



一种基于核数分配的任务智能调度方法

刘柳, 凡益民, 刘田, 张琰

A task intelligent scheduling method based on the number of cores

LIU Liu, FAN Yimin, LIU Tian, and ZHANG Yan

在线阅读 View online: <https://doi.org/10.12265/j.ejors.2024115>

您可能感兴趣的其他文章

Articles you may be interested in

基于强化学习的定向无线网络抗干扰资源调度算法

An anti-jamming resource scheduling algorithm for directional wireless communication networks based on reinforcement learning
电波科学学报. 2020, 35(4): 531-541

联合天线选择与用户调度的大规模MIMO系统能效优化算法

Energy-efficient optimization algorithm for massive MIMO systems with joint antenna selection and user scheduling
电波科学学报. 2023, 38(5): 853-860

面向感知任务的无人机数量编配与频谱资源联合规划方法

A joint planning method for the number of UAVs and spectrum resource in perceptual missions
电波科学学报. 2023, 38(5): 764-772

基于智能反射面的无线抗干扰通信方法

Intelligent reflecting surface assisted anti-jamming approach for wireless communications
电波科学学报. 2021, 36(6): 877-886

抛锚状态下的船载地波雷达目标自适应检测方法

Adaptive target detection method of shipborne HFSWR in anchored state
电波科学学报. 2022, 37(2): 251-261

一种基于自适应核字典学习的SAR目标识别方法

SAR target recognition method based on adaptive kernel dictionary learning
电波科学学报. 2019, 34(1): 60-64



关注微信公众号, 获得更多资讯信息

刘柳, 凡益民, 刘田, 等. 一种基于核数分配的任务智能调度方法[J]. 电波科学学报, 2025, 40(2): 246-251+275. DOI: 10.12265/j.cjors.2024115
LIU L, FAN Y M, LIU T, et al. A task intelligent scheduling method based on the number of cores[J]. Chinese journal of radio science, 2025, 40(2): 246-251+275. (in Chinese). DOI: 10.12265/j.cjors.2024115

一种基于核数分配的任务智能调度方法

刘柳^{1,2} 凡益民² 刘田^{2*} 张琰¹

(1. 西安电子科技大学通信工程学院, 西安 710071; 2. 中国电子科技集团公司第十研究所, 成都 610036)

摘要 新一代航天测控与通信地面系统由大规模异构云计算资源池构成。为了给不同的测控通信任务合理分配可用算力, 提出了一种基于核数分配的智能调度方法, 该算法根据每个任务占用核数大小来分配不同 CPU/GPU 进行任务处理。建立了任务核数分配问题的马尔科夫决策过程模型, 根据执行任务数量和可供调度的处理器数量建立 Q 表, 制定各处理器资源负载均衡的奖励值最大化原则, 寻找使调度性能最优的任务分配路径, 逐次计算奖励和更新 Q 值, 完成多周期训练, 根据奖励回路不断迭代优化输出调度结果。实验分析结果表明, 本文算法可以在保证任务均衡部署的前提下最大化计算核的利用率, 相比已有算法提升了多核分配的合理性, 优化了系统整体性能。

关键词 任务; 利用率; 核数; 调度

中图分类号 TP399

文献标志码 A

文章编号 1005-0388(2025)02-0246-06

DOI 10.12265/j.cjors.2024115

A task intelligent scheduling method based on the number of cores

LIU Liu^{1,2} FAN Yimin² LIU Tian^{2*} ZHANG Yan¹

(1. School of Telecommunications Engineering, Xidian University, Xi'an 710071, China; 2. The 10th Research Institute of China Electronics Technology Group Corporation, Chengdu 610036, China)

Abstract The new generation of ground systems for space tracking, telemetry, and command (TT&C) and communication is composed of a large-scale heterogeneous cloud computing resource pool. To allocate available computing power to different TT&C and communication tasks reasonably, an intelligent scheduling method based on core allocation is proposed. This algorithm allocates different CPUs/GPUs for task processing based on the number of cores required by each task. This paper establishes a Markov decision process (MDP) model for the task core allocation problem, constructs a Q -table based on the number of tasks to be executed and the number of available processors for scheduling, formulates a principle of maximizing reward values for load balancing across processor resources, searches for the task allocation path that optimizes scheduling performance, iteratively calculates rewards and updates Q -values, completes multi-cycle training, and continuously iterates and optimizes the output scheduling results based on the reward circuit. Experimental analysis results show that this algorithm can maximize the utilization rate of computing cores while ensuring balanced task deployment. Compared with existing algorithms, it enhances the rationality of multi-core allocation and optimizes the overall system performance.

Keywords task; utilization; number of cores; scheduling

0 引言

随着近年来我国在轨飞行器数量跃居世界前列, 已规划低轨卫星数量过万, 空间平台数量呈几何

倍增长, 天地测控与通信容量需求激增。现有测控基础设施“一站一制”, 与测控对象绑定, 通信制式封闭, 烟囱式建设, 测控资源孤立难以自适应转换和复用。由于陆、海、空、天各域之间电波传播环境差别

巨大,导致测控体制、波形和链路协议各不相同,造成了多模态融合瓶颈,推动了新一代航天地面基础设施逐步向软件可定义、功能可重构的大规模航天测控云架构演进^[1]。

聚合后的测控云化资源池需支持多制式多通道,超大规模处理资源调配困难。因此,如何合理分配可用的计算资源给不同的任务,成为了行业研究的热点问题^[2-5]。

基于马尔可夫决策过程的任务分配算法^[6]是一种基于状态转移概率的任务调度算法,它利用状态转移矩阵来表述任务间转换的可能性。每个任务都有一个当前状态及一个状态转移概率矩阵^[7-8],用来描述任务之间的转移概率。每次进行调度行动时,都会依据当前的状态和状态转移可能性矩阵决定采取哪个行动,且会相应地更新状态转移概率矩阵。文献[9]通过强化学习算法训练调度器,动态地分配任务到可用的资源,从而最小化任务执行时间。文献[10]基于强化学习的任务调度方法,在边缘设备上有效地分配任务,以提高资源利用率和性能。文献[11]用实验数据对节点进行深度学习训练,再用强化学习方法进行资源分配,成功解决了动态资源的调度难题。文献[12]面向工业物联网中异构资源以及动态行为带来的挑战,提出了一个利用强化学习来同时优化能耗和成本的实时调度方法。文献[13]把计算机资源看成是一种组合优化问题,并提出了一种基于强化学习和模仿学习相结合的方法。

上述资源分配方案的模型及方法都较为复杂,需要大量的训练数据来提高学习效果,导致在处理分配调度问题时效率较低。实际应用中任务与资源存在大组合空间搜索难题,传统方法面临训练样本少、调度时效差等问题,迫切需要低复杂度的高效算法满足多任务-多资源动态分配的强实时性。本文提出了一种根据每个任务占的核数大小来分配不同算力的基于马尔可夫决策过程的任务分配算法。该算法可以保证任务均衡部署的前提下,使多核利用率最大化。通过对不同任务占的核数大小的分析,使任务之间的转移概率更加公平,从而提高系统的整体性能。

1 系统模型

1.1 状态转移矩阵和概率

在基于马尔可夫决策过程的任务分配算法中,状态转移矩阵是非常重要的。状态转移矩阵的计算需要考虑到任务之间的转移概率,其可用以下公式表示:

$$P(X_{t+1}|X_t) = P(X_{t+1}|X_t, X_{t-1}, \dots, X_0) \quad (1)$$

若该组变量 $\{X_0, X_1, \dots, X_t\}$ 符合给定的条件概率 P ,它们所构成的集合就被称为马尔可夫链。样本轨迹指由这些随机变量之间连续的转换所构成的路径,即马尔可夫链。

转变的状态转换具有一定的概率,称为转移概率。能够以表格形式展现转移概率的状态转移图,称为马尔可夫表,即它是基于当前状态的转移概率描述。发生的可能性也能够通过公式里的条件概率来展现:

$$P' = P(s_{t+1}|s_t) \quad (2)$$

从状态 s_t 到 s_{t+1} 的条件概率 P' 表示马尔可夫链的关键特性,认为未来的状态只依赖于当前的状态,不受过去历史的影响。

1.2 动作的选择

马尔可夫决策过程是由马尔可夫链增添了智能体的行为和奖励元素,因此转移函数发生了变化,有

$$P' = P(s_{t+1}|s_t, a_t) \quad (3)$$

a_t 在状态 s_t 下将任务分配给适当的处理器。

转移概率是指当前状态下执行某个动作后转向另一种状态的可能性。马尔可夫决策过程的样本轨迹被比喻为马尔可夫链的一种扩展或升级形式:

$$\mathcal{M}_N = \{s_0, a_0, s_1, a_1, r_1, s_2, a_2, r_2, \dots, s_N, a_N, r_N\} \quad (4)$$

它们分别对应着当前的状态 s_N 、正在进行的动作 a_N 以及当前状态 r_N 。这是一种由初始状态 s_0 按照特定策略 $\pi(a|s)$ 驱动,生成与当前状态相关的行动,随之获得反馈奖励的状态-行为-奖励序列。 $\pi(a|s)$ 指的是状态 s 转换成动作 a 的可能性,即:

$$\pi(a|s) = P(a|s) \quad (5)$$

策略生成和状态迁移在马尔可夫决策过程中都带有随机元素,马尔可夫过程实际上能生成众多路径,且每条路径都配有相应的出现概率,例如某个路径 \mathcal{M}_N 的概率为

$$P(\mathcal{M}_N) = P(s_0) \prod_{i=0}^{N-1} P(a_i|s_i) P(s_{i+1}|s_i, a_i) \quad (6)$$

1.3 奖惩值

智能体的目的是让整体路径的奖励和惩罚达到最大,在处理某个情境任务时,将有限且拥有终端状态的轨迹的奖惩值整体进行定义:

$$G = r_1 + r_2 + r_3 + \dots + r_N \quad (7)$$

在从不截止的任务里没有终止阶段,按照之前的公式来核算整个过程的奖励和惩罚就不合适了。当产生的结果是无穷大时,就无法让智能体的奖励

和惩罚总和达到最大。在马尔科夫决策过程里,增添折扣因子 $\gamma \in [0, 1]$,给全程的回馈奖励设定一个折减值。对于智能体,

$$G = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^N r_{N+1} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{i+1} \quad (8)$$

折扣因子影响对将来奖励和即时奖励的看重程度。当折扣因子为0时,重视即时奖励;而当其为1时,倾向于未来奖励。

1.4 状态-动作值函数

策略函数 $\pi(s): S \rightarrow A$ 用来映射状态到行为。策略函数本质上是描述每个状态下应采取的动作,即追求在所有情况下采取最佳行动策略,以便最大程度地收获奖励。

在某种策略下,状态值函数 π 用来评估某个状态优劣的指标,定义为

$$V^\pi = E_\pi[G_i | s_i = s] \quad (9)$$

状态值函数能够计算在指定策略 π 中,从某个状态 s 出发预期能得到的收益,带入 G_i 可以得出:

$$V^\pi = E_\pi \left[\sum_{i=0}^{\infty} \gamma^i r_{i+1} | s_i = s \right] \quad (10)$$

策略的差异会导致状态值函数的变化。

状态-动作值函数(Q 函数)用来评估在既定策略 π 下执行某个动作在特定状态下产生效益优劣的工具,其具体定义是在策略 π 下的状态-动作价值:

$$Q^\pi(s, a) = E_\pi[G_i | s_i = s, a_i = a] \quad (11)$$

即按照策略 π ,从一个状态 s 出发执行行为 a 所能期望得到的收益,带入 G_i 得出:

$$Q^\pi(s, a) = E_\pi \left[\sum_{i=0}^{\infty} \gamma^i r_{i+1} | s_i = s, a_i = a \right] \quad (12)$$

状态值函数评价的是状态的价值,而 Q 函数评估的是特定状态的动作价值。

2 智能调度算法

针对动态测控与通信任务流的传统资源分配常采用基于规则的调度方法,存在着板卡资源碎片多、硬件利用率低、资源分配不合理等问题,本文提出根据测控任务流特征与硬件资源池状态,实时更新搜索的处理核分配策略,通过将优化问题建模成马尔科夫决策过程并迭代求解,实现测控任务流与硬件计算单元间的有效匹配。

贝尔曼方程常被应用于求解马尔科夫决策过程问题,其中最优值函数就是与所有策略下值函数相比能够产生最大值的值函数,即:

$$V^*(s) = \max_{\pi} V^\pi(s) \quad (13)$$

最优策略能够产生出最优值函数 $V^*(s)$,而 Q 函数的最大值也可以用来计算这个最优值函数:

$$V^*(s) = \max_a Q^*(s, a) \quad (14)$$

值函数的贝尔曼方程可表示为

$$V^\pi(s) = \sum_a \pi(s, a) \sum_{s', r} P(s', r | s, a) [r + \gamma V^\pi(s')] \quad (15)$$

Q 函数的贝尔曼方程可以用另一种形式来表述:

$$Q^\pi(s, a) = \sum_{s', r} P(s', r | s, a) \left[r + \gamma \sum_{a'} \pi(a' | s') Q^\pi(s', a') \right] \quad (16)$$

动态规划方法用于求解得到贝尔曼方程。动态规划将复杂问题分解为更简单的子问题,随后求解每一个分支问题并存储答案,遇到重复的问题时不会再次计算,而是直接使用先前得到的结果。

2.1 值迭代和策略迭代

动态规划的实施方式存在两种:值迭代和策略迭代。

如图1所示,值迭代的步骤如下:

- 1) 初始化随机值函数,即每个状态为随机值;
- 2) 计算所有状态-动作的值,即 Q 函数的 $Q(s, a)$ 值;
- 3) 用最大的 $Q(s, a)$ 值对值函数进行更新;
- 4) 一直重复这个过程,直到值函数达到最优,即收敛。

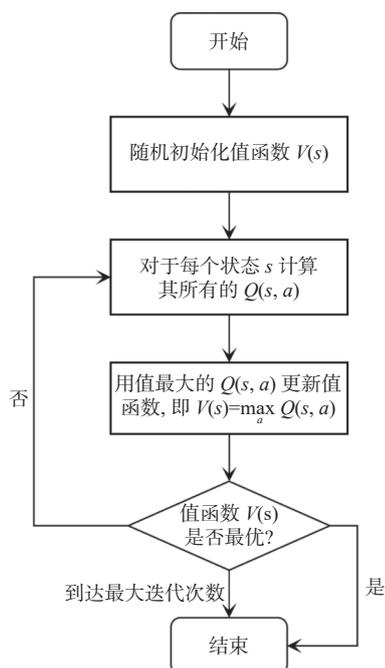


图1 值迭代流程

Fig. 1 Value iteration flowchart

如图 2 所示,策略迭代的步骤如下:

- 1) 随机初始化一批策略;
- 2) 评估通过随机策略得到的值函数是否最优,即策略评估;
- 3) 对非最优策略进行优化提升;
- 4) 不断重复上述步骤,直到更新后的策略和更新策略一样,即找到最优策略。

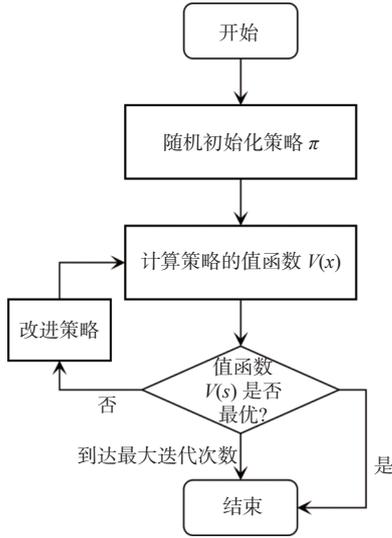


图 2 策略迭代流程

Fig. 2 Policy iteration flowchart

如图 3 所示,按照贝尔曼动态规划理念,智能体在当前状态下能够获得的时间累计折扣性奖励 R_t 可以这样表述:

$$R_t = r_{t+1} + \gamma r_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (17)$$

式中, r_{t+1} 为当前状态下的即时收益,对不同时间点的奖励给予重要性评估。智能体在给定策略 π 下,在特定时间点 t 处于某个状态 s_t 下执行某动作 a_t 后获得奖励的期望值被称为动作值函数 $Q(s_t, a_t)$:

$$Q(s_t, a_t) = E_{\pi}(R_t | s_t, a_t) = E_{\pi}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots | s_t, a_t) \quad (18)$$

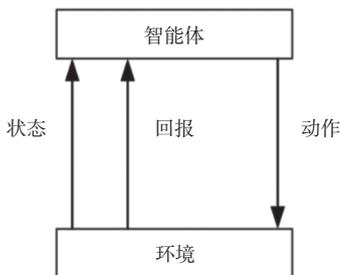


图 3 强化学习过程

Fig. 3 Reinforcement learning process

依据给定的状态转移规律,可得动作值函数的贝尔曼方程形式,即:

$$Q(s_t, a_t) = r_t + \gamma \sum_{s_{t+1}} P(s_{t+1} | s_t, a_t) \sum_{a_{t+1}} \pi(a_{t+1} | s_{t+1}) Q_{\pi}(s_{t+1}, a_{t+1}) \quad (19)$$

进一步可得到动作值函数的最优贝尔曼方程:

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) \quad (20)$$

根据时序差分学习原则,动作值函数的更新公式可以写为

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha [r_{t+1} + \gamma \max_{a_{t+1} \in A(s_{t+1})} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (21)$$

其选择动作的策略依据为

$$\pi(s) = \arg \max_{a \in A(s)} Q(s, a) \quad (22)$$

2.2 算法执行步骤

整个核数分配算法可以概括为:系统根据执行任务数量和可供调度的处理器数量建立 Q 表;制定各处理器资源均衡分配得到最大化奖励值的原则,寻找使调度性能最优的任务分配路径;逐次计算奖励和更新 Q 值,完成相应训练周期;根据奖励值不断迭代优化输出调度结果。图 4 为算法的流程图及核心步骤。

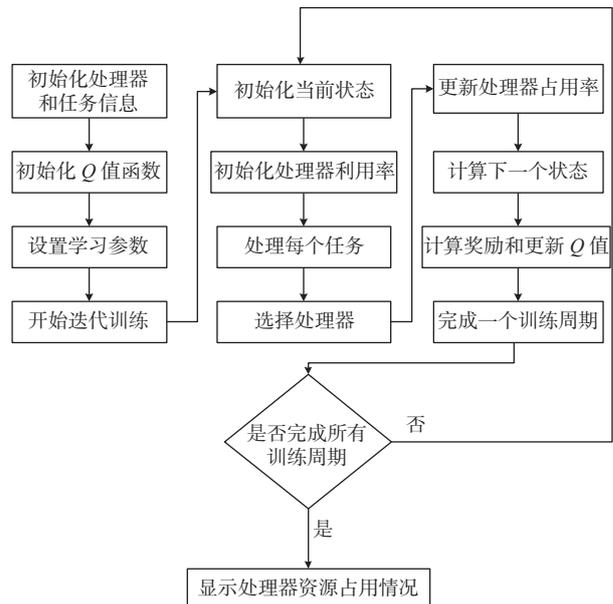


图 4 调度算法流程图

Fig. 4 Scheduling algorithm flowchart

步骤 1 模型初始化

1) 确定处理器个数及处理能力(以核数为单位);输入任务流个数及每个任务占用的资源数即核数。

2) 初始化 Q 值函数,创建 Q 值函数矩阵,该矩阵行的个数等于任务的个数,列的个数等于处理器的个数。

3) 设置学习率、探索率和迭代次数,并根据式(8)设置折扣因子。

步骤2 调度策略

在每次迭代中,从任务流中第一个任务开始,对于每个任务,选择处理器处理的策略有两种:

1) 根据式(14),以当前状态的 Q 值选择最优执行的处理器。

2) 根据设置的探索率,随机选择处理器,全局探索可执行任务的处理器,不会使算法陷入局部最优解。

两种策略按照式(5)确定合适的比例可以保证算法在一定程度上进行探索和利用。

步骤3 执行强化学习

1) 根据调度策略选择最优的动作后,将任务初次分配给相应的处理器,同时计算当次处理器资源利用率,实时更新初次分配后的处理器占用率。

2) 在分配一个任务后,根据所有处理器的资源利用率和当前任务选择的处理器以及考虑未来任务的潜在奖励,利用学习率、折扣因子来综合计算奖励,并更新式(21)中的 Q 值函数。

步骤4 输出调度结果

在迭代次数完成后,使用学习到的 Q 值函数根据式(22)对每个任务进行调度分配,每个处理器上任务对应的 Q 值最大,即为需调度的处理器位置。

步骤5 分配性能评估

在完成调度结果后,输出各个处理器的资源占用情况和 Q 值矩阵在平均值和 Frobenius 范数上的变化量迭代趋势,研判算法调度性能。

3 仿真结果与分析

探索率设置为 0.115,即学习过程中有 11.5% 的概率随机选择一个动作来探索环境,使算法在运行相同迭代次数后产生不同的结果,增加全局探索性。设置学习率为 0.15,折扣因子为 0.9,迭代次数为 900。

输入 15 个核数占比不同的测控任务,每个任务占用的核数为 8~14。供调度的处理器有 5 个,每个处理器能支持的最大核数为 40~80。表 1 为 3 次算法运行后的输出结果,包含每个任务的调度结果。

根据表 1,在探索因子的存在下,每次迭代到相同次数时,会产生不同的调度方案,3 次运行结果见表 2。可以看出,虽然 3 种调度方案分配结果不同,但每个处理器的资源占用情况都达到了均衡配置,并不存在某个处理器过于空闲或忙碌。

选取 Q 值矩阵的平均值和 Frobenius 范数作为收敛变化表征指标,如图 5 所示。可以发现在 900 次迭代过程中,由于探索因子的存在, Q 值矩阵的平均值和 Frobenius 范数变量在收敛过程中存在一定的抖

表 1 15 个输入任务对应的调度结果

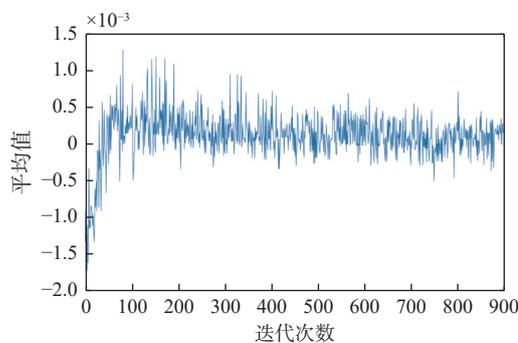
Tab. 1 15 tasks corresponding scheduling results

输入任务	调度方案1	调度方案2	调度方案3
1	1	1	5
2	5	2	1
3	1	1	2
4	5	5	4
5	4	5	5
6	1	3	3
7	2	3	1
8	2	4	3
9	3	4	2
10	4	3	4
11	3	1	3
12	4	2	5
13	1	1	1
14	5	2	2
15	4	5	4

表 2 不同方案各处理器资源占用情况

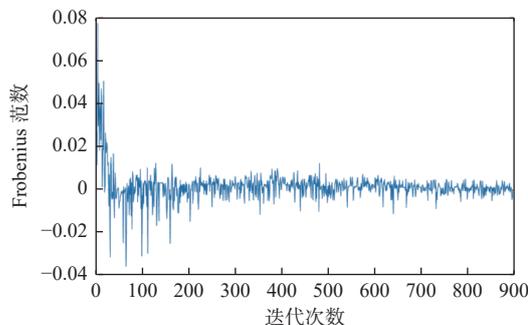
Tab. 2 Resource consumption of different schemes for each processor

处理器	调度方案1	调度方案2	调度方案3
1	46.666 7	50.000 0	43.333 3
2	36.666 7	43.333 3	46.666 7
3	40.000 0	40.000 0	50.000 0
4	56.666 7	43.333 3	46.666 7
5	50.000 0	53.333 3	43.333 3



(a) 平均值变化量

(a) The average change in value



(b) Frobenius 范数变化量

(b) The change in the Frobenius norm

图 5 Q 值矩阵的平均值和 Frobenius 范数变化

Fig. 5 The average change in the Q -value matrix and the Frobenius norm

动;随着迭代次数增大后, Q 值矩阵的最终整体变化趋势都会近似收敛到 0,达到各处理器资源均衡分配的目标。

引入文献 [14] 的合作学习以及随机分配方法与本文算法进行对比,结果如图 6 所示。可以看到,本文算法在完成任务分配后,所得到的资源占用曲线较为平缓,资源占用率最大与最小变化在 10% 以内,而其他两种方法在不同处理器的资源占用率起伏较大,占用率最大与最小变化远超过 10%,未达到较好的资源分配结果。

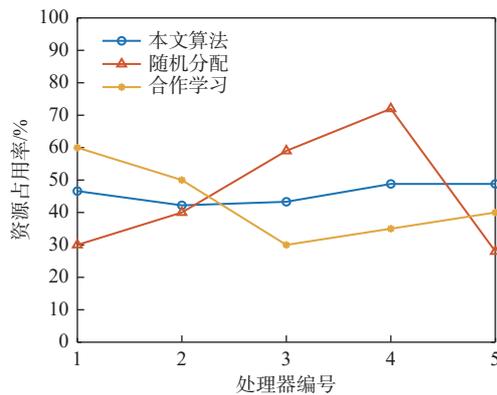


图 6 不同算法的资源占用率

Fig. 6 Resource utilization curves of different algorithms

4 结 论

本文提出了一种基于每个测控任务所占核数大小来确定 CPU/GPU 等不同处理器的任务分配算法。在保证任务均衡部署的前提下,该算法可使多处理器利用率最大化。通过对不同任务占核数大小的分析发现,该算法在测控应用中具有很好的效果,具有很好的扩展性,且使任务分配更加合理,从而提高测控系统的整体性能。

参考文献

- [1] 李超,焦义文,傅诗媛,等. 软件定义测控系统体系架构与关键技术[J]. 中国空间科学技术, 2023, 43(3): 14-24.
- [2] ZHAO N, YE Z, PEI Y, et al. Multi-agent deep reinforcement learning for task offloading in UAV-assisted mobile edge computing[J]. *IEEE transactions on wireless communications*, 2022, 21(9): 6949-6960.
- [3] ZHAO N, LIU Z, CHENG Y. Multi-agent deep reinforcement learning for trajectory design and power allocation in

multi-UAV networks[J]. *IEEE access*, 2020, 8: 139670-139679.

- [4] DING R, GAO F, SHEN X S. 3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: a deep reinforcement learning approach[J]. *IEEE transactions on wireless communications*, 2020, 19(12): 7796-7809.
- [5] 李冠雄,李桂林. 基于强化学习的合作频谱分配算法[J]. *电波科学学报*, 2022, 37(1): 8-14.
- LI G X, LI G L. Cooperative spectrum allocation algorithm based on reinforcement learning[J]. *Chinese journal of radio science*, 2022, 37(1): 8-14. (in Chinese)
- [6] WANG C, DENG D, XU L, et al. Resource scheduling based on deep reinforcement learning in UAV assisted emergency communication networks[J]. *IEEE transactions on communications*, 2022, 70(6): 3834-3848.
- [7] JIANG F, WANG K, DONG L, et al. Stacked autoencoder-based deep reinforcement learning for online resource scheduling in large-scale MEC networks[J]. *IEEE Internet of Things journal*, 2020, 7(10): 9278-9290.
- [8] KUAI Z, WANG T, WANG S. Fair virtual network function mapping and scheduling using proximal policy optimization[J]. *IEEE transactions on communications*, 2022, 70(11): 7434-7445.
- [9] ZHANG Q, SUN B, ZHANG J. A survey of reinforcement learning-based scheduling in cloud computing[J]. *IEEE access*, 2021, 9(1): 587-602.
- [10] MAO H, ALIZADEH M, MENACHE I, et al. Deep reinforcement learning for job scheduling in data centers[C]// *Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2016.
- [11] YU W, LIANG F, HE H, et al. Deep reinforcement learning-based task scheduling for edge computing[J]. *IEEE Internet of Things journal*, 2018, 18(8): 1861-1874.
- [12] MA X, XU H, GAO H, et al. Real-time virtual machine scheduling in industry IoT network: a reinforcement learning method[J]. *IEEE transactions on industrial informatics*, 2022, 19(2): 2129-2139.
- [13] GUO W, TIAN W, YE Y, et al. Cloud resource scheduling with deep reinforcement learning and imitation learning[J]. *IEEE Internet of Things journal*, 2020, 8(5): 3576-3586.
- [14] 梁智韬,贾高伟,王建峰. 无人机集群任务分配中群智能算法性能对比[J]. *火力与指挥控制*, 2022, 47(6): 28-37.
- LIANG Z T, JIA G W, WANG J F. The comparison of swarm intelligence algorithm about the mission assignment of UAV swarm[J]. *Fire control & command control*, 2022, 47(6): 28-37. (in Chinese)

tional Conference on Signal, Information and Data Processing (ICSIDP). Chongqing: IEEE, 2019: 1-5.

[18] 王雪松. 现代雷达电子战系统建模与仿真[M]. 北京: 电子工业出版社, 2010.

[19] 李杰然. 基于信息熵的灵巧噪声干扰识别方法[J]. 舰船电子工程, 2021, 41(12): 32-35.

LI J R. Method of smart noise jamming recognition based on Shannon entropy[J]. *Ship electronic engineering*, 2021, 41(12): 32-35. (in Chinese)

[20] 李文杰, 闫世强, 蒋莹, 等. 自适应确定 DBSCAN 算法参数的算法研究[J]. 计算机工程与应用, 2019, 55(5): 1-7+148.

LI W J, YAN S Q, JIANG Y, et al. Research on method of self-adaptive determination of DBSCAN algorithm parameters[J]. *Computer engineering and applications*, 2019, 55(5): 1-7+148. (in Chinese)

作者简介

关中意 (1999—), 男, 陕西人, 西安电子科技大学雷达信号处理全国重点实验室硕士研究生, 研究方向为干扰感知、信号处理。E-mail: gzy_1999@foxmail.com

兰岚 (1993—), 女, 陕西人, 西安电子科技大学雷达信号处理全国重点实验室副教授, 硕士生导师, 主要研究方向为波形分集阵列、雷达抗干扰方法、MIMO 雷达技术。E-mail: lanlan@xidian.edu.cn

朱圣棋 (1984—), 男, 江西人, 西安电子科技大学雷达信号处理全国重点实验室教授, 博士生导师, 主要研究方向为雷达运动目标检测、频率分集阵列、波形分集阵列、雷达信号处理。E-mail: zhushengqi8@163.com

(上接第 251 页)

作者简介

刘柳 (1990—), 女, 中国电子科技集团公司第十研究所工程师, 博士研究生, 研究方向为测控通信技术。E-mail: liu_swjtu@126.com

凡益民 (1996—), 男, 中国电子科技集团公司第十研究所工程师, 硕士, 研究方向为测控通信技术。E-mail: 420764441@qq.com

刘田 (1981—), 男, 中国电子科技集团公司第十研究所研究员, 博士, 研究方向为测控通信技术。E-mail: liutian139@139.com

张琰 (1983—), 男, 西安电子科技大学教授, 博士, 研究方向为通信与信息系统、无线自组通信技术。E-mail: yanzhang@xidian.edu.cn