



具有角色意识的社区服务型时空众包任务分配

徐静如, 董红斌, 赵炳旭, 冀若含

引用本文:

徐静如,董红斌,赵炳旭,冀若含. 具有角色意识的社区服务型时空众包任务分配[J]. 智能系统学报, 2023, 18(2): 293–304.

XU Jingru,DONG Hongbin,ZHAO Bingxu,JI Ruohan. Community service-oriented spatiotemporal crowdsourcing task allocation with role awareness[J]. *CAAI Transactions on Intelligent Systems*, 2023, 18(2): 293–304.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202205017>

您可能感兴趣的其他文章

多中心维修任务分配研究现状

Research status of multicenter maintenance task assignment

智能系统学报. 2022, 17(3): 448–458 <https://dx.doi.org/10.11992/tis.202109008>

群智能新研究:角色匹配的狼群劳动分工

A new approach to swarm intelligence: role-matching labor division of a wolf pack

智能系统学报. 2021, 16(1): 125–133 <https://dx.doi.org/10.11992/tis.202007043>

基于Faster R-CNN的多任务增强裂缝图像检测方法

Multi-task enhanced dam crack image detection based on Faster R-CNN

智能系统学报. 2021, 16(2): 286–293 <https://dx.doi.org/10.11992/tis.201910004>

基于禁忌搜索的时空众包任务分配算法

Spatiotemporal crowdsourcing task assignment algorithm based on tabu search

智能系统学报. 2020, 15(6): 1040–1048 <https://dx.doi.org/10.11992/tis.202006055>

面向多机器人动态任务分配的事件驱动免疫网络算法

Event-driven immune network algorithm for multi-robot dynamic task allocation

智能系统学报. 2018, 13(6): 952–958 <https://dx.doi.org/10.11992/tis.201707022>

基于多Agent的面向订单的离散制造系统建模与仿真研究

Modeling and simulation of order-oriented discrete manufacturing system based on multi-Agent

智能系统学报. 2016, 11(2): 233–240 <https://dx.doi.org/10.11992/tis.201506008>

DOI: 10.11992/tis.202205017

网络出版地址: <https://kns.cnki.net/kcms/detail/23.1538.TP.20221008.1132.006.html>

具有角色意识的社区服务型时空众包任务分配

徐静如, 董红斌, 赵炳旭, 冀若含

(哈尔滨工程大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001)

摘要: 为了适应社区众包配送中投递箱或代收点容纳量有限等情况, 本文提出社区服务型时空众包任务分配问题, 根据用户定义的时间将物品较为实时地配送到地。针对该问题, 本文通过基于角色的协作模型 E-CARGO(environments-classes, agents, roles, groups, objects)形式化问题, 针对高资格值工人配送高价值订单集的目标, 提出基于贪婪分配的 PQGR(places-qualification-based greedy)算法、基于考虑代理和角色冲突的团队多角色分配方法的 PQGM(places-qualification-based GMAC)算法以及进一步缩短运行时间的改进 PQGM 算法。数据处理和量化方面, 提出基于核密度聚类的新型角色感知方法以实现任务的有效划分, 提出基于学习遗忘曲线的代理地点资格值多阶段量化模型, 实现代理地点资格值的在线学习和自适应更新。最后, 本文在 gMission 数据集和合成数据集上进行实验, 验证了算法的有效性和效率。

关键词: 社区服务型时空众包; 任务分配; E-CARGO; 基于角色协作; 核密度聚类; 角色感知; 学习遗忘曲线; 地点资格值多阶段量化

中图分类号: TP311 **文献标志码:** A **文章编号:** 1673-4785(2023)02-0293-12

中文引用格式: 徐静如, 董红斌, 赵炳旭, 等. 具有角色意识的社区服务型时空众包任务分配[J]. 智能系统学报, 2023, 18(2): 293-304.

英文引用格式: XU Jingru, DONG Hongbin, ZHAO Bingxu, et al. Community service-oriented spatiotemporal crowdsourcing task allocation with role awareness[J]. CAAI transactions on intelligent systems, 2023, 18(2): 293-304.

Community service-oriented spatiotemporal crowdsourcing task allocation with role awareness

XU Jingru, DONG Hongbin, ZHAO Bingxu, JI Ruohan

(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract: This paper proposes the community service-oriented spatiotemporal crowdsourcing task allocation problem to adapt to the limited amount of delivery boxes or collection points in community crowdsourcing. The articles will be relatively delivered to the ground in real time according to the user-defined time. This paper proposes a PQGR algorithm based on greedy allocation, a PQGM algorithm based on the GMRACRA method, and a PQGMA algorithm that further shortens the running time to address the aforementioned problem. This algorithm aims to realize the objective of delivering high-value order sets by workers with high qualification value through the formalization problem of E-CARGO based on the role cooperation model. Considering data processing and quantification, a new role perception method based on kernel density clustering is proposed to realize effective task division. The places-qualification-based multistage quantitative model of an agent is proposed on the basis of learning and forgetting curves to realize online learning and adaptive updating of agent location qualification value. Finally, experiments on gMission and synthetic datasets are conducted to verify the effectiveness and efficiency of the algorithm.

Keywords: community service-oriented spatiotemporal crowdsourcing; task allocation; E-CARGO; role-based collaboration; kernel density clustering; role perception; learning and forgetting curve; multistage quantification of site qualification value

收稿日期: 2022-05-16. 网络出版日期: 2022-10-09.
基金项目: 黑龙江自然科学基金项目(LH2020F023).
通信作者: 董红斌.E-mail: donghongbin@hrbeu.edu.cn.

近年随着我国大力推进扶贫惠农和社区文明建设, 通过网络对产品进行售卖并以社区为基本单位配送成为一种迅速普及的消费方式, 较为常

见的有美团优选、多多买菜等。这类众包的配送模式有配送到户模式、直营便利店模式、社区店加盟代收点模式和冷链改造的智能快递投递箱模式等。

其中,对于应用普遍的社区店加盟代收点和智能快递投递箱等模式配送时存在一些问题:1)各平台在店租压力大、消费水平高的一线城市,会受到高额的设备维护费用和店铺租赁费用的限制,代收点容量和投递箱数量有限,可能存在物品数量超出投递箱收纳量或消费高峰期店家储存不当的情况;2)虽在代收地点设置冷链技术改造的冷箱或保温箱,代收点可以一定程度上保证新鲜度和安全性,但较发货点仍差异较大;3)所送达的物品包括及时性相对较强的物品。考虑到以上问题,本文提出社区服务型时空众包(community service-oriented spatiotemporal crowdsourcing, CSSC)任务分配问题,实时地安排工人将物品在收货时间的软时间窗内放入代收点。另外,现疫情阶段,封控区的变化以及各种不确定因素对工人的熟悉度以及应变能力要求提高,本文提出以熟悉度为基础的代理地点资格值并纳入分配考虑,任务多时高地点资格值代理可确保本次以及后面任务的顺利完成。

因此,本文所面向的问题是带冲突的同起点不同时不同固定终点的任务分配问题,目标是对于高价值量的订单集派遣高地点资格值的工人配送。针对上述目标,本文基于贪婪分配策略和 GMRA-CRA 方法提出 CSSC 问题任务分配方式,数据处理和量化方面,基于核密度聚类进行角色划分,基于学习遗忘曲线多阶段量化代理的地点资格值。

1 相关工作

现时空众包任务分配问题的研究中,融入群体智能思想的多工人协作场景具有较好的适应性和效果。朱海滨博士等^[1]提出的基于角色的协作(RBC)是处理协作问题的一种有效方法。RBC为适应性协作(adaptive collaboration, AC)^[2-4]的一种,使用角色作为底层机制,强调了角色的重要性,并提出了 E-CARGO(environments-classes, agents, roles, groups, objects)模型^[5],包括环境(Environment)、类(Class)、代理(Agent)、角色(Role)、组(Group)和对象(Object)。该模型囊括了一些基于角色的基本概念,可以基于此完成模型的搭建。

RBC实现任务分配主要基于以下模型及其拓展模型:基于KM算法,应用团队角色分配(group role assignment, GRA)^[6-7]将一个角色分配给多个

代理来寻找最优的角色分配计划,使整个团队的绩效最大化;在 GRA 基础上加入一个代理可能被分配给多个角色的情况,通过线性规划工具包实现团队多角色分配(group multirole assignment, GMRA)^[8];引入同角色代理可能冲突的情况^[9],以及进一步引入同代理角色冲突的情况,实现考虑冲突的团队多角色分配 GMRACRA^[10]等。本文基于 GMRACRA 模型实现带有角色和代理冲突的任务分配。

以上方法可适用于多种工业工程^[11-14]以及社会仿真^[15-18]问题。与本文相关的社会仿真类研究有 Zhu^[19]提出通过带限制的 GRA 和模拟退火实现的代理分类, Huang 等^[20]针对最后一英里分配问题通过 GRA 方法进行建模分配, Jiang 等^[21]针对自服务时空众包问题提出 QGRA(quasi group role assignment)并融入工人偏好对问题建模,通过 GMRA 方法进行分配等。受以上仿真的启发,本文针对 CSSC 模式任务分配问题通过角色建模,但由于终点是固定的, Huang 等^[20]的方法和 Jiang 等^[21]中的多次分配不适用于本文的分配问题。且 Huang 等^[20]中所采用的自适应角色方法是通过设置类的数量间接控制角色内的数量,若最后循环取最优值时类较少则类内数量无法保证,在实际生活中没有考虑代理所用配送设备问题。本文针对类的划分考虑了类内的任务数量,且考虑了多人完成一个角色的情况,适应性更强。

另外,现有涉及代理对某地点资格值的研究中,存在直接随机生成熟悉度前三名地点的熟悉度值^[20],根据艾宾浩斯遗忘曲线定义^[22](其中设定遗忘曲线的遗忘程度为 $\frac{100h}{z(\lg t)+h}$)等方式。前者这类方法随机性太强,没有学习、变化的过程,后者根据艾宾浩斯遗忘曲线量化的每次学习程度都达到百分之百且没有累积的过程。本文模型中代理地点资格值量化同学习遗忘有着相似的变化过程,但对各个地点同时存在遗忘和累积的过程,适用性和可持续性强。

2 CSSC 问题定义

通过 E-CARGO 模型将 CSSC 问题形式化,系统 \sum 可以描述为9元组 $\sum ::= \langle C, O, A, M, R, E, G, s_0, H \rangle$,其中 C 是类集, O 是对象集, A 是代理集, M 是消息集, R 是角色集, E 是环境集, G 是组集, s_0 是系统的初始状态, H 是用户集。本文使用 m 来表示代理集合 A 的大小,即 $m = |A|$, n 表示角色集的大小,即 $n = |R|$, i_1, i_2, \dots 表示代理的索引, j_1, j_2, \dots

表示角色的索引。

定义1 任务被定义为 $t ::= \langle \text{id}, \text{tName}, \text{time}, \text{location}, \text{value}, \text{place_id} \rangle$, 其中 id 、 tName 分别是任务 t 的序号和名称, 后面4个属性分别代表了任务的预计交付时间、位置、价值和该任务所选择的代收点 id 。任务集用 Tasks 表示。

定义2 角色被定义为 $r ::= \langle \text{id}, \text{place_id}, \text{value}, \text{task_set} \rangle$, 其中 id 是角色 r 的序号, 即角色 j ($0 \leq j < n$), 后面3个属性分别代表了角色 j 的代收点 id 、该角色的总价值以及所包含的任务集。角色集用 R 表示。

本文角色定义为不同的任务集, 这样可以将任务分配中一对多问题转换为一对一问题, 各任务集中的任务相互独立且互不包含。本文设计角色感知模块定义任务到角色(任务集)的转换过程, 具体内容在第3.1节中呈现。

定义3 代理被定义为 $a ::= \langle \text{id}, \text{AP_list}[i] \rangle$ 。其中 id 是代理的序号, 即代理 i ($0 \leq i < m$); $\text{AP_list}[i]$ 表示代理 i 访问各代收点时间序列的列表, $\text{AP_list}[i][p]$ 存储代理 i 访问代收点 p 的时间序列。该序列用于多阶段量化代理地点资格值来作为任务分配的参考因素, 具体定义及量化策略在第3.2节中呈现。代理集用 A 表示。

定义4 组 g 中的一个元组 $\langle a, r \rangle$ 是指一个角色和代理的分配。

定义5 角色最小范围向量 L 是一个 n 维向量, 表示一个角色 j 完成所需的最少代理数量。 $L[j] = 1$ ($0 \leq j < n$) 表明角色 j 所需代理的数量为1, 该角色(众包任务集)至少需要一个人完成。

定义6 代理能力向量 L^a 是一个 m 维向量, 表示每个代理所能接受的最大角色数量, 本文中 $L^a[i] = 1$ ($0 \leq i < m$) 表示一天内最多可以分配给代理 i 的角色数。例如, 给定5个代理, $L^a[i] = [8, 12, 9, 9, 11]$ 。

定义7 T 是大小为 $m \times n$ 的分配矩阵, $T[i, j] \in \{0, 1\}$, $T[i, j] = 1$ 表示将角色 j 分配给代理 i , $T[i, j] = 0$ 则不分配。

定义8 代理冲突矩阵 A^c 是一个 $m \times m$ 的对称矩阵, $A^c[i_1, i_2] \in \{0, 1\}$ ($0 \leq i_1, i_2 < m$), 其中 $A^c[i_1, i_2] = 1$ 表示代理 i_1 和 i_2 冲突, 当值为0时表示不冲突。 $A^c[i, i] = 0$ ($0 \leq i < m$) 定义为一个代理与自身没有冲突。

本文代理冲突考虑的是在一个多对一的分配问题中避免有合作的两个代理有较强的敌对情绪从而影响合作完成任务的情况。实验部分以5%的概率随机设置冲突情况。

定义9 角色冲突矩阵 R^c 是一个 $n \times n$ 的对称

矩阵, $R^c[j_1, j_2] \in \{0, 1\}$ ($0 \leq j_1, j_2 < n$), 其中 $R^c[j_1, j_2] = 1$ 表示角色 j_1 与角色 j_2 冲突, 当值为0时表示不冲突。 $R^c[j, j] = 0$ ($0 \leq j < n$) 表示一个角色与自身没有冲突。

本文角色冲突考虑的是不同角色送货时间的冲突, 考虑到一定半径内的移动距离与备货时间, 当一个角色的送货时间与另一个角色的送货时间相差30 min以上则定义为不冲突, 小于30 min则为角色冲突。

定义10 资格矩阵 Q 是一个 $m \times n$ 的矩阵, 其中 $Q[i, j] \in [0, 1]$ 表示代理 i ($0 \leq i < m$) 对于角色 j ($0 \leq j < n$) 的资格值。

本文中, 定义资格值受代理对该角色所在代收点资格值和该角色的总价值影响。本着在线学习、可持续的原则, 第3.2节中定义代理地点资格值多阶段量化方法求得地点资格值 F_{ik} , 下标 ik 是代理 i 对第 k 个地点的表示。资格值矩阵计算公式如式(1)所示, 其中 V_j 为角色 j 的总价值。

$$Q[i, j] = F_{ik} \cdot V_j \quad (0 \leq i < m, 0 \leq j < n) \quad (1)$$

定义11 组效益 σ' 定义为已分配代理的资格之和, 如式(2)所示, σ' 的意义是众包工作者在整个调度过程中使得高地点资格值的代理完成高价值任务的最大组效益。

$$\sigma' = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i, j] \times T[i, j] \quad (2)$$

可行性定义 如果一个角色 j ($0 \leq j < n$) 被分配足够的代理, 即 $\sum_{i=0}^{m-1} T[i, j] \geq L[j]$, 则该角色 j 是可行的。如果每个角色 j 都可行, 即 $\sum_{i=0}^{m-1} T[i, j] \geq L[j]$ ($0 \leq j < n$) 都成立, 则该分配矩阵 T 是可行的, 如果 T 可行那么该组 g 也是可行的。

3 数据处理方法与算法

3.1 角色感知算法

为实现从一对多分配到一对一分配的转换, 简化分配过程, 采用角色感知算法(role perception algorithm, RPA)将任务划分为不同的角色(任务集)。本文根据用户自己选择的代收点将任务划分, 若没有选择则就近分配。然后在不同的代收点处根据时间对任务进一步划分, 划分为不同的角色。为自动确定集群数量, 本文应用核密度聚类定义具有类内数量限制的角色感知方法。

核密度聚类中核密度估计(kernel density estimation, KDE)可以无需假设密度分布或特征参数来估计数据集的可能分布。本文应用的核函数

为高斯函数(即正态分布),核密度估计的公式如式(3),其中 h 为带宽。在概率密度估计曲线的基础上以极值为分界点划分区间,从而得到各个时间段,将各时间段内的任务划分为一个角色,完成角色的分配。

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (3)$$

如算法 1 所示,输入为当天的众包任务集 $Tasks$ 、核密度聚类带宽 h 和地点集 P ,输出为角色集 R 。 T_p 是以地点 p 为代收点的任务集, \minNum 、 \maxNum 分别指当任务集过大需要分割时一个角色类中最大任务数和最小任务数(本文中设置最小、最大任务数为 3 和 8)。KernelDensityClustering(T_p, h) 是对任务集 T_p 进行核密度聚类的函数。第 1~3 行将任务加入代收点任务集,第 4~23 行分别将各代收点的任务划分为角色集。第 5 行利用核密度聚类算法对该代收点的任务进行划分,第 6 行对划分结果进行进一步处理。第 7~9 行将任务数目小于最大限制的任务集打包为一个角色。否则,根据第 10~22 行将其进一步以 (\minNum, \maxNum) 内的随机数 $Temp$ 为大小分块,直至最后一块数量小于 \maxNum 。

算法 1 角色感知算法

输入 $Tasks, h, P$

输出 R

1) For t in $Tasks$:

2) 将 t 加入所选择代收点的任务集 T_p

3) End for

4) For p in P :

5) Result=KernelDensityClustering(T_p, h)

6) For r in Result:

7) If r 中包含的任务数小于 \maxNum :

8) 将 r 中的任务集打包为角色放入 R

9) End if

10) Else:

11) Flag=0

12) While(Flag< r 中包含的任务数):

13) Temp←取 [\minNum, \maxNum] 内的随机数

14) If Flag+Temp≤(r 中包含的任务数):

15) 将 $r[Flag, Flag+Temp)$ 打包为角色放入 R

16) Flag←Flag+Temp

17) End if

18) Else:

19) 将 $r[Flag, len(r))$ 打包为角色放入 R

20) Flag←Flag+Temp

21) End while

22) End for

23) End for

24) Return R

3.2 代理地点资格值的多阶段量化模型

为了根据代理对代收点的出勤时间和次数定义地点资格值,本文受推荐系统中兴趣度量化的启发^[23],根据学习遗忘曲线提出代理地点资格值的多阶段量化模型(places-qualification-based multistage quantitative model, PQMQM)。模型中,若近期频繁派送某代收点订单,则资格值较大。

在 PQMQM 中,每次到达代收点就是一次提升以熟悉度为基础的地点资格值的过程,多次到达同一代收点就是重复学习的过程。图 1 为代理 3 次重复学习时的地点资格值变化过程。

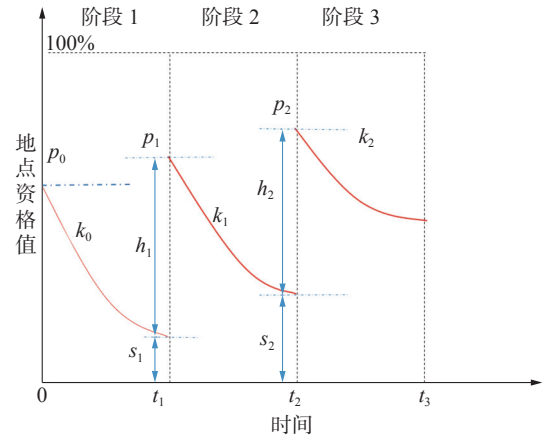


图 1 代理地点资格值变化过程

Fig. 1 Agent place qualification value change process

由图 1 可以看出每次学习后都是一个地点资格值降低的过程,但每次学习与遗忘时资格值的增加量和遗忘速率都是不同的。如图 1, p_n 是阶段初始值, k_n 为遗忘速率, h_n 是新一轮重复学习后资格值的增加量, s_n 为上一阶段经历遗忘后剩余的资格值。可以看出 $p_n = h_n + s_n$, $k_0 > k_1 > k_2$, $h_1 > h_2$, 随着代理一段时间内重复去往某个地点,每次新了解的内容(新增的资格值)逐渐减少, h 的值逐渐减小。具体的计算公式如下:

$$s_n = p_{n-1} e^{-k_{n-1}(t_n - t_{n-1})} \quad (4)$$

$$h_n = p_0 e^{-\frac{n}{d_1}} \quad (5)$$

$$p_n = p_{n-1} e^{-k_{n-1}(t_n - t_{n-1})} + p_0 e^{-\frac{n}{d_1}} \quad (6)$$

$$k_n = \frac{\ln(1 + (\theta - 1)e^{-k_{n-1}(t_n - t_{n-1})}) - \ln \theta}{-(t_n - t_{n-1})} \quad (7)$$

$$\theta_n = (t_n - t_{n-1})^{d_2} \cdot \theta_0 \quad (8)$$

$$d_2 = \begin{cases} 1 + 0.3 \cdot (t_n - t_{n-1} - 1), & t_n - t_{n-1} \leq 2 \\ 0.5 \cdot (t_n - t_{n-1}), & t_n - t_{n-1} > 2 \end{cases} \quad (9)$$

式中: d_i 取值为 4.5; t_{n-1} 和 t_n 为重复学习的任意相邻时刻; k_{n-1} 为遗忘曲线从 t_{n-1} 到 t_n 的遗忘速率。 θ 为惰性因子,根据时间差 $t_n - t_{n-1}$ 来调整 k_n ,当两个时间差相差越大,则该代理当前阶段的 θ 值越大。式(4)通过指数函数定义前一阶段经历遗忘后的剩余资格值,式(5)定义当前阶段资格值的增加量,式(6)表示当前时刻经历过学习遗忘过程后的总资格值,式(7)调整式(6)中的遗忘速率,式(8)和(9)调整式(7)中的惰性因子。

本文以天为单位,设定最大时间阈值 T_{\max} 表示参照历史信息的时间跨度。实验中 $T_{\max} = 15$ 对于每个工人根据近 15 天的历史数据中对 5 个代收点的访问情况进行计算。具体代理地点资格值的多阶段量化模型伪代码如下。

算法 2 PQMQM 算法

输入 $P, A, AP_list[][]$

输出 F

- 1) For a in $[0, 1, 2, \dots, \text{len}(A)]$:
- 2) For p in $[0, 1, 2, \dots, \text{len}(P)]$:
- 3) 初始化 k_0, p_0, θ_0
- 4) $M \leftarrow AP_list[a][p]$ 中时间点的数量
- 5) $t_0 \leftarrow AP_list[a][p]$ 中第 1 个时间点的值
- 6) For i in $[1, 2, \dots, M-1]$:
- 7) $t_i \leftarrow AP_list[a][p]$ 中第 $i+1$ 个时间点的值
- 8) 计算 d_i 的值
- 9) $\theta_i = (t_i - t_{i-1})^{d_i} \cdot \theta_0$
- 10) $p_i = p_{i-1} e^{-k_{i-1}(t_i - t_{i-1})} + p_0 e^{-\frac{i}{d_i}}$
- 11) $k_i = \frac{\ln(1 + (\theta - 1)e^{-k_{i-1}(t_i - t_{i-1})}) - \ln \theta}{-(t_i - t_{i-1})}$
- 12) End for
- 13) $F[a][p] \leftarrow$ 计算当前时刻的地点资格值
- 14) End for
- 15) End for
- 16) Return F

算法 2 输入为地点集 P 、代理工人集 A 和存放每个工人对每个地点访问时间列表的 $AP_list[][]$, 输出为当前时间每个工人对每个地点的地点资格矩阵 F , $AP_list[a][p]$ 表示第 a 个工人对第 p 个代收点的访问时间列表。伪代码中第 1~3 行针对每个工人和每个地点进行遍历并初始化, 第 4 行获取第 a 个代理对第 p 个地点的访问时间列表中时间点的数量 M , 第 5 行获取访问时间列表中的第一个时间点的值 t_0 。第 6~7 行获取时间列表中除 t_0 外的每一个时间值, 第 8~11 行迭代更新 θ 、 p_i 、 k_i , 计算每个阶段的地点资格值, 第 13 行根据最后一个阶段的资格值计算当前时刻的资格值并赋

值给资格矩阵, 最终完成循环输出代理地点的资格矩阵 F 。

该方法在实际考勤中有激励作用, 一方面对于持续工作的工人存在缺勤也可以维持较高的资格值, 另一方面对于没有持续工作的工人若在一定时间内持续工作其地点资格值会较快速增长。地点资格值以熟悉度为基础但不同于熟悉度, 多次到达后资格值不一定呈现上升趋势。例如, 当设置 $k_0 = 1.1$, $p_0 = 0.5$, $\theta_0 = 3.5$, 访问时间列表为 $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]$ 、 $[0, 5, 7, 9, 11, 12, 13, 14]$ 和 $[0, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]$ 时, 代理地点资格值变化如图 2 所示。

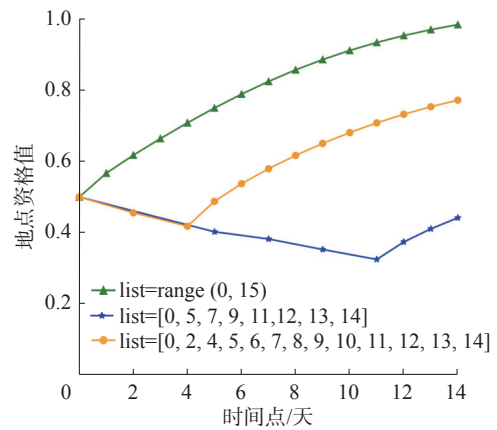


图 2 代理资格值变化情况

Fig. 2 Changes in the agent's qualification value

4 问题解决方法与算法

4.1 基于贪婪的解决方法

对于 CSSC 模式的任务分配问题, 针对地点资格值和价值这两个优化目标, 本文提出了基于地点资格值的贪婪分配方法 (places-qualification-based greedy algorithm, PQGR)。

该方法首先根据角色价值由高到低进行排序, 然后优先为价值高的角色根据所需代理数量匹配资格值高的代理, 匹配时需判断其是否存在代理和角色冲突。其具体的伪代码如算法 3 所示。

算法 3 中, 输入为任务集 $Tasks$ 、地点集 P 、代理集 A 、时间序列 $AP_list[][]$ 、带宽 h 的最小取值 $\min h$ 和最大取值 $\max h$, 输出为分配矩阵 T_{\max} 。具体实现中, 第 1 行根据不同的 h (具体取值范围在第 5.2 节中呈现) 进行遍历, 第 2~5 行计算并初始化角色集 R 、地点资格值矩阵 F 、 m 、 n 和分配矩阵 T , 第 6~17 行对角色集 R 中的每个角色 r 分配代理。第 7 行定义循环的次数为该角色所需代理的数量。第 8~10 行遍历代理集中的所有代理,

找出满足以下 4 个条件的代理中地点资格值最大的代理: 代理没有分配给该角色 (所需代理数大于 1 时判断), 代理和已分配的任务、已分配的代理没有冲突, 代理所担任的角色数未达到最大值。第 11~14 行判断是否无法分配角色, 若存在可以分配的代理则如 15 行进行分配并更新分配矩阵 T 。最后完成所有循环后返回组效益最大的 T_{\max} 。

算法 3 PQGR 算法

输入 Tasks, P , A , AP_list[[]], minh, maxh

输出 T_{\max}

- 1) $F \leftarrow$ 根据 PQMQM (A , P , AP_list) 算法求得资格矩阵
- 2) For h in [minh, maxh]:
- 3) $R \leftarrow$ 根据 RPA(Tasks, h , P) 算法求得划分的角色集
- 4) $m \leftarrow A$ 中的代理数, $n \leftarrow R$ 中的角色数
- 5) $T \leftarrow$ 初始化值全为 0 的分配矩阵
- 6) For r in R :
- 7) For i in [0, 1, 2, ..., $L[r]$]:
- 8) For a in A :
- 9) 找到一个符合条件的地点资格值最大的代理
- 10) End for
- 11) If 没有符合条件的代理:
- 12) 恢复该角色已更改分配矩阵 T 的项
- 13) Break
- 14) End if
- 15) 将角色分配给代理并更新分配矩阵 T
- 16) End for
- 17) End for
- 18) End for
- 19) $T_{\max} \leftarrow$ 具有最大组效益的 T
- 20) Return T_{\max}

4.2 基于 GMRACRA 的解决方法

为使得高地点资格值的代理去完成高价值的角色, 本文提出了基于地点资格值的 GMRACRA 分配方法 (places-qualification-based GMRACRA algorithm, PQGM)。

定义 12 根据定义 1~11, 给定 Q 、 L 、 L^a 、 R^c 、 A^c 和 T , PQGM 算法是通过 GMRACRA 方法找到一个可行的 T 来实现最大组效益。问题定义如下:

$$\max \sigma' = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i, j] \times T[i, j] \quad (10)$$

$$T[i, j] \in \{0, 1\} \quad (0 \leq i < m) \quad (0 \leq j < n) \quad (11)$$

$$\sum_{i=0}^{m-1} T[i, j] = L[j] \quad (0 \leq j < n) \quad (12)$$

$$\sum_{j=0}^{n-1} T[i, j] = L^a[j] \quad (0 \leq i < m) \quad (13)$$

$$A^c[i_1, i_2] \times (T[i_1, j] + T[i_2, j]) \leq 1 \quad (0 \leq i_1 \neq i_2 < m, 0 \leq j < n) \quad (14)$$

$$R^c[j_1, j_2] \times (T[i, j_1] + T[i, j_2]) \leq 1 \quad (0 \leq i < m, 0 \leq j_1 \neq j_2 < n) \quad (15)$$

其中, 式 (10) 是问题的目标——最大化组效益, 式 (11) 是分配矩阵 T 取值的 0-1 约束, 式 (12) 表示按照角色所需的代理数量分配代理, 式 (13) 表示分配给代理的角色数不超过该代理的能力范围, 式 (14) 保证共同承担一个角色的两个或多个代理之间不会有冲突, 式 (15) 保证一个代理所承担的角色之间不会发生冲突。通过问题建模, CSSC 问题变成了一个 0-1 线性规划问题, 可以通过 IBM ILOG CPLEX 工具包 (CPLEX)^[24] 完成分配。

必要条件定义 在 GMRACRA 方法下 CSSC 问题有可行解的必要条件为各角色所需的代理之和要小于各代理的代理能力值之和, 如式 (16) 所示。在不考虑约束条件式 (14) 和式 (15) 的情况下满足这个条件才可以得到一个可行解。

$$\sum_{j=0}^{n-1} L[j] \leq \sum_{i=0}^{m-1} L^a[i] \quad (16)$$

算法 4 PQGM 算法

输入 Tasks, P , A , AP_list[[]], minh, maxh

输出 T_{\max}

- 1) $F \leftarrow$ 根据 PQMQM (A , P , AP_list) 算法求得资格矩阵
- 2) For h in [minh, maxh]:
- 3) $R \leftarrow$ 根据 RPA(Tasks, h , P) 算法求得划分的角色集
- 4) $m \leftarrow A$ 中的代理数, $n \leftarrow R$ 中的角色数
- 5) $T \leftarrow$ 初始化值全为 0 的分配矩阵
- 6) $Q = \text{GenerateQualificationMatrix}(F, R)$
- 7) $RC = \text{GenerateRoleConflictMatrix}(R)$
- 8) $AC = \text{GenerateAgentConflictMatrix}(A)$
- 9) $L = \text{GenerateRoleRangeVector}(R)$
- 10) $LA = \text{GenerateAbilityLimitVector}(A)$
- 11) $T = \text{doAssignment}(m, n, Q, AC, RC, L, LA)$
- 12) End for
- 13) $T_{\max} \leftarrow T$ which has best performance
- 14) Return T_{\max}

伪代码如算法 4 所示。输入输出与算法 3 中相同。其中第 1 行根据 h 取值范围进行遍历, 第

2~5行求得角色集 R 、地点资格值矩阵 F 、 m 、 n 和分配矩阵 T 。第6行GenerateQualificationMatrix(F , R)通过地点资格值矩阵 F 和角色集 R 生成大小为 $m \times n$ 的资格矩阵 Q 的函数。同理,第7~10行生成角色冲突矩阵 RC 、代理冲突矩阵 AC 、角色范围向量 L 和代理能力范围向量 LA ,其中 L 和 LA 的值符合式(16)的约束条件。第11行调用分配函数doAssignment(m , n , Q , AC , RC , L , LA)进行分配,得到分配矩阵 T 。最后循环后返回最优组效益的分配矩阵 T_{\max} 。流程图如图3所示。

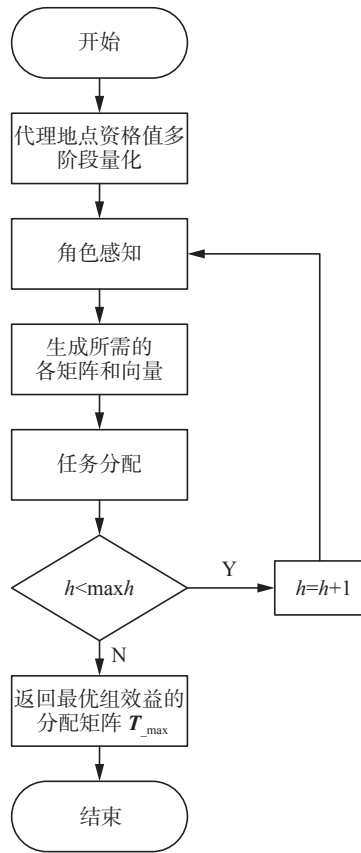


图3 PQGM 算法流程图

Fig. 3 Flow chart of PQGM algorithm

如表1所示存在15个订单 $t_1 \sim t_{15}$, 订单位置和所需的送达时间如表, 假定用户所选择的代收点为距离最近的代收点。订单横纵坐标范围为 $[0, 100]$, 时间以分钟定义, 范围是 $[0, 480]$, 如表2所示设置5个代收点 $P_1 \sim P_5$, 安排5个代理 $a_1 \sim a_5$ 进行分配。

首先, 对于每个代收点根据时间将任务划分形成如图4所示的角色(任务集), 角色分别为 $r_1 \sim r_5$ 。其中 r_1 包含任务 t_1 ; r_2 包含任务 t_2, t_8 ; r_3 包含任务 t_3, t_9, t_{10}, t_{11} ; r_4 包含任务 t_4, t_5, t_6, t_7 ; r_5 包含任务 $t_{12}, t_{13}, t_{14}, t_{15}$ 。然后根据随机生成的15天内代理对地点的访问时间序列生成如表3所示

的地点资格值矩阵, 进一步计算可得如表4所示的资格值 Q 矩阵。设置角色最小范围向量 $L = (1, 1, 1, 1, 2)$, $L^a = (8, 5, 6, 6, 5)$, 设置代理 a_1 和 a_2 冲突。调用本文提出的PQGM算法可得如表5所示的分配矩阵。

表1 订单的时间与位置分布

Table 1 Time and location distributions of orders

任务	时间/min	位置	价值/元
t_1	0	(32,70)	41
t_2	4	(21,22)	22
t_3	7	(82,13)	10
t_4	8	(55,85)	33
t_5	8	(57,93)	65
t_6	9	(65,75)	48
t_7	10	(43,65)	11
t_8	12	(27,45)	35
t_9	15	(83,4)	19
t_{10}	20	(99,21)	55
t_{11}	23	(94,41)	30
t_{12}	30	(34,59)	6
t_{13}	32	(30,8)	23
t_{14}	32	(28,1)	41
t_{15}	35	(44,14)	32

表2 收货地点的位置分布

Table 2 Location distributions of receiving places

P_1	P_2	P_3	P_4	P_5
(17,72)	(47,82)	(81,87)	(34,40)	(75,17)

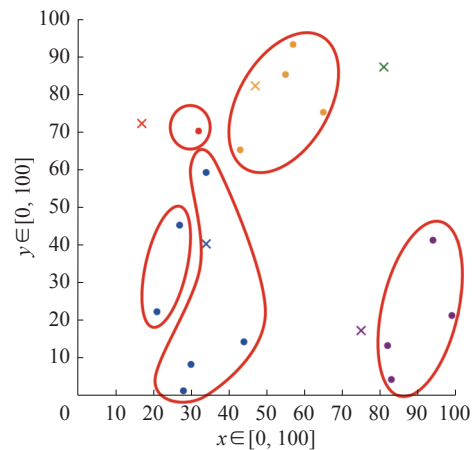


图4 订单的位置分布和角色划分

Fig. 4 Location distribution and role division of orders

表 3 代理的地点资格值矩阵

Table 3 The agent's place qualification value matrix

代理/地点	P_1	P_2	P_3	P_4	P_5
a_1	0.66	0.12	0.65	0.98	0.21
a_2	0.97	0.46	0.33	0.44	0.50
a_3	0.32	0.63	0.91	0.51	0.49
a_4	0.43	0.31	0.72	0.96	0.21
a_5	0.17	0.55	0.56	0.59	0.79

表 4 资格值 Q 矩阵Table 4 Q matrix of qualification values

代理/角色	r_1	r_2	r_3	r_4	r_5
a_1	27.06	55.86	23.94	18.84	99.96
a_2	39.77	25.08	57.00	72.22	44.88
a_3	13.12	29.07	55.86	98.91	52.02
a_4	17.63	54.72	23.94	48.67	97.92
a_5	6.97	33.63	90.06	86.35	60.18

表 5 PQGM 算法分配矩阵 T Table 5 Assignment matrix T of PQGM

代理/角色	r_1	r_2	r_3	r_4	r_5
a_1	1	0	0	0	1
a_2	0	1	0	0	0
a_3	0	0	0	1	0
a_4	0	0	0	0	1
a_5	0	0	1	0	0

5 实验与评估

5.1 实验设置

实验环境：处理器为 Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz 2.70 GHz, 内存容量为 8 GB, Windows 10 操作系统, JDK 为 Java 8。

本文实验采用的数据集为 gMission 数据集和遵循正态分布的合成数据集。gMission^[25] 是一个基于研究的通用空间众包平台, 时间范围为 $[0, 480]$ 。由于原数据量较大, 所以所用的数据集都是从原数据集中等间隔取得。如图 5、图 6 所示, 以工人和任务的半径为 1000 m 的数据集为例得出该数据集任务数量随时间变化趋势, 以及以半径为 1000 m 的数据集等间隔数取 1000 个任务统计 $t \in [120, 150)$, $t \in [240, 270)$, $t \in [360, 390)$ 3 个时间段任务的地点位置分布。由图可见时间分布和

位置分布较为均匀, 因此随机生成 5 个地点 $P_1 \sim P_5$ 作为货物代收点。

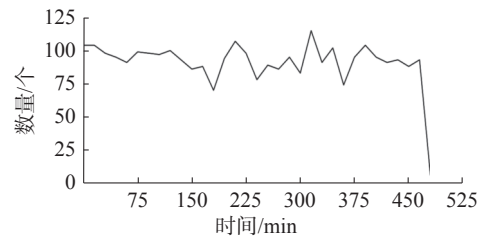


图 5 gMission 数据集中任务数量随时间变化趋势

Fig. 5 Trend of the number of tasks in gMission dataset over time

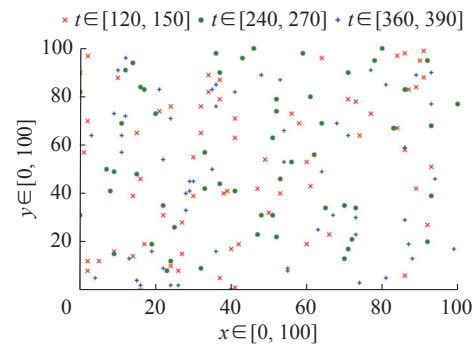


图 6 gMission 数据集中不同时间段任务地点分布

Fig. 6 Tasks location distribution in different time periods in the gMission dataset

实验变量设置如表 6 所示。另外, 以 5% 的冲突概率生成代理间的冲突矩阵, 以 90% 的概率设置角色所需的代理数为 1, 以 10% 的概率设置角色所需的代理数为 2 或 3, 假定任务选定的代收点为位置距离最近的代收点。

表 6 数据集变量设置

Table 6 Datasets variable settings

参数	设定值
gMission 任务数 $ T $	100, 200, 300, 400, 500
正态分布任务数 $ T $	500
工人数 $ W $	12
μ	15, 25, 45, 65, 85
σ	3.75
L^a	[8, 15]
T_{\max}	15
k_0	1.1
p_0	0.5
θ_0	3.5

对于合成数据集, 基于 gMission 真实数据集的观测, 实验中生成任务的位置和时间均匀分

布,位置产生在 100×100 的范围内,时间产生在 $[0, 480]$ 的范围内。由于 μ 往往对性能有更大的影响,任务价值遵循 $\mu=[15, 25, 45, 64, 85]$, $\sigma=3.75$ 的正态分布,其他变量的设置与 gMission 数据集中相同。由于 k_0 、 p_0 、 θ_0 的不同设置对该模型有着较大的影响, k_0 越大资格值函数下降速度越快, p_0 越大函数初始点越高。对于 θ_0 , 由于时间差 $t_n - t_{n-1}$ 越大, k_n 和 θ 越大, θ 越大反而使得 k_n 越小,因此 θ_0 决定了初始阶段工作强度对整个资格值的影响。基于本文分配问题的场景和激励的要求,经多次调试设置 $k_0 = 1.1$, $p_0 = 0.5$, $\theta_0 = 3.5$ 。

5.2 参数对算法的影响

核密度聚类算法中带宽 h 用作平滑参数,用来平衡结果中的偏差和方差。较大的带宽会使得概率密度曲线非常平滑,较小的带宽会使得曲线非常尖锐。

由于带宽设置不同时对聚类影响较大,现设置带宽取值 $[1, 20]$, 间隔为 1。对于带宽范围的限制包括以下 3 个方面:一是角色数量不可多到任务无解;二是控制同一角色内时差不超 60 min;三是随着带宽增加,角色数量无下降趋势。对角色划分后当任务数为 100~500 时应用 PQGM 算法分配后的组效益(组最优资格总值)如图 7 所示。

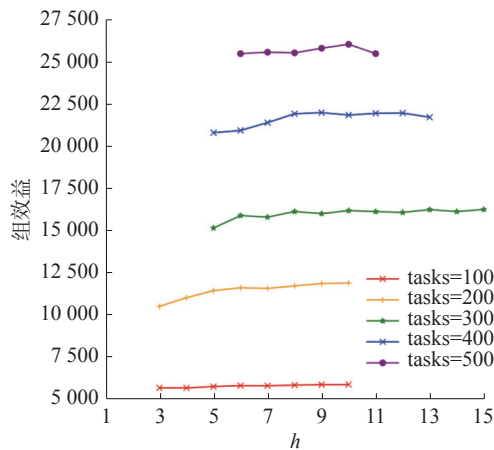


图 7 PQGM 算法组效益
Fig. 7 Group benefits of PQGM algorithm

由图 7 可得,应用 PQGM 算法,当任务数为 $[100, 200, 300, 400, 500]$ 时带宽 h 有效值为 $[3 \sim 10, 3 \sim 10, 5 \sim 15, 5 \sim 13, 6 \sim 11]$ 。带宽有效范围的下限由 3 逐渐增加到 6。开始下限为 3 是由于当任务数为 100、200 时小带宽角色划分不稳定导致超出角色内最大时差。下限逐渐增大是因为超出了 12 个工人能够担任的角色数,且随着任务数增多达到工人能完成的角色数范围所需的带宽逐渐增大。

另外由图 7 可看出,任务数为 100、200 时由

于单个角色包含的任务数增多易超出最大时差,带宽的最大限制相对较小。任务数从 100~300 增加时带宽的最大限制有增长的趋势,是因为任务数量增多时对于一个角色内最多 8 个任务的限制来说逐步不易超出最大时差。但 300~500 时带宽的最大限制逐步减小是由于对于较大的任务数带宽的增大对角色划分无明显改进作用。

由图 8 可得应用 PQGR 算法当任务数为 $[100, 200, 300, 400, 500]$ 时,不同带宽组效益的变化趋势。图 8 与图 7 相比,由于 PQGR 算法在任务数量较多时虽然任务完成率不为 100%,但不存在无解的情况,因此较小带宽也存在相对应的组效益但值较小。任务数为 100、200 时小带宽对应的组效益缺失同样是由于超出角色内最大时差。

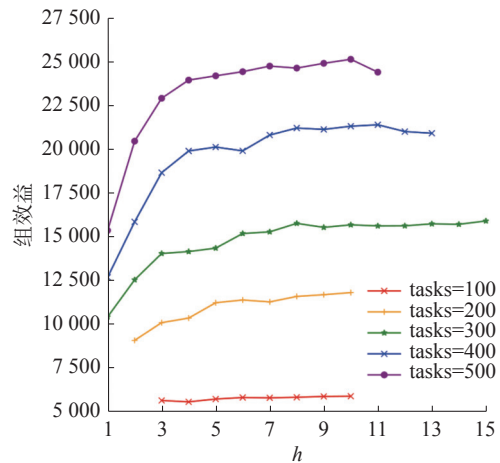


图 8 PQGR 算法组效益
Fig. 8 Group benefits of PQGR algorithm

综合上述 2 种方法组效益的变化幅度和范围,由于波动较小且为了降低核密度聚类划分角色的不稳定性,带宽取值范围均为 $[1, 20]$,聚类后不符合限制条件的不进行下一步分配。

5.3 算法比较

本节在 gMission 数据集和合成数据集上进行了实验,对比算法为没有通过 RPA 算法划分角色,而是在时间范围内按顺序随机聚类的 PQGM 和 PQGR 算法,将 4 种算法进行对比,验证了算法的有效性和可行性。

图 9 是任务数为 100~500 时应用 4 种算法的组效益。可以看出 PQGM 算法效果相较于另外 3 种算法来说效果较好,且随着任务数量的增加优势逐渐增大,没有应用 RPA 的 PQGR 算法效果最差,另外两个相差不大。由于是循环取最优值,4 种算法的任务完成率都较高, PQGM 系列算法都为 100%, PQGR 系列算法在 98%~100%。

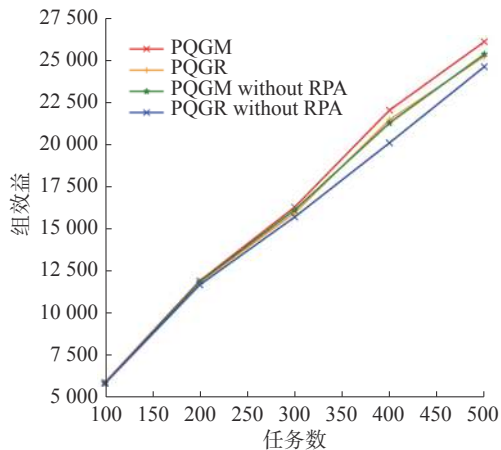


图9 gMission数据集组效益

Fig. 9 Group benefits of gMission datasets

图10给出了算法的运行时间, 可以从图中看到, 没有运用角色感知算法(RPA)的2个算法因为只需执行一次, 运行时间较短, 其值维持在0和1附近, 但没有RPA划分角色的PQGM算法不能保证所产生的任务数对于当前代理集有解。PQGR算法运行时间是单调递增的且低于PQGM算法, 随着任务数增多两者差距增大。PQGR算法运行时间维持在25 s以内, PQGM算法时间维持在35 s以内。

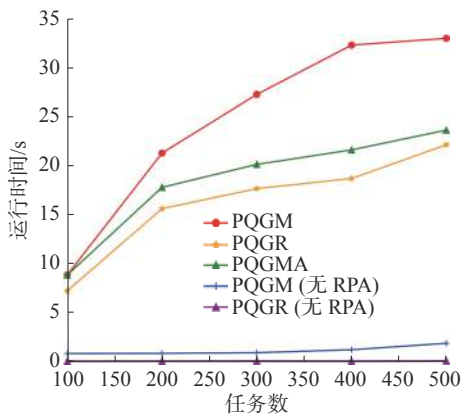


图10 gMission数据集运行时间

Fig. 10 Running time of gMission datasets

经过多次实验发现, 应用PQGM算法求解时, 由于代理数不变, 问题从无解过渡到有解时, 角色数量是有一定规律的, 本文称这个数量范围为过渡范围。本文实验设置中代理数为12, 工人一天所能承担的角色范围为[8, 15]时过渡范围约为[100, 120], 因此可以根据角色数量规避掉确定无解的数据点, 缩短总的任务运行时间。本文以120为边界值, 形成PQGM-adjusted(PQGMA)方法, 如图10所示明显缩短了运行时间。

本文分别对2个目标——地点资格值和价值进行对比。由于每个角色地点资格值取值范围为

(0, 1)、价值不定, 每个任务价值取值(0, 100], 因此计算每个角色的地点资格值均值和每个任务的价值均值。地点资格值均值如图11所示, 随着任务数增多, 所有算法的均值都呈现下降趋势, PQGM算法均值最大, 没有RPA划分角色的PQGR算法均值最小。由于任务完成率较高, 4种算法的价值均值差距较小, 当任务数较大时PQGM类算法有优势。

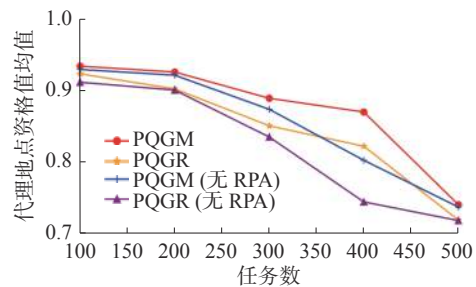


图11 gMission数据集地点资格值均值

Fig. 11 Mean location qualification value of gMission datasets

另外, 受Zhu等^[22]代理分类算法中限制每个类别代理数量的启发, 实验应用Zhu提出的AC_SAGRA+算法的变体(简称GAC)进行任务划分生成角色集。当任务数为100时, 将利用核密度聚类算法和GAC算法进行任务划分的PQGM算法和PQGR算法进行比较, 如表7所示, GAC划分后的角色所得组效益略优于核密度聚类所得组效益, 但由于其模拟退火迭代次数较多, 运行时间较长, 整体来说不如核密度聚类效果好。

表7 算法对比
Table 7 Algorithm contract

算法	角色数	时间/s	PQGM	PQGR
核密度	40	0.25628	5883.14	5860.88
GAC	41	578.877	5889.57	5874.15

对于正态分布任务数为500, 任务价值以 $\mu=[15, 25, 45, 64, 85]$, $\sigma=3.75$ 的正态分布生成的数据集分析, 没有RPA数据处理的PQGMA算法产生的角色集不一定有解。如图12所示, 随着 μ 值的增加4种算法的组效益都在增加, 差距逐渐增大, PQGMA算法的优势明显。运行时间如图13所示, 当任务数为500时, 4种算法的运行时间在一定范围内波动。地点资格值均值与价值均值与真实数据集相仿。如图14所示PQGMA类算法任务完成率为100%, PQGR类算法任务完成率位于98%~100%。

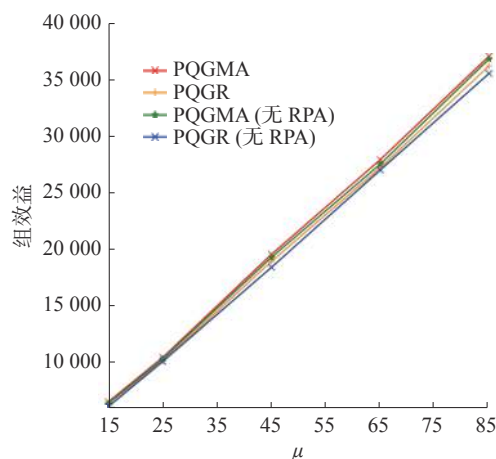


图 12 正态分布数据集组效益

Fig. 12 Group benefits of normally distributed datasets

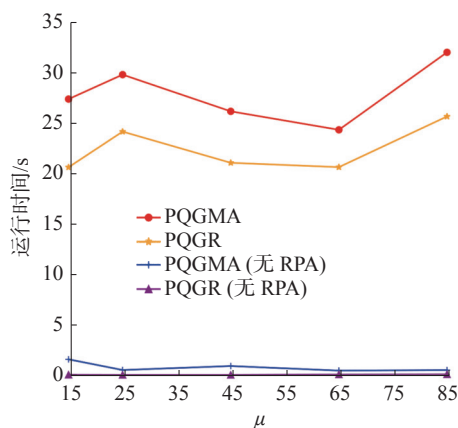


图 13 正态分布数据集运行时间

Fig. 13 Running time of normally distributed datasets

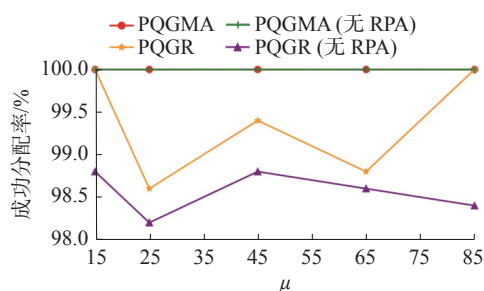


图 14 正态分布数据集任务完成率

Fig. 14 Task completion rate of normally distributed datasets

6 结束语

本文首次提出了针对社区服务型时空众包 (CSSC) 模式的任务分配问题并将其通过 RBC 形式化表达。为实现派遣高地点资格值代理配送高价值量订单集的目标, 任务分配上提出基于地点资格值的 PQGR 算法、PQGM 算法和 PQGMA 算法。在数据处理和量化方面: 提出基于核密度估计聚类的角色感知方法; 提出代理资格值动态变

化的代理地点资格值多阶段量化模型, 对于资格值进行在线学习。最后在 gMission 数据集和合成数据集上进行了实验, 验证了算法的有效性和效率。在未来的工作中, 可以继续尝试其他算法去优化聚类效果, 加入其他需要考虑的因素平衡角色数量和组效益的关系。

参考文献:

- [1] ZHU Haibin, ZHOU Mengchu. Role-based collaboration and its kernel mechanisms[J]. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, 2006, 36(4): 578–589.
- [2] SHENG Yin, ZHU Haibin, ZHOU Xianzhong, et al. Effective approaches to adaptive collaboration via dynamic role assignment[J]. *IEEE transactions on systems, man, and cybernetics: systems*, 2016, 46(1): 76–92.
- [3] ZHU Haibin. Adaptive collaboration systems: self-sustaining systems for optimal performance[J]. *IEEE systems, man, and cybernetics magazine*, 2015, 1(4): 8–15.
- [4] ZHU Haibin, HOU Ming, ZHOU Mengchu. Adaptive collaboration based on the E-CARGO model[J]. *International journal of agent technologies and systems*, 2012, 4(1): 59–76.
- [5] ZHU Haibin, LIU Dongning, ZHANG Siqin, et al. Solving the Many to Many assignment problem by improving the Kuhn-Munkres algorithm with backtracking[J]. *Theoretical computer science*, 2016, 618: 30–41.
- [6] ZHU Haibin, SHENG Yin, ZHOU Xianzhong, et al. Group role assignment with cooperation and conflict factors[J]. *IEEE transactions on systems, man, and cybernetics: systems*, 2018, 48(6): 851–863.
- [7] ZHU Haibin, ZHOU Mengchu, ALKINS R. Group role assignment via a Kuhn-munkres algorithm-based solution[J]. *IEEE transactions on systems, man, and cybernetics-part A: systems and humans*, 2012, 42(3): 739–750.
- [8] ZHU Haibin, LIU Dongning, ZHANG Siqin, et al. Solving the group multirole assignment problem by improving the ILOG approach[J]. *IEEE transactions on systems, man, and cybernetics: systems*, 2017, 47(12): 3418–3424.
- [9] ZHU Haibin. Avoiding conflicts by group role assignment[J]. *IEEE transactions on systems, man, and cybernetics: systems*, 2016, 46(4): 535–547.
- [10] ZHU Haibin. Group multi-role assignment with conflicting roles and agents[J]. *IEEE/CAA journal of automatica sinica*, 2020, 7(6): 1498–1510.
- [11] LIU Dongning, HUANG Baoying, ZHU Haibin. Solving the tree-structured task allocation problem via group multirole assignment[J]. *IEEE transactions on automation sci-*

- ence and engineering, 2020, 17(1): 41–55.
- [12] JIANG Qian, ZHU Haibin, QIAO Yan, et al. Agent evaluation in deployment of multi-SUAVs for communication recovery[J]. IEEE transactions on systems, man, and cybernetics: systems, 2021, 52(11): 6968–6982.
- [13] ZHU Haibin. Maximizing group performance while minimizing budget[J]. IEEE transactions on systems, man, and cybernetics: systems, 2020, 50(2): 633–645.
- [14] ZHU Haibin, HOU Ming, WANG Chun, et al. An efficient outpatient scheduling approach[J]. IEEE transactions on automation science and engineering, 2012, 9(4): 701–709.
- [15] JIANG Qian, LIU Dongning, ZHU Haibin, et al. Equilibrium means equity? an E-CARGO perspective on the golden mean principle[EB/OL]. (2022–03–05)[2022–05–01]. <https://ieeexplore.ieee.org/document/9732178>.
- [16] 钱晨, 张以文, 吴其林, 等. 基于 E-CARGO 模型的云制造服务组合方法 [J]. 小型微型计算机系统, 2020, 41(9): 2006–2011.
- QIAN Chen, ZHANG Yiwen, WU Qilin, et al. Cloud manufacturing service composition method based on E-CARGO model[J]. Journal of Chinese computer systems, 2020, 41(9): 2006–2011.
- [17] 李晓会, 董红斌. 基于 E-CARGO 模型的共乘出行匹配建模与优化方法 [J]. 计算机应用, 2022, 42(3): 778–782.
- LI Xiaohui, DONG Hongbin. Modeling and optimization method of ride-sharing matching based on E-CARGO model[J]. Journal of computer applications, 2022, 42(3): 778–782.
- [18] 李炜, 吴群群, 张以文. 基于 E-CARGO 模型的开发推荐方法 [J]. 计算机应用, 2022, 42(2): 557–564.
- LI Wei, WU Qunqun, ZHANG Yiwen. Developer recommendation method based on E-CARGO model[J]. Journal of computer applications, 2022, 42(2): 557–564.
- [19] ZHU Haibin. Agent categorization with group role assignment with constraints and simulated annealing[J]. IEEE transactions on computational social systems, 2020, 7(5): 1234–1245.
- [20] HUANG Baoying, ZHU Haibin, LIU Dongning, et al. Solving last-Mile logistics problem in spatiotemporal crowdsourcing via role awareness with adaptive clustering[J]. IEEE transactions on computational social systems, 2021, 8(3): 668–681.
- [21] JIANG Qian, LIU Dongning, ZHU Haibin, et al. Quasi group role assignment with role awareness in self-service spatiotemporal crowdsourcing[J]. IEEE transactions on computational social systems, PP(99): 1–13.
- [22] PENG Chaoqun, ZHANG Xinglin, OU Zhaojing, et al. Task planning considering location familiarity in spatial crowdsourcing[J]. ACM transactions on sensor networks, 2021, 17(2): 16.
- [23] 于洪涛, 崔瑞飞, 董芹芹. 基于遗忘曲线的微博用户兴趣模型 [J]. 计算机工程与设计, 2014, 35(10): 3367–3372, 3379.
- YU Hongtao, CUI Ruifei, DONG Qinqin. Micro-blog user interest model based on forgetting curve[J]. Computer engineering and design, 2014, 35(10): 3367–3372, 3379.
- [24] IBM. IBM ILOG CPLEX Optimization Package. [EB/OL]. (2021–01–01)[2022–05–01]. <http://www-01.ibm.com/software/integration/optimization/cplexoptimization-studio/>.
- [25] CHEN Zhao, FU Rui, ZHAO Ziyuan, et al. gMission[J]. Proceedings of the VLDB endowment, 2014, 7(13): 1629–1632.

作者简介:



徐静如, 硕士研究生, 主要研究方向为时空众包、群体智能。



董红斌, 教授, 博士生导师, 中国计算机学会高级会员, 主要研究方向为机器学习、人工智能、多智能体系统和数据挖掘。主持和完成国家自然科学基金、工信部基础研究项目、黑龙江省自然科学基金项目, 荣获黑龙江省高校科学技术奖和黑龙江省优秀高等教育科学成果奖。主编教材 2 部。发表学术论文 90 余篇。



赵炳旭, 博士, 主要研究方向为时空众包、移动计算和人工智能。