度非常敏感,在大规模高冲突条件下需要卫星之间反复协商。虽然 MUSARLA-COP 也受到任务规模与任务冲突程度的影响,但是通过引入黑板结构的通信方式,有效降低了 MUSARLA-COP 在协同任务规划过程中的通信代价。

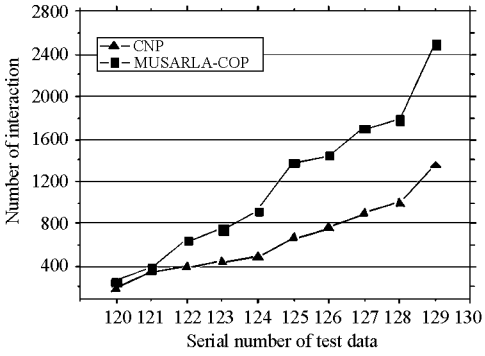

图 6 MUSARLA-COP 与 CNP 交互次数对比
Fig.6 Comparison of the number of interaction

6 结 论

针对多星协同任务规划问题,在分析约束与多星协同特点的基础上给出了面向多星的协同任务规划模型,并采用强化学习算法求解每颗卫星的最优动作选择策略,引入多星联合惩罚算子消解多星之间的任务选择冲突,为减小协同规划过程中的通信代价,设计了基于黑板结构的多星交互方式。通过对比试验分析,表明算法在不同规模和任务冲突程度下以相对小的通信代价取得了较优的规划结果。从而证明该算法充分发挥了各颗卫星的自治性,能够合理利用卫星资源,有效解决多星任务规划问题。

参 考 文 献:

[1] Khatib L, Frank J. Interleaved Observation Execution and Rescheduling on Earth Observing Systems[C]//Proceedings of the 13th International Conference on Automated Planning and Scheduling, Trento, Italy, 2003.

[2] Schetter T, Campbell M, Surka D. Multiple Agent-based Autonomy for Satellite Constellations[J]. Artificial Intelligence, 2003 (145): 147 – 180.

[3] Cesta A, Ocon J, Rasconi R, et al. Simulating On-board Autonomy in a Multi-agent System with Planning and Scheduling[C]//Proceedings of 20th International Conference on Planning and Scheduling, Toronto, Canada, 2010.

[4] 高黎. 对地观测分布式卫星系统任务协作问题研究[D]. 长沙:国防科技大学, 2007.

[5] 陈浩,等. 基于外包合同网的自治电磁探测卫星群任务规划研究[J]. 宇航学报, 2009, 30 (6): 2285 – 2291.

[6] Smith R G, Davis R. Frameworks for Cooperation in Distributed Problem Solving [J]. IEEE Trans. On Systems, Man, and Cybernetics, 1981, 11 (1): 61 – 70.

[7] Modi P J, Shen W, Tambe M, Yokoo M. An Asynchronous Complete Method for Distributed Constraint Optimization[C]//Proceedings of 2nd Autonomous Agent and Multi-agent System, Melbourne, Australia, 2003.

[8] Tan M. Multi-agent Reinforcement Learning: Independent vs. Cooperative Agents [C]//Proceedings of 10th International Conference on Machine Learning, Amherst, MA, 1993: 330 – 337.

[9] Busoniu L, Schutter B D, Babuska R. Learning and Coordination in Dynamic Multiagent Systems[R], Technical Report 05 – 019, Delft Center for Systems and Control, Delft University of Technology, The Netherlands, 2005.

[10] Busoniu L, Schutter B D. A Comprehensive Survey of Multiagent Reinforcement Learning[J]. IEEE Trans. Syst. Man, Cyber., 2008, 38(2): 156 – 172.

[11] Hu J, Wellman M P. Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm [C]//Proceedings of 15th International Conference on Machine Learning, Madison, WI, 1998: 242 – 250.