计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220647
60(11): 2650−2659, 2023

一种考虑隐私保护的深度强化学习任务分配模型

杨明川[1] 朱敬华[1,2] 李元婧[1] 奚赫然[1,2]
[1](黑龙江大学计算机科学与技术学院 哈尔滨 150006)
[2](数据库与并行计算重点实验室(黑龙江大学) 哈尔滨 150006)
(mrvincenty@163.com)

Task Allocation Model Based on Deep Reinforcement Learning Considering Privacy Protection

Yang Mingchuan[1], Zhu Jinghua[1,2], Li Yuanjing[1], and Xi Heran[1,2]
[1](*School of Computer Science and Technology*, *Heilongjiang University*, *Harbin* 150006)
[2](*Key Lab of Database and Parallel Computing* (*Heilongjiang University*), *Harbin* 150006)

Abstract Mobile crowdsensing (MCS) is a new mode for collecting and mining data and intelligent decision-making with mobile intelligent devices. The key to the high performance of MCS is the efficient method of task allocation. The traditional algorithm (greedy algorithm or ant algorithm) assumes that workers and tasks are static. It's not fit for the scene where the position and time of workers and tasks change continuously. In addition, the existing methods usually make decisions by the central server based on the collected information, which usually leads to leakage of workers' privacy. Therefore, we propose a task allocation method based on deep reinforcement learning (DRL) with privacy protection. Firstly, aiming to maximize the two-way benefits of workers and platforms and realize Nash equilibrium, the task allocation is modeled as a dynamic programming problem of multi-objective optimization. Secondly, the model based on proximal policy optimization (PPO) of DRL for training and learning model parameters is proposed. Finally, we use the local differential privacy method to add random noise to the sensitive information of workers to protect privacy. The central server trains the whole model to obtain the optimal allocation strategy. In this paper, the astringency, revenue and task cover rate are experimentally evaluated. The results show that the proposed method has significant improvement in different indexes, and can protect the privacy of workers, compared with the traditional methods and other DRL based methods.

Key words mobile crowdsensing; task allocation; deep reinforcement learning; local differential privacy; dynamic programming

摘 要 移动群智感知(mobile crowdsensing, MCS)是利用大规模移动智能设备进行数据收集、数据挖掘和智能决策的新范式,高效的任务分配方法是MCS获得高性能的关键.传统的贪婪算法或蚂蚁算法假设工人和任务固定,不适用于工人和任务的位置、数量和时间动态变化的场景.而且,现有任务分配方法通常由中央服务器收集工人和任务的信息进行决策,容易导致工人隐私泄露.因此,提出具有隐私保护的深度强化学习(deep reinforcement learning, DRL)模型来获得优化的任务分配策略.首先,将任务分配建模为多目标优化的动态规划问题,旨在最大化工人和平台的双向收益,实现纳什均衡.其次,提出基于DRL的近端策略优化(proximal policy optimization, PPO)模型进行训练,学习模型参数.最后,通过本地差分隐

收稿日期: 2022-07-24; **修回日期**: 2022-11-04
基金项目: 黑龙江省自然科学基金-联合引导项目 (LH2022F045)
This work was supported by the Natural Science Foundation of Heilongjiang Province-Joint Guidance Project(LH2022F045).
通信作者: 朱敬华(zhujinghua@hlju.edu.cn)

私方式，对工人位置等敏感信息加入随机噪声实现隐私保护，并由中央服务器训练整个模型，获得最优分配策略. 对收敛时间、最大收益和任务覆盖率等指标进行实验评估，在模拟数据集上的实验结果表明，与传统方法和其他基于 DRL 的方法对比，该方法在不同的评估指标上均有明显提升，并且能够保护工人的隐私.

关键词 移动群智感知；任务分配；深度强化学习；本地差分隐私；动态规划

中图法分类号 TP391

移动群智感知(mobile crowdsensing, MCS)是 Ganti 等人[1]提出的，是一种在用户或社区之间感知和共享数据的新方法. Guo 等人[2]给 MCS 更为明确的定义："MCS 是一种新的感知范式，使普通公民能够贡献从移动设备感知或生成的数据，聚合和融合云中的数据，用于人群智能提取和以人为中心的服务交付."[3]随着高性能的便携式移动设备与高速智能网络的普及，移动群智感知技术快速发展并深入到智慧医疗、交通流量预测以及智慧城市等各个领域，需要采集处理的数据集也日渐庞大，MCS 系统需要大量用户的参与和贡献[4]，如何在数量庞大的参与者中选择合适的参与者完成给定的感知计算任务，且最大化平台和用户的收益显得格外重要. 任务分配系统主要由 3 部分组成：平台(任务发布者)、工人(用户，携带移动智能设备负责采集感知数据)和任务(如收集某地区的空气质量数据、监测某路段的交通流量等). 如图 1 所示，任务由平台基于一定的收益计算机制分配给工人，然后工人利用移动智能设备到指定任务点进行相关感知数据的收集并上传到平台获取相应报酬. 一方面，考虑到任务的时效性以及预算，任务应当被合理地分配给合适的工人，保证任务分配合理化的同时尽可能最大化平台总收益；另一方面，工人上传信息时通常无法避免暴露自身位置等隐私信息. 因此，在 MCS 系统中，合理的任务分配机制与工人信息的隐私保护问题尤为重要. 传统的任务分配算法，如蚂蚁算法、贪婪算法，适合于小规模数据集，应用于工人与任务信息固定的静态系统，但实际问题中工人与任务的位置、状态信息会不断改变，因此，深度学习被越来越多的研究者引入到这样的动态系统中来解决相应的动态规划问题.

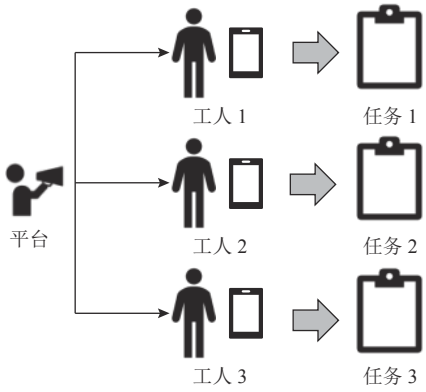


Fig. 1 Diagram of MCS task assignment system

图 1 MCS 任务分配系统图示

深度强化学习(deep reinforcement learning, DRL)可以基于过去的经验，通过智能体选择动作与环境交互并获得相应的状态和回报[5]，在每次进行决策的过程中，智能体的策略选择的概率分布不断调整，最终达到最优的全局策略. 因此在动态的 MCS 问题中，DRL 往往能发挥更好的性能. 在 DRL 的众多方法中，DQN(deep Q-network)[6]和 A3C(asynchronous advantage actor-critic)[7]可以表现出良好性能，但仅限在离散的动作空间中；DDPG(deep deterministic policy gradient)[8]是一种离线的、确定性的方法，相对不适合需要实时控制解决方案的动态场景；TRPO(trust region policy optimization)[9]采用信任区域方法，其性能优于许多随机在线策略梯度方法，更适合于需要更多探索的场景；PPO(proximal policy optimization)[10]是一种无模型的、基于策略的、基于梯度的强化学习方法，它在连续控制问题的表现极其优异，并具有 TRPO 的相应优点且实现复杂性要低得多. 本文的算法采用了 PPO 框架，该框架可以更好地适配离散型和连续型的状态/动作集合，甚至在复合型的状态/动作集合的表现也较为良好，并且相对于其他的 DRL 方法，PPO 的表现也较为优异，具有更快的收敛性.

同时，考虑到工人在与平台进行数据交互时往往会暴露自己的移动轨迹信息，因而本文采用本地差分隐私在工人与平台的交互中进行隐私保护. 差分隐私的概念最早由 Dwork[11]提出，建立在严格的数学理论基础上，对隐私保护提供了量化的评估方法和严谨的数学证明. 本文方法在平台与工人的信息交互中，利用本地差分隐私的方法，对其位置信息加入随机噪声，最大限度地保护工人的隐私信息.

本文是面向 MCS 任务分配问题，使用 DRL 与差分隐私方法在保护隐私的前提下获得优化的任务分配策略. 将动态环境下的任务分配问题定义为一个

基于离散型数据集来进行动态规划的优化问题，并使用基于 DRL 的算法来解决动态环境下的任务分配问题. 具体来说，在每次迭代开始时，该算法观察了之前迭代中平台的分配策略、平台收益、工人收益、现有任务信息以及工人信息. 根据观察结果，由基于 DRL 的算法来决定工人分配到的任务以及任务的顺序，在此过程中利用差分隐私来对工人相关隐私信息做了模糊化处理. 本文的目标是最大化平台的总收益和工人收益，被定义为工人收益与平台总收益的联合约束问题，此外还考虑了隐私保护的相关问题. 本文的主要贡献总结为 4 个方面：

1)将 MCS 动态场景下的任务分配问题建模为一个多目标优化问题，并证明为 NP-hard 问题. 充分考虑了在 MCS 的任务分配问题中，工人与任务状态信息不断变化的动态系统，以及工人与平台进行数据交互的隐私保护的必要性.

2)提出基于 DRL 的 PPO 方法求解该优化问题. 相比于传统方法，DRL 在中大型数据集的 MCS 问题中表现性能良好、收敛性好，能更快达到最优解，同时考虑了真实的 MCS 中工人和任务的动态性，利用 DRL 方法更适用于解决此类动态的、非确定性的 MCS 问题.

3)提出基于差分隐私的任务分配方法. 在工人的智能移动设备与中央服务器交互中利用本地差分隐私的方法，对工人的位置信息加入随机噪声，模糊化工人位置信息，进而解决中央服务器收集工人信息时存在的隐私泄露问题.

4)通过实验评估本文方法的有效性和高性能. 通过模拟数据集的实验，对比了传统方法与现有方法，验证了本文的模型具有稳定性能，且收敛效果较好；此外还进行了消融实验，证明了加入隐私保护方法的有效性.

1 相关工作

1.1 传统任务分配方法

Cheung 等人[12]研究了时间敏感和位置依赖的感知任务的分配问题. 考虑到具有不同初始位置、移动成本和速度以及声誉级别的异构用户，提出了一种贪婪算法来计算该问题的近似解. 该算法要求每个用户专注于自己的收益，并向用户提供一个异步的分布式算法来计算用户的移动性计划. 该算法的设计目标是最大化用户收益，但无法适用于工人状态信息变化的动态系统. 在文献 [13] 中，Li 等人提出了基于蚁群算法 ACO(ant colony optimization)的启发式多任务调度算法来确定任务调度策略，对工人福利的计算模型进行理论分析，利用基于 ACO 的启发式多任务调度算法来确定任务调度策略，以最大限度地提高工人的利益. 但该方法同样仅适用于静态系统，对于动态系统，越来越多的研究者倾向于基于 DRL 的算法[14-15].

1.2 基于 DRL 的算法

2013 年谷歌的 DeepMind 团队发表了利用强化学习玩 Atari 游戏的文章[16]，DRL 开始炙手可热，相关算法被更多的研究者引入到各个领域. Kim 等人[17]将 DRL 的方法应用于无人机的任务分配问题上，用基于 Q 值的深度强化学习算法 DQN 来实现快速策略收敛，从而有可能适用于更大规模的系统，进而解决难以量化的由随机环境引起的无人机移动性的随机性问题. Tao 等人[18]使用双深度 Q 网络(double deep Q-network, DDQN)来解决任务分配问题，作为一个具有时间窗的路径规划问题，考虑了感知任务的位置依赖性和时间敏感性，以及工人在最大旅行距离方面的资源限制. 在文献 [19] 中，Patel 等人针对联邦环境下计算资源分配问题，提出了一个旨在使系统总成本最小化的优化问题，并将其定义为训练时间和能量消耗的加权和. 考虑到非线性约束的难度和网络质量的不稳定，该团队设计了一种基于 DRL 的经验驱动算法，该算法可以在不了解网络质量的情况下收敛到接近最优解.

1.3 差分隐私

在文献 [11] 中，差分隐私的概念被 Dwork 首次提出，该文章通过严谨的数学证明，可以保证数据变化时用户隐私不受攻击者所知的背景知识的影响. 之后 Dwork 对原有差分隐私的概念进行改进，提出本地化差分隐私[20]，将信息的隐私处理工作转移到用户端，对差分隐私进行量化，每个用户单独对敏感数据进行处理，使得隐私保护更为彻底. Chen 等人[21]将本地差分隐私应用于位置数据保护，通过对位置数据加入拉普拉斯噪声，实现对位置数据的隐私保护. Wang 等人[22]提出了一种基于差分隐私以及 Hilbert 曲线的位置保护方法，将位置映射到一维空间中，通过拉普拉斯噪声对位置信息进行扰动，将处理后的位置信息发送给平台来实现位置信息保护.

2 问题定义

假设在该系统中有 n 个携带智能移动设备的工

人 $W=\{w_1, w_2, \cdots, w_n\}$，$m$ 个任务 $V=\{v_1, v_2, \cdots, v_m\}$. 进而工人的智能移动设备可用 $Dev=\{Dev_1, Dev_2, \cdots, Dev_n\}$ 表示，并定义第 i 个工人由 $w_i=\{P_{w_i}, V_{w_i}, x_i, y_i\}$表示，第 j 个任务由 $v_j=\{t_{v_j}, r_{v_j}, x_j, y_j, r_{p_j}\}$表示，其中$(x_i, y_i)$和$(x_j, y_j)$表示坐标，$P_{w_i}$是第 i 个工人该时刻所有任务的开销集合，V_{w_i}是该工人当前被分配到的任务队列，t_{v_j}表示第 j 个任务所需时间，r_{v_j}表示该任务的奖励，r_{p_j}表示完成该任务平台可获取的收益. 该系统是动态的，即工人与任务状态位置信息不断改变，在不同时刻下工人完成已有任务后会有“闲置状态”，此时需要在每次迭代时将“闲置”的工人重新放入在“待分配”工人的队列里，同时每个工人可接受的任务也是有限的，这需要根据工人的报酬以及任务对于工人的收益进行约束，例如距离较远的任务的开销大于收益则不会分配给工人，进而间接限制了工人所接受任务的数量，这就避免了任务分配不均的问题.

工人完成任务是有时效性的，因此在每个工人与任务里加入了时间戳，记录工人完成任务的时间，并标注出每个任务的完成时间限制. 此外，考虑到工人开销的差异性，即任务对于每个工人的开销应该是不一样的，因而加入了笛卡儿坐标系，为每个工人和任务设置了位置坐标，每个工人依据距离和自己未完成的任务量计算任务开销. 例如对于第 i 个工人，计算第 m 个任务开销时，距离越远，任务开销越大，自身未完成任务量(加权后的任务数量)越多，任务开销越大，反之亦然. 这里，定义第 i 个工人对于第 j 个任务的开销为:

$$P_{w_i}^j = \zeta \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2} + \alpha\theta_i, \tag{1}$$

$$\theta_i = \sum_{v_j \in V_{w_i}} t_{v_j}, \tag{2}$$

其中 θ_i 表示该工人的完成任务所需总时间，ξ 表示第 i 个工人到第 j 个任务点的欧氏距离权重，α 为时间权重，这里体现出了每个工人的任务开销的差异性.

可定义第 i 个工人的收益以及平台总收益为

$$r_i = \sum_{v_j \in V_{w_i}} (r_{v_j} - P_{w_i}^j), \tag{3}$$

$$R_p = \sum_{v_j \in V_{\text{out}}} (r_{p_j} - r_{v_j}), \tag{4}$$

其中 r_i 表示第 i 个工人的收益，R_p 表示此时平台总收益，V_{out} 表示此时所有被分配出去的任务集合.

最终，将这一个基于离散型数据集的动态策略优化问题定义为

$$\max\left\{\lambda_1 R_p + \lambda_2 r_i\right\}, \tag{5}$$

$$\text{s.t. } 0 < P_{\min} \leqslant P_{w_i}^j, \forall w_i \in W, \tag{6}$$

其中式(5)代表最大化平台总收益同时也要最大化当前第 i 个工人的收益，λ_1 与 λ_2 为两者的收益权重，在式(6)中的约束代表第 i 个工人完成第 j 个任务时的报酬要保证不小于最小值 $P_{\min}$，且 $P_{\min}$ 是一个大于 0 的常量.

从式(3)(4)中可以看出平台总收益与工人的收益是负相关的，而在本文的问题中，更希望两者都能达到最大化，因此采用联合优化方式，通过调节权重值来平衡双方利益，实现双方的纳什均衡.

由上述的问题定义可证明 MCS 的任务分配问题是一个 NP-hard 问题. 首先假设一个特殊情况，即只有一个工人，任务集合不变. 然后，该工人有一个设定的最大旅行距离，且支付给工人的报酬设置为 0. 最后，平台的总利润等于该工人完成任务的报酬，这也映射到定向运动问题，且该问题已被证明为 NP-hard 问题[23]，则可推论本文的问题同样是 NP-hard 问题.

3 系统模型及算法

3.1 系统模型概述

本文的系统模型如图 2 所示，任务发布者在模型中作为中央服务器，每个工人的智能移动设备可看作分布式的小型处理器. 在整个系统中，中央服务器与各个工人的移动设备在隐私保护的环境中进行信息交互. 首先，每个工人智能移动设备将相关信息经过差分隐私的处理后传至中央服务器. 之后，中央服务器获取该时刻全局的工人与任务的状态信息，经

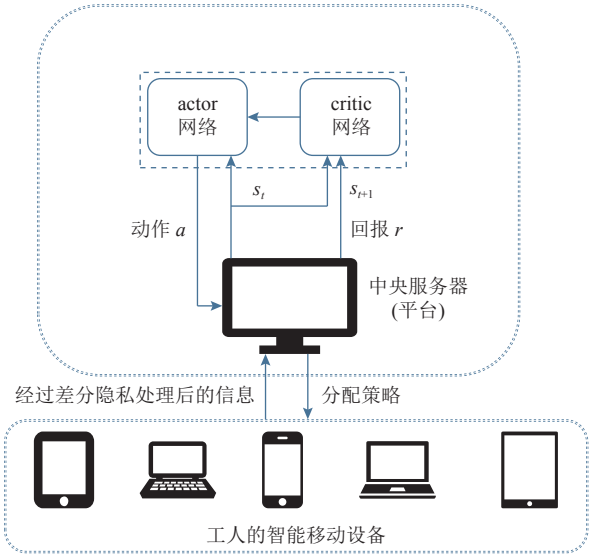


Fig. 2 Illustration of the system model

图 2 系统模型图示

过基于 DRL 的动态策略优化算法，制定相应的分配策略，最终传给各工人智能移动设备. 同时在交互过程中，中央服务器中采用 PPO 的算法进行训练并决策，在每轮训练时考虑了系统的动态性与差异性问题，即不同工人对于相同的任务会受到距离以及未完成的任务量影响，进而每个任务对不同工人的开销应是不一样的，且在每轮迭代时可能会产生完成了当前任务的“空闲”工人，在模型定义中考虑了以上问题，在每轮迭代会动态更新全局信息，最终得到全局最优的策略.

3.2 PPO 模型

中央服务器中应用了基于 DRL 的动态策略优化算法 PPO 进行决策推演. 传统的 PPO 算法源于 A-C 网络[24] 的思想，如图 3 所示，该算法由 actor 网络和 critic 网络构成，每一次迭代时，actor 网络会根据一定的动作决策概率分布进行动作选择，并与环境进行交互，获得该时刻的状态和相应的回报，此时 critic 网络将根据动作、状态和回报的集合计算相应的收益函数(有时是 TD-error，用于评价 actor 网络的动作)，并传给 actor 网络和环境，actor 网络基于此调整动作的决策概率分布，并进行下一步动作选择，最终获得最优策略.

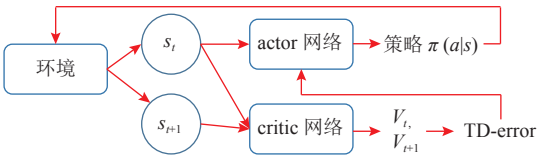


Fig. 3 Block diagram of A-C network

图 3 A-C 网络的框架图

在本文的算法中，将传统 PPO 框架做了调整，即在收益函数的定义上采用了双约束. 动作空间、状态空间以及回报约束定义为：

1)动作空间. 动作集中包含了工人与任务的匹配信息，并采用 2 维向量表来表示，其中定义第 i 个工人在第 k 次迭代时被分配的任务集合为 c_k^i. 同时，每个工人设备中将存储任务分配记录以及任务完成顺序. 每次迭代时由中央服务器决策进行分配，中央服务器依据上一轮交互所得到的全局信息(工人和任务数量、工人报酬、任务收益等)，计算平台总收益，并进行匹配. 其定义为

$$\boldsymbol{a}_k = (c_k^1, \cdots, c_k^i, \cdots, c_k^n). \tag{7}$$

2)状态空间. 状态集合中，记录了工人与任务的相关信息(工人的任务时序、各任务的收益、雇佣工人的开销、可用工人数量和剩余任务数量等)，这里由第 k 个状态可用工人集合 W_k 与可用任务集合 V_k 来表示. 每次迭代开始，每个工人根据中央服务器传递的数据，计算各个任务对于自己的开销与收益，并于中央服务器进行交互，更新此时的本地信息以及中央服务器的全局信息. 综上所述，将状态集合定义为

$$s_k = \{W_k, V_k\}. \tag{8}$$

3)回报约束. 参照式(5)的联合约束，力求保证平台收益最大化的同时，尽可能增加工人的累积收益. 将该问题视为平台与工人间非合作性竞争的纳什均衡问题. 将平台总收益的计算定义为平台整体收益减去所有工人开销，而单个工人的收益定义为工人获得报酬减去完成任务的开销. 回报约束中的平台总收益的优先级高于单个工人优先级，此处根据收益权重进行调整. 这里回报约束定义为

$$r_k = \lambda_1 R_p + \lambda_2 \sum_{w_i \in W_{\text{out}} \bigcup W_k} r_{w_i}, \tag{9}$$

其中 W_{out} 为已分配任务的工人集合，W_k 为未分配的任务集合.

在 PPO 模型的训练过程中，critic 网络根据回报 r_k 以及动作/状态集合 $\{a, s\}$ 计算其 $Value$ 值以及优势函数 A_k，进而对于下一次 actor 网络中动作选择的策略 π 进行调整，相关定义为：

$$A_k = \sum_{i=0}^{\infty} (\gamma\lambda)^i \partial_{k+i}, \tag{10}$$

$$Value_\pi(s_k) = \sum_{a \in A} \pi(a_k, s_k) \left[r_k + \gamma \sum_{s_k \in S} P_{s_k s_{k+1}} Value_\pi(s_{k+1}) \right], \tag{11}$$

$$Loss = \hat{E}_k[(A_k + Value_\pi(s_k) - Value_\pi^{\text{old}}(s_k))^2], \tag{12}$$

$$\partial_k = r_k + \gamma Value_\pi(s_{k+1}) - Value_\pi(s_k), \tag{13}$$

其中第 k 轮迭代时的价值函数为 $Value_\pi(s_k)$，γ 为折扣率，$P_{s_k s_{k+1}}$ 为状态转换概率，A_k 为此时的优势函数，λ 为优势函数权重，最终损失函数可用 $Loss$ 表示.

3.3 本地差分隐私

这里引入地理不可区分性的概念，即存在 2 个位置点 x 和 $x' \in X$，Z 是 X 通过映射机制 D 的输出结果，若 D 满足地理不可区分性，则对所有欧几里得距离 $d(x, x') \leqslant r$，其中 r 为该映射机制保护的范围，报告位置点 $z \in Z$，则有

$$D(x)(z) \leqslant \mathrm{e}^{\varepsilon d(x,x')} D(x')(z),\ x, x' \in X, z \in Z, \tag{14}$$

$$D(x)(z) = \frac{\varepsilon^2}{2\pi} \mathrm{e}^{-\varepsilon d(x,x')}. \tag{15}$$

式(14)(15)输入为 x 和 x' 时，根据该映射机制 D 的查询函数 $D(x)(z)$，将得到相同输出 z 的概率. 位

置信息中应用差分隐私是为了使真实位置点信息与其近似位置点信息拥有地理不可区分性，从而达到隐私保护的目的.

本文采用本地差分隐私的算法. 首先，工人的移动设备定位当前位置信息$(x_{\text{real}}, y_{\text{real}})$. 其次，根据当前位置坐标划定模糊位置范围，该范围是一个以 R 为模糊半径的圆形区域. 在该范围内指定 $\varepsilon \in R^2$，根据机制 D 确定候选的位置坐标集合，并根据拉普拉斯机制随机噪声，敏感度设为 Δf，该噪声服从$(0, \Delta f/\varepsilon)$的拉普拉斯分布. 最后在候选坐标集合中随机选取模糊化后的位置坐标(x, y)，并作为位置信息上传至平台.

3.4 基于 PPO 的任务分配算法

基于 A-C 网络的思想，利用 PPO 模型训练并学习任务分配策略，该方法与本文的问题非常匹配，并已成功地应用于许多其他领域. 在 DRL 的众多策略优化方法中，PPO 在易于实现样本复杂性和易于调优之间取得了平衡，以最小化目标函数进行计算和更新，同时确保与以前策略的偏差相对较小. 因此，在本文算法中的策略优化过程采用了 PPO 算法.

在本文的模型里包括了一个历史策略缓冲区 *Cache*、策略 π、actor 网络与 critic 网络，在算法 1 中展示了该模型算法的伪代码. 首先，初始化 PPO 框架，随机赋予 actor 网络与 critic 网络的相关参数相应的初始值 θ_a 和 θ_v，将 θ_a 作为初始的策略参数(行①). 随后迭代开始，最大迭代次数为 K(行②). 在环境中获取当前的可用工人集合 W_k 以及可用任务集合 V_k 的信息，其中工人的位置信息根据本地差分隐私已做模糊化处理，最终得到第 k 次迭代的状态(行③~⑧). 然后，基于状态 s_k 根据当前策略在 actor 网络中进行动作选择(行⑨)，将此时的动作集合 a_k 输入到环境中计算相应的回报 r_k 以及下一轮的状态集合 s_{k+1}(行⑩). critic 网络中计算 A_k 以及 $Value_k$，并将集合 $\{s_{k+1}, s_k, a_k, r_k, A_k, Value_k\}$存储到历史策略缓冲区 *Cache* 中(行⑪~⑫). 当 *Cache* 装满时计算偏导数，并基于根据梯度上升策略更新策略参数 θ_a(行⑬~⑮). 在从 *Cache* 中学习信息后，actor 网络的新参数 θ_a 分配给策略进行下一次采样. 同时，历史策略缓冲区被清空(行⑯).

算法 1. 基于 PPO 的动态策略优化算法.

① 分别利用权重 θ_a 和 θ_v 随机初始化 actor 网络与 critic 网络，$\theta_a^{\text{old}} \leftarrow \theta_a$；

② for $k = 1, 2, \cdots, K$ do

③ for $n = 1, 2, \cdots, N$ do

④ $W_k^n = w_n$；

⑤ $V_k^n = v_n$；

⑥ end for

⑦ 获取当前可用工人集合和任务集合 $W_k = \{W_k^1, W_k^2, \cdots, W_k^n\}$，$V_k = \{V_k^1, V_k^2, \cdots, V_k^n\}$；

⑧ 获取当前状态集合 $s_k = \{W_k, V_k\}$；

⑨ 根据 $\pi(a_k|s_k, \theta_a^{\text{old}})$获取动作集合 a_k；

⑩ 获取 r_k 以及下一轮的状态集合 s_{k+1}；

⑪ critic 网络中计算 A_k 以及 $Value_k$；

⑫ 将 $\{s_{k+1}, s_k, a_k, r_k, A_k, Value_k\}$存储到 *Cache* 中；

⑬ if $t\%|Cache| == 0$ then

⑭ $$\Delta\theta_a = \frac{1}{|Cache|}\sum_{j=1}^{|Cache|}\{[r_j + \gamma Value(s_{j+1};\theta_v) - Value(s_j;\theta_v)]^2, A_k\};$$

⑮ 根据梯度上升策略，利用 $\Delta\theta_a$ 更新 θ_a；

⑯ $\theta_a^{\text{old}} \leftarrow \theta_a$，清空 *Cache*；

⑰ end if

⑱ end for

4 实 验

4.1 实验参数设置

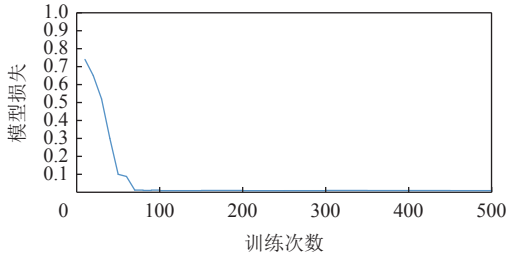
本文选用 Gowalla 和 TaskMe 这 2 个数据集进行模拟实验，从中提取部分数据的位置以及时间信息，并将添加在一定范围内随机生成的数据作为任务奖励等其他信息，最终生成拥有 2 000 个任务和 1 000 个工人的模拟数据集合. 其中，每个任务的奖励设置在 8~20 的范围内并按照 N~$(12, 4)$的正态分布进行随机生成，任务的时间则设置在 10~60 的范围内随机生成. 最后，根据实验的不同要求，选用该数据集中部分任务以及工人的数据信息在一个 200×200 的正方形传感区域空间内进行模拟实验.

首先，设置了不同的工人与任务数量下损失函数的对比实验，目的是验证工人与任务数量对损失函数的收敛性的影响. 在一个 200×200 的正方形传感区域空间内，分别测试了 80 个任务和 5 个工人、300 个任务和 15 个工人、800 个任务和 30 个工人这 3 种不同情境下的损失函数. 其次，与现有的传统方法(贪婪算法、蚂蚁算法)以及其他 DRL 方法(DDQN)针对收敛速度、最大收益以及任务覆盖率的对比实验. 该部分将蚂蚁算法中蚂蚁数、迭代次数和随机选择的概率分别设置为 10, 30 000, 0.1，对于基于 DDQN 的算法将其重播内存容量设为 10 000 次，迭代次数

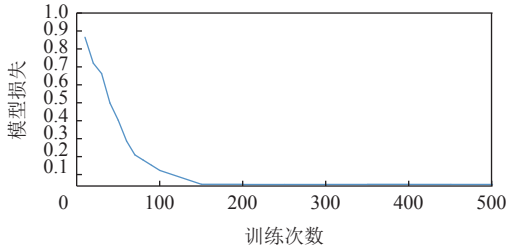
设为 30 000 次, 随机选择的概率从 0.9 开始, 然后逐渐衰减到 0.1. 最后, 通过消融实验来验证隐私保护的有效性. 在该实验中将本文算法与 DDQN 的算法以及去除掉差分隐私时的算法进行比较, 实验设置参数与对比实验相同.

4.2 模型损失

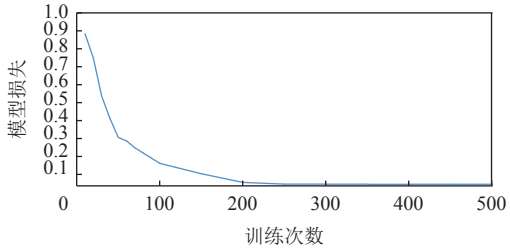
本节进行了不同工人数量以及任务数量的模拟实验, 图 4(a)中迭代次数在 100 次以内, 大约在第 70 次时达到收敛; 图 4(b)中模型在迭代次数约 120 次时达到收敛; 图 4(c)中在迭代 280 次时达到收敛. 可以看出, 该算法收敛效果主要受到工人与任务的数量影响, 随着其数量的增多, 收敛速度将变慢.



(a) 80 个任务和 5 个工人



(b) 300 个任务和 15 个工人



(c) 800 个任务和 30 个工人

Fig. 4 System loss for different numbers of tasks and workers

图 4 任务与工人不同数量下的系统损失

4.3 对比实验

本节实验不仅针对传统 MCS 任务分配方法(即蚂蚁算法和贪婪算法的对比), 而且加入了同为基于 DRL 的任务分配算法(即基于 DDQN 的算法), 在任务覆盖率、性能、收敛性以及最大收益上做了相应对比实验. 在图 5 中, 可以看到基于 DDQN 以及基于 PPO 的 2 种 DRL 算法在系统平均开销的收敛情况, 结果表示基于 DDQN 的算法虽然可以比本文算法能更快收敛, 但本文算法可以达到更小的平均系统开销. 图 6 展示了 4 种算法的平台收益情况, 贪婪算法和蚁群算法由于是静态的算法, 因而不需要多轮迭代, 但其平台收益与基于 DRL 的算法相比差距甚远; 而基于 DDQN 的算法同样有更快的收敛性, 但本文的基于 PPO 的算法可以最终达到最大收益.

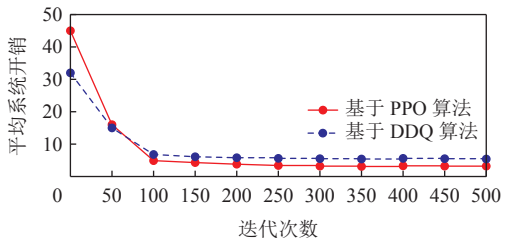


Fig. 5 Comparison of average system costs

图 5 平均系统开销的比较

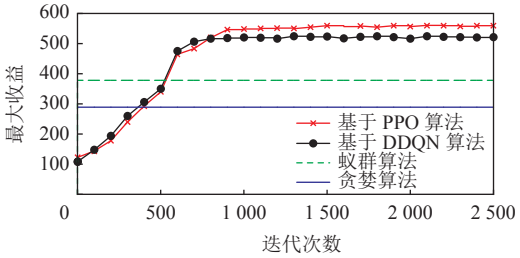


Fig. 6 Comparison of maximum profits

图 6 最大收益的比较

首先引入任务覆盖率的概念: 当一个任务在其可接受的时间范围内被分配出去且完成, 则可称为该任务被覆盖. 因此任务覆盖率可被定义为被分配掉的任务数与总任务数的比值. 如表 1 所示, 在平台最大利润和任务覆盖率上, 基于 DDQN 和 PPO 的 2 种 DRL 算法均远高于贪婪算法和蚁群算法等传统方法, 且基于 PPO 算法比基于 DDQN 的算法表现更为优异. 图 7~9 展示了 4 种算法在平均开销、总开销以及工人平均收益上的对比, 结果表明本文的基于 PPO 算法均优于其他算法, 而贪婪算法表现最差.

Table 1 Comparison of Maximum Profit and Coverage Ratio

表 1 最大利润与覆盖率的对比

算法	最大利润	覆盖率
贪婪算法	298	0.41
蚁群算法	352	0.52
基于 DDQN 算法	502	0.72
基于 PPO 算法(本文)	512	0.76

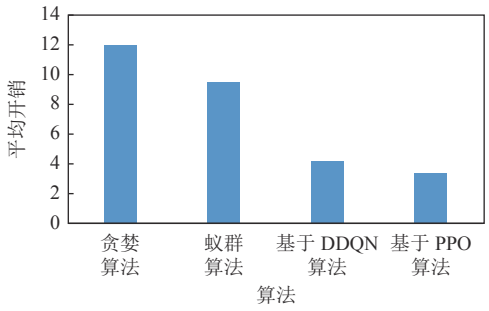


Fig. 7 Comparison of average costs

图 7 平均开销的对比

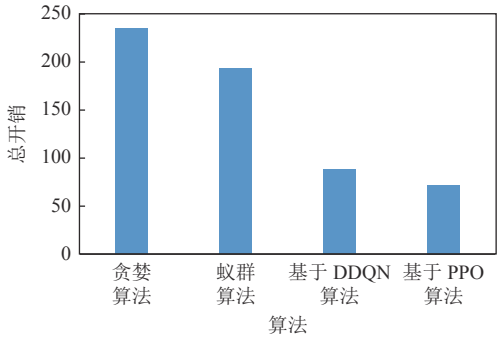


Fig. 8 Comparison of total costs

图 8 总开销的对比

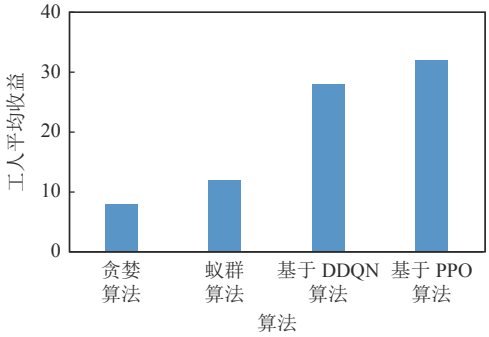


Fig. 9 Comparison of average revenue of workers

图 9 工人平均收益的对比

4.4 消融实验

如图 10 所示，本文针对差分隐私的有效性做了消融实验，实验中对比了有差分隐私的性能以及没有差分隐私的性能，并与基于 DDQN 的算法模型进行对比. 实验结果表明，去除差分隐私性能会有更好的效果，是因为模糊化的位置信息影响了模型的计算性能，但加入差分隐私的方法可以在不损失过多性能的前提下保护工人信息的隐私. 此外，为了验证该算法的性能随差分隐私保护程度的变化，消融实验中加入了不同的隐私保护机制覆盖范围下的算法性能的对比实验，如图 11 所示，其中 r 表示本文 3.3 节所提到的差分隐私机制的保护范围，当保护范围越大时，则需保证该范围的地理不可区分性，故保护程度越高. 由此可见，随着保护范围的增加算法性能所受影响较大，需选择合适的保护强度，实现在保护隐私的前提下保证算法性能.

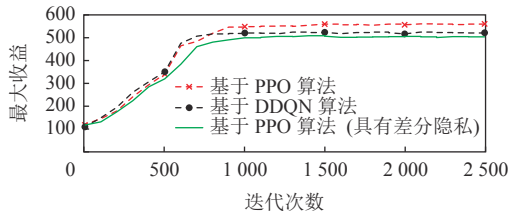


Fig. 10 Ablation experiment

图 10 消融实验

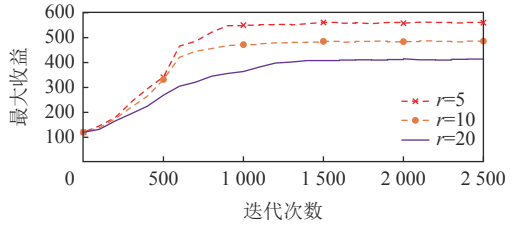


Fig. 11 Comparison under different protection levels

图 11 不同保护程度下的对比

5 结论与展望

在本文中，针对 MCS 的感知任务分配问题，在工人与任务的位置、状态信息不断改变的动态系统中，考虑了任务分配机制的合理性与工人信息的隐私保护等问题，将其定义为一个基于离散型数据集来进行动态规划的优化问题，并利用差分隐私和深度强化学习的相关算法及模型去解决该问题. 将 PPO 模型作为决策模型训练和学习，在每次迭代中，考虑当前状态下的每个工人开销的差异性以及完成任务的时序性等因素，利用联合约束，在保证平台收益最大化的同时，尽可能增加工人的累积收益，在这样的动态系统中不断优化分配策略. 此外，还在工人的移动设备与中央服务器的交互中加入了差分隐私的方法来保证工人的隐私. 实验结果也证明了本文方法的有效性.

在未来的工作中，将探索更多隐私保护的策略，并且在保证隐私的同时进一步提升模型的性能. 此外，也考虑在该模型中加入数据预测的机制，在收集处理感知数据的同时，基于历史经验数据进行某一范围内的数据预测，提升模型整体效率.

作者贡献声明：杨明川负责实验及相关研究工作，并完成论文撰写；朱敬华提出算法思路，设计论文整体框架；李元婧负责数据分析并协助撰写论文；奚赫然提出修改意见并修改论文.

参 考 文 献

[1] Ganti R K, Fan Ye, Hui Lei. Mobile crowdsensing: Current state and future challenges[J]. IEEE Communications Magazine, 2011, 49(11): 32−39

[2] Guo Bin, Yu Zhiwen, Zhou Xingshe, et al. From participatory sensing to mobile crowd sensing[C]//Proc of the 12th IEEE Int Conf on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS). Piscataway, NJ: IEEE, 2014: 593−598

[3] Lu Anqi, Zhu Jinghua. Worker recruitment with cost and time constraints in mobile crowd sensing[J]. Future Generation Computer Systems, 2020, 112(15): 819−831

[4] Li Doudou, Zhu Jinghua, Cui Yanchang. Prediction-based task allocation in mobile crowdsensing[C]//Proc of the 15th Int Conf on Mobile Ad-Hoc and Sensor Networks (MSN). Piscataway, NJ: IEEE, 2019: 89−94

[5] Wang Zhen, Zhu Jinghua, Li Doudou. Prediction based reverse auction incentive mechanism for mobile crowdsensing system[C]//Proc of the 13th Int Conf on Combinatorial Optimization and Applications. Berlin: Springer, 2019: 541−552

[6] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529−533

[7] Kungurtsev V , Egan M , Chatterjee B , et al. Asynchronous optimization methods for efficient training of deep neural networks with guarantees[C]//Proc of the 35th AAAI Conf on Artificial Intelligence. Palo Alto, CA : AAAI, 2021: 8209−8216

[8] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]//Proc of the 13th Int Conf on Machine Learning. New York: PMLR, 2014: 387−395

[9] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization[C]//Proc of the 14th Int Conf on Machine Learning. New York: PMLR, 2015: 1889−1897

[10] Liu Quan, Zhai Jianwei, Zhang Zongzhang, et al. A survey on deep reinforcement learning[J]. Chinese Journal of Computers, 2018, 41(1): 1−27 (in Chinese)
(刘全, 翟建伟, 章宗长, 等. 深度强化学习综述简[J]. 计算机学报, 2018, 41(1): 1−27)

[11] Dwork C. Differential privacy [C]//Proc of the 33rd Int Colloquium on Automata, Languages, and Programming. Berlin: Springer, 2006: 1−12

[12] Cheung Manhon, Hou Fen, Huang Jianwei, et al. Distributed time-sensitive task selection in mobile crowdsensing[J]. IEEE Transactions on Mobile Computing, 2020, 20(6): 2172−2185

[13] Li W, Jia Bin, Xu Haotian, et al. A multi-task scheduling mechanism based on ACO for maximizing workers' benefits in mobile crowdsensing service markets with the Internet of things[J]. IEEE Access, 2019, 7: 41463−41469

[14] Guo Bin, Liu Yan, Wang Leye, et al. Task allocation in spatial crowdsourcing: Current state and future directions[J]. IEEE Internet of Things Journal, 2018, 5(3): 1749−1764

[15] Zhou Zhenyu, Liao Haijun, Gu Bo, et al. Robust mobile crowd sensing: When deep learning meets edge computing[J]. IEEE Network, 2018, 32(4): 54−60

[16] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. Nature, 2015, 518(6254): 539−543

[17] Kim I, Morrison J R. Learning based framework for joint task allocation and system design in stochastic multi-UAV systems[C]//Proc of the 15th Int Conf on Unmanned Aircraft Systems (ICUAS). Piscataway, NJ: IEEE, 2018: 324−334

[18] Tao Xi, Song Wei. Task allocation for mobile crowdsensing with deep reinforcement learning[C/OL]//Proc of the 18th IEEE Wireless Communications and Networking Conf (WCNC). Piscataway, NJ: IEEE, 2020[2022-08-21].https://ieeexplore.ieee.org/abstract/document/9120489

[19] Patel T, Wagenhäuser A, Eibel C, et al. What does power consumption behavior of hpc jobs reveal?: Demystifying, quantifying, and predicting power consumption characteristics[C]//Proc of the 34th IEEE Int Parallel and Distributed Processing Symp (IPDPS). Piscataway, NJ: IEEE, 2020: 799−809

[20] Raskhodnikova S, Smith A, Lee H K, et al. What can we learn privately[C]//Proc of the 54th Annual Symp on Foundations of Computer Science. Piscataway, NJ: IEEE, 2008: 531−540

[21] Chen Rui, Fung Benjiamin C M, Desai B C, et al. Differentially private transit data publication: A case study on the montreal transportation system[C]//Proc of the 18th ACM SIGKDD Int Conf on Knowledge Discovery and Datamining. New York: ACM, 2012: 213−221

[22] Wang Jie, Wang Feng, Li Hongtao. Differential privacy location protection scheme based on Hilbert curve[J]. Security and Communication Networks, 2021, 9(7): 366−417

[23] Golden B L, Levy L, Vohra R. The orienteering problem[J]. Naval Research Logistics, 1987, 34(3): 307−318

[24] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//Proc of the 33rd Int Conf on Machine Learning. New York: PMLR, 2016: 1928−1937

Yang Mingchuan, born in 1997. Master. His main research interests include mobile crowd-sensing and deep reinforcement learning.
杨明川, 1997年生. 硕士. 主要研究方向为移动群智感知和深度强化学习.

Zhu Jinghua, born in 1976. PhD, professor. Her main research interests include uncertain data management, data mining, and sensor networks.
朱敬华, 1976年生. 博士, 教授. 主要研究方向为不确定数据管理、数据挖掘、传感器网络.

Li Yuanjing, born in 1997. Master. Her main research interests include mobile crowd-sensing, combinatorial optimization, and approximation algorithms.

李元婧, 1997 年生. 硕士. 主要研究方向为移动群智感知、组合优化和近似算法.

Xi Heran, born in 1980. Master, lecturer. His main research interests include uncertain data management, data mining, and sensor networks.

奚赫然, 1980 年生. 硕士, 讲师. 主要研究方向为不确定数据管理、数据挖掘、传感器网络.