

空间众包中可拒绝情况下的在线任务分配

林荟荟¹, 黄杰¹, 李玉², 万健¹

(1. 浙江科技学院 信息与电子工程学院, 杭州 310023; 2. 杭州电子科技大学 计算机学院, 杭州 310018)

摘要: 现实中空间众包任务可能会遭到任务执行者的拒绝,为解决该问题提出动态可拒绝的空间众包处理方法。首先,为降低被拒绝的概率,运用主成分分析法(principal components analysis,PCA)计算任务执行者对任务的兴趣度;然后针对任务动态分配问题采用批处理模式解决,提出基于 MaxFlow 的排序算法(sequence algorithm base on MaxFlow,SMF)和基于 KM 算法(Kuhn-Munkres,KM)的不重复构造交替树算法(non-repetitive construction of alternating tree algorithm based on KM,NR-KM)寻找全局最大匹配下最高兴趣度分配方案;最后将贪心算法(greedy algorithm)、KM 算法和 SMF 算法作为对照算法,与 NR-KM 算法在 CPU 时间成本、任务分配数量和任务分配兴趣度 3 个方面进行比较。结果表明 NR-KM 算法相对于 KM 算法、SMF 算法在分配效率上分别提高 11% 和 9%。可见,NR-KM 算法能高效解决可拒绝情况下空间众包任务的分配,对解决空间众包任务涉及执行者意愿的分配问题具有参考价值。

关键词: 空间众包;兴趣度预测;在线任务分配

中图分类号: TP312

文献标志码: A

文章编号: 1671-8798(2021)03-0227-07

Online task assignment under rejectable conditions in spatial crowdsourcing

LIN Huihui¹, HUANG Jie¹, LI Yu², WAN Jian¹

(1. School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, Zhejiang, China; 2. School of Computer and Software, Hangzhou Dianzi University, Hangzhou 310018, Zhejiang, China)

Abstract: In reality, spatial crowdsourcing tasks may be rejected by task executors. To solve this problem, a dynamic and rejectable spatial crowdsourcing method was proposed. First, in order to lower the probability of rejection, the principal components analysis(PCA) was used to calculate the interest of the executor in the task; then, the batch processing mode was adopted to solve the problem of task dynamic allocation, and sequence algorithm based on MaxFlow (SMF) and non-repetitive construction of alternating tree algorithm based on KM(NR-KM)

收稿日期: 2021-01-09

基金项目: 国家自然科学基金项目(61972358)

通信作者: 万健(1969—),男,福建省泉州人,教授,博士,主要从事云计算及大数据研究。E-mail:wanjian@zust.edu.cn。

were proposed to find the highest interest allocation scheme under the global maximum matching; finally, greedy algorithm, KM algorithm(Kuhn-Munkres) and SMF algorithm were used as the control algorithm to compare the CPU time cost, task assignment quantity and task assignment interest with the NR-KM algorithm. The results show that the NR-KM algorithm improves the allocation efficiency by 11% and 9%, respectively, compared with KM algorithm and SMF algorithm. It can be seen that the NR-KM algorithm efficiently solves the assignment of spatial crowdsourcing tasks under rejectable conditions, and has reference value for solving the assignment problem of spatial crowdsourcing tasks involving the executor's will.

Keywords: spatial crowdsourcing; interest prediction; online task assignment

随着移动网络的发展和移动设备的普及,空间众包越来越受到人们的关注^[1],任务分配问题是空间众包的核心问题^[2-4]。目前学术界主要讨论如何将任务高效地分配给任务执行者,同时对其分配的优化目标主要为最大化整体任务分配数量^[5-6]。Hien 等^[7]扩展了分配模型,为每个任务执行者-任务对设置权重,并以最大化总任务分配权重为目标。但是现实中任务执行者和任务是动态变化的,该模型仍然无法实现在整个时间线上最大化任务分配数量的目标。Chen 等^[8]提出一种基于预测的任务分配算法,然后用贪心算法最大化当前与未来的任务分配数量。但通过采样的方式进行预测,在时间和空间上预测准确度不高。

以上研究都是基于平台分配任务,任务执行者保证执行空间众包任务。但现实中每个任务执行者对任务的选择具有主观性,任务执行者都希望能执行符合他们意愿的任务,比如在自己所处位置附近^[9]、任务报酬高^[10]的任务,或是与自己的知识、技能相匹配的任务^[11]。Zheng 等^[12]首次定义有拒绝的空间众包问题,提出 4 种近似分配算法来提高分配的效率,但采用的是静态分配的方法得到最优分配。本研究基于平台分配任务的方式(server assigned tasks, SAT)^[13-14]探讨空间众包中任务执行者可拒绝情况下的在线任务分配问题^[15],实现提出动态可拒绝的空间众包处理方法:首先最大化任务分配数量,即实现最大匹配;然后,针对任务执行者的主观性,为减少任务执行者拒绝的概率,寻找全局最大匹配下最高兴趣度分配方案。

1 问题定义与处理方法

现有研究^[16]已证明空间众包全局最优任务分配是非确定性多项式难题(non-deterministic polynomial hard, NP-hard)。利用批处理模式解决空间众包中任务执行者可拒绝情况下的在线任务分配问题,给定集合 $p = \{1, 2, 3, \dots\}$ 表示整个时间段,其中每个时间片 p 上存在不同的任务执行者和任务集合,并且在每个时间片内的任务执行者和任务信息均为已知。

1.1 动态移动的任务执行者

假设集合 $W_p = \{w_1, w_2, \dots, w_m\}$ 是在某时间片上 m 个移动任务执行者的集合, $w_i \in W_p$ 是其中一名任务执行者,可形式化定义为 $w_i = \langle l_i, p_{ij}, r_i \rangle (1 \leq i \leq m)$, 其中 l_i 为在时间片 p 时任务执行者的位置; p_{ij} 为任务执行者 i 对任务 j 的兴趣度; r_i 为任务执行者可达任务范围。本文中一个任务执行者最多只能完成一个任务。

1.2 有约束的任务

假设集合 $T_p = \{t_1, t_2, \dots, t_n\}$ 是在某时间片上 n 个有时间、位置约束的任务集合, $t_j \in T_p$ 是其中一个任务,可形式化定义为 $t_j = \langle s_j, l_j, d_j, m_j, b_j \rangle (1 \leq j \leq n)$, 其中 s_j 为任务开始时间; l_j 为任务的地点; d_j 为任务时长; m_j 为任务移动距离; b_j 为任务价格。每个任务最多被一个任务执行者执行,并且任务执行者只接受在其可达范围内的任务。

1.3 最大匹配下最高兴趣度问题

给定任务执行者集合 W 和任务集合 T , 在任意时间片上有 m 个任务执行者, n 个任务。满足时间、空间约束条件下,任务执行者 w_i 对空间任务 t_j 有兴趣,兴趣度权重为 p_{ij} , 任务执行者-任务分布图如图 1(a) 所示。图 1(b) 表示任务执行者和任务的待分配情况,每个任务执行者、每个任务表示为二分图中的两边

顶点,其中任务执行者和任务的顶点类型不同。任务执行者顶点 w_i 和空间任务顶点 t_j 之间存在一条边,如此,任务执行者-任务对 $\langle w_i, t_j \rangle$ 是有效的。

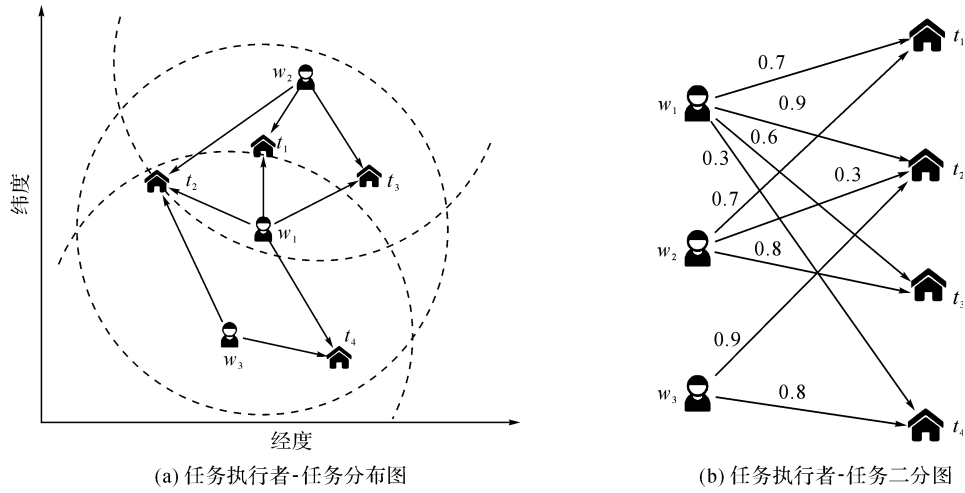


图1 任务执行者-任务分布图和二分图

Fig. 1 Executor-task distribution and bipartite graph

定义1 任务分配数量:在一个时间段内有任务执行者集合 W 和任务集合 T ,任务分配策略要求被分配执行的任务数量最大。

定义2 任务分配兴趣度:假设任务 t_j 分配给任务执行者 w_i ,则 p_{ij} 为有效分配兴趣度,任务分配策略要求在任务最大化分配的情况下,分配兴趣度最大。

定义3 最大匹配下最高兴趣度问题:结合定义1和定义2,在给定时间段 $P = \{1, 2, 3, \dots\}$ 中有 p 个时间片,设 n_p 为某时间片上被分配的任务数量, S_p 为该时间片上任务分配兴趣度,则该时间段内的任务总分配兴趣度分数为 $S = \{ \max \sum_{p=1}^{|P|} n_p \mid \sum_{p=1}^{|P|} S_p \}$ 。

1.4 动态可拒绝的空间众包处理方法

本研究提出的动态可拒绝空间众包处理方法如图2所示。任务执行者和任务动态提交信息至空间众包平台,平台首先划分时间片,查看时间片内的任务与任务执行者是否满足时间和空间约束,为降低任

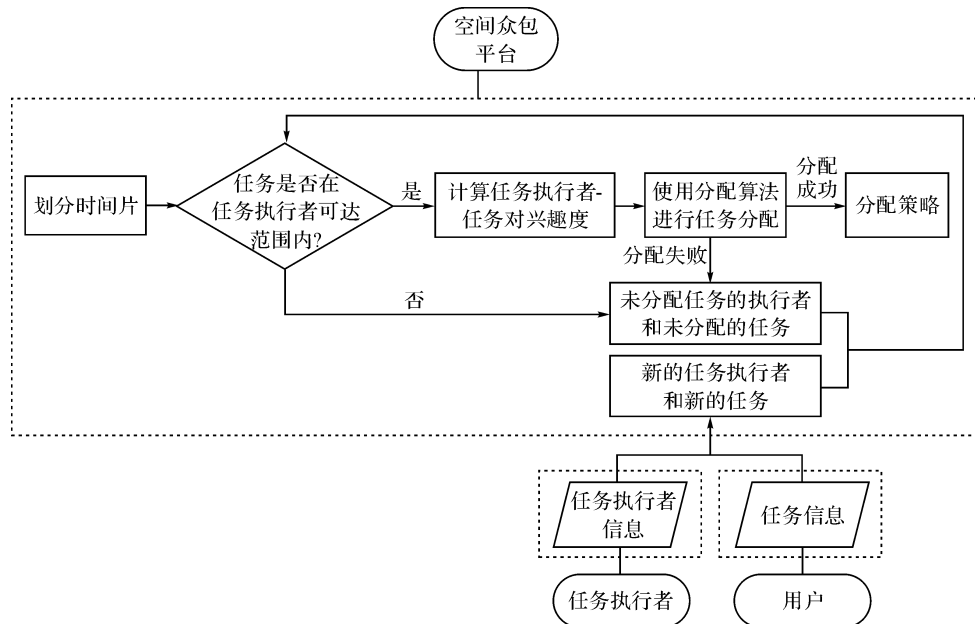


图2 动态可拒绝空间众包处理方法

Fig. 2 Dynamic rejectable spatial crowdsourcing framework

务执行者拒绝的可能性,计算任务执行者-任务对兴趣度,再使用分配算法进行任务分配。若任务分配成功,则将分配策略发送给任务执行者让其执行;若分配失败,当前时间片内未分配任务的执行者和未分配的任务将与下一时间片内任务执行者和任务一起等待下一次任务分配。

2 动态可拒绝的空间众包分配方法

2.1 兴趣度预测

利用主成分分析法(principal components analysis, PCA)^[17]预测任务执行者对任务的兴趣度。首先构建兴趣度计算指标体系,将任务执行者和任务之间的距离^[18]作为负向指标,将任务移动距离^[19]、任务时长^[19]、任务价格^[20]作为正向指标;然后构建初始化矩阵,归一化处理后得到标准化矩阵;继而计算标准化矩阵得到相关系数矩阵,并计算对应的特征值、特征向量;接着计算每个主成分的方差贡献率,确定主成分个数;最后确定主要评估指标的权重,并进行归一化修正,得到任务执行者对任务的兴趣度。具体计算过程如下:

获取一定时间段内任务执行者的任务数据并做归一化处理,得到归一化矩阵 \mathbf{Z} 。得到 p 维的随机变量后,构造样本的协方差矩阵 $\mathbf{R} = [\mathbf{r}_{ij}]_{p \times p} = \frac{\mathbf{Z}^T \mathbf{Z}}{n-1}$ 。其中 $r_{ij} = \frac{\sum \mathbf{Z}_{kj} \mathbf{Z}_{ki}}{n-1}, i, j = 1, 2, \dots, p$, 解样本协方差矩阵 \mathbf{R} 的特征方程 $|\mathbf{R} - \lambda \mathbf{I}_p| = 0$ 得 p 个特征根,将特征值由大到小排列,得到 $\lambda_1, \lambda_2, \dots, \lambda_p$, 对应的特征向量为 $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_p$, 特征向量 \mathbf{e} 满足 $|\mathbf{e}| = 1$, 每个特征值对应一个主成分,每个特征向量包含 p 个元素。

计算每个主成分的方差贡献率 $v, v_a = \lambda_a / (\lambda_1 + \lambda_2 + \dots + \lambda_p), a$ 为正整数, $1 \leq a \leq p, v_a$ 表示第 a 个主成分的方差贡献率, λ_a 表示第 a 个主成分对应的特征值。

按照特征值大于 1、累计方差贡献率大于指定 0.85 的原则,提取满足条件的特征值个数作为最终选择的主成分个数;如果满足条件的特征值个数为 b , 选择 b 个主成分, $\lambda_1, \lambda_2, \dots, \lambda_b$ 为 b 个主成分分别对应的特征值,其分别对应的特征向量 $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_b$ 为 b 个主成分的特征向量,特征向量和标准化后的数据相乘,得到主成分的线性表达式:

$$Y_c = \mathbf{e}_{c1} \mathbf{Z}_1 + \mathbf{e}_{c2} \mathbf{Z}_2 + \dots + \mathbf{e}_{cp} \mathbf{Z}_p. \quad (1)$$

式(1)中: Y_c 为第 c 个主成分, c 为正整数, $1 \leq c \leq b; \mathbf{e}_{cp}$ 为某个指标在第 c 个主成分线性表达式中的系数,表示第 c 个特征向量中的第 p 个元素; \mathbf{Z}_p 为第 p 个指标经过标准化处理后的值。

以主成分的方差贡献率为权重,对主要评估指标在各个主成分线性表达式中的系数进行加权平均,计算每个主要评估指标的综合权重;将所有主要评估指标的综合权重进行归一化,得到每个主要评估指标的权重值 w', w'_j 表示第 j 个主要评估指标的综合权重除以所有主要评估指标的综合权重之和,根据获得的权重值进行加权计算,得到每个任务执行者的兴趣度并映射到 $[0, 1]$ 之间。

2.2 动态分配算法

Hamamda 等^[21]验证了批处理模式在任务分配上的有效性,故本研究采用批处理的模式解决动态分配问题。

2.2.1 基于 MaxFlow 的排序算法

采用基于 MaxFlow 的排序算法(sequence algorithm base on MaxFlow, SMF)求解分配策略。建立一个额外的源点 Start 和一个额外的汇点 End,源点与顶点连接一条容量 c 为 1、权重为 0 的弧,汇点也执行类似的操作;两边顶点任务执行者 w_i 和任务 t_j 的弧容量 c 设置为 1,权重设置为兴趣度 $p_{i,j} \in [0, 1]$;完成网络 $G = (V, E, C, p)$ 构建。

通过增广路算法来求得该网络的最大流量。该算法优点是每次调整流量都遵循最短路径的原则,直到找不到增广路,此时的总流量和兴趣度分数即为所求答案。但是在寻找最大兴趣度增广路的时候需先求出最短路径,已知增流到最后的流量 $n = \{\min(|W_p|, |T_p|)\}$,针对有多条弧的权重

为0的特性,只需对二分图中的权重按照从小到大的顺序排列,再根据排序作为有向路径增流^[22]至 n 值或不存在可增广路为止,如此就找到最小兴趣差最大流路径,转换后也就是找到最大兴趣度最大流路径。

2.2.2 基于KM的不重复构造交替树算法

实际构建二分图的时候,任务执行者与任务数不一定相同。若两边顶点数不一致时,补齐较少边的集合,并将其权重设为0。传统KM算法(Kuhn-Munkres,KM)可求解最大匹配最高兴趣度问题,通过不断修改可行顶标得到相等子图进行增广路探索,直到找到最大匹配为止。一般对其优化都是给每个右顶点加一个“松弛量”函数 s ,每次开始寻找增广路时初始化为无穷大。在寻找增广路时,检查任务执行者 w_i 和任务 t_j 的边,如果不在相等子图中,让 $s[j] = \min\{s[j], l_w[i] + l_t[j] - p[i][j]\}$ 。这样在修改顶标时,取所有不在交错树中的任务集顶点的 s 值中的最小值作为 d 值即可。但是每轮匹配过的顶点会被清除,需要重新开始搜索建立交替树。

本文提出基于KM的不重复构造交替树算法(non-repetitive construction of alternating tree algorithm based on KM, NR-KM),在原来树的基础上接着搜索新加入的边。设顶点 w_i 的顶标为 $l_w[i]$,顶点 t_j 的顶标为 $l_t[j]$,顶点 w_i 与 t_j 之间的权重为 $p[i][j]$ 。定义 $s[i]$ 为原值和 $l_w[i] + l_t[j] - p[i][j]$ 的较小值,取所有不在交错树中的 $s[i]$ 值作为 d 。每次减掉 d 后, s 变为0的点肯定有新加入的边,所以不重新搜索原来的树,从这些点开始接着原树搜索,直到交错树扩展到存在增广路为止。

3 试验结果与分析

使用滴滴盖亚数据计划^[23]中西安二环局部区域的滴滴快专车平台的订单司机轨迹数据。在数据中选取的事件发生在东经 $108.921\ 859^\circ \sim 109.009\ 348^\circ$,北纬 $34.204\ 946^\circ \sim 34.279\ 936^\circ$,轨迹点的采集间隔是2~4 s,轨迹点经过绑路处理,以保证数据都能够对应到实际的道路信息。采用某一天滴滴快专车平台的订单司机轨迹数据,有119 019个订单,共涉及178 56个任务执行者。在试验研究中,由于任务执行者和任务有属性约束,需补充数据集属性。通过原始真实数据集计算出任务时长(unix时间戳形式)、任务移动距离(换算成两地之间的距离);根据滴滴快车订单计价规则,计算任务价格。

试验主要考虑时间片 p 和任务执行者可达范围 r 对任务分配算法的影响。时间片范围的设定在一定程度上影响该时间片内任务执行者与任务的数量,间接影响分配结果;任务执行者可达范围直接影响任务能否被任务执行者执行,决定任务执行者-任务有效对的数量,影响分配结果。本研究设置的试验参数见表1(参数数值加粗的为默认值),对每个参数变化的结果都进行50组的试验,取50组试验结果的平均值。所有试验均在一台处理器为Intel(R)Core(TM)i7-7700,CPU为3.60 GHz,内存为16 GB的计算机上进行。

表1 试验参数

Table 1 Experimental parameters

参数	数值
时间片/s	120,180, 240 ,300
任务执行者可达范围/km	1,2, 3

为了评估任务分配算法的性能,用3个指标来对比贪心算法(greedy algorithm,GA)、KM算法和我们提出的SMF和NR-KM算法。3个指标为:1)CPU时间成本,即整个时间段内任务分配的CPU时间成本;2)任务分配数量,即整个时间段内任务分配的数量;3)任务分配兴趣度,即在整个时间段内任务分配兴趣度之和。

在默认参数的情况下,4种分配策略的性能比较见表2。GA算法得到整个时间段内分配近似解,没能从全局考虑得到最优解;SMF算法、KM算法和NR-KM算法得到的任务分配数量和兴趣度分数是相同的。NR-KM算法时间效率最优,相比SMF算法速度提高9%,由于重标号每次对一个边进行扫描操作,而SMF算法需要维护标号和队列操作;NR-KM算法因不需要每次重新构建交错树,相比KM算法分配速度平均快11%。

表 2 4 种分配策略的性能比较

Table 2 Performance comparison of four allocation strategies

算法	CPU 时间成本/s	任务分配数量/个	任务分配兴趣度
GA	454.88	68 672	52 815
SMF	410.08	76 704	63 189
KM	429.28	76 704	63 189
NR-KM	386.32	76 704	63 189

时间片大小对算法分配时间效率的影响如图 3(a) 所示,在任务执行者可达范围为默认值的情况下,时间片的大小直接导致任务执行者和任务数量的不同,当时间片较小时,4 种算法的分配能力相差不大;但随着时间片的增大,任务执行者和任务数量在一定程度上增多,因此所有算法 CPU 时间成本增加。任务执行者可达范围对算法分配时间效率的影响如图 3(b) 所示,在时间片以默认值划分的情况下,当任务执行者可达范围较小时,有效任务执行者-任务对有限,不能体现本文算法的优势;但随着任务执行者可达范围的增大,任务执行者可分配的任务数量增加,相应的所有算法的 CPU 时间成本都增加,相比之下,本文算法性能最优。

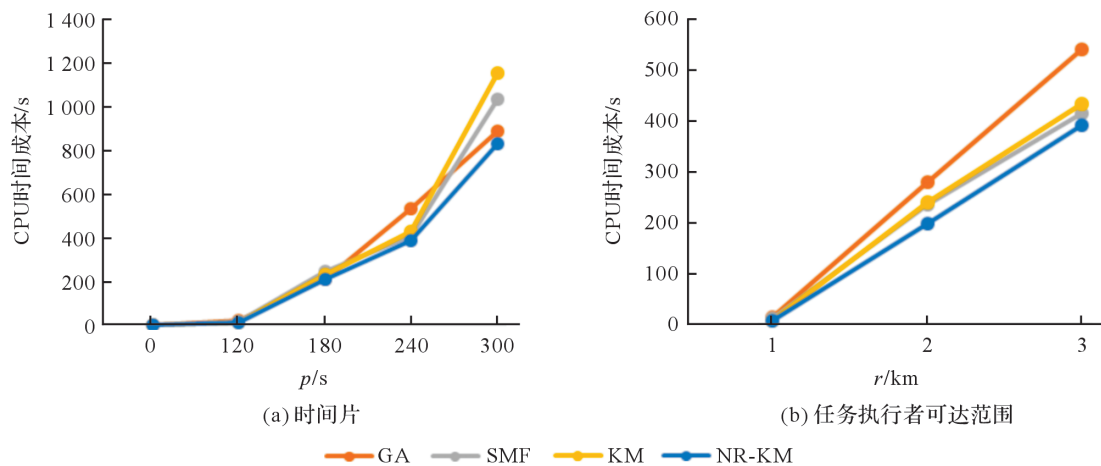


图 3 时间片和任务执行者可达范围对算法分配时间效率的影响

Fig. 3 Effect of time slices and range of executor on allocated time efficiency of NR-KM

时间片和任务执行者可达范围对算法任务分配数量的影响如图 4 所示,时间片和任务执行者可达范围对算法任务分配兴趣度的影响如图 5 所示。SMF、KM 和 NR-KM 3 种算法分配的结果是一致的,这从侧面反映了算法寻找分配结果的正确性。在时间片比较小或任务执行者可达范围比较小的时候,可供选择的任务-任务执行者分配是有限的,故这 3 个算法与 GA 算法得到的任务分配数量与任务分配兴趣度差别不大;但随着时间片和可达范围参数值的增大,逐渐显现出 GA 算法的劣势。

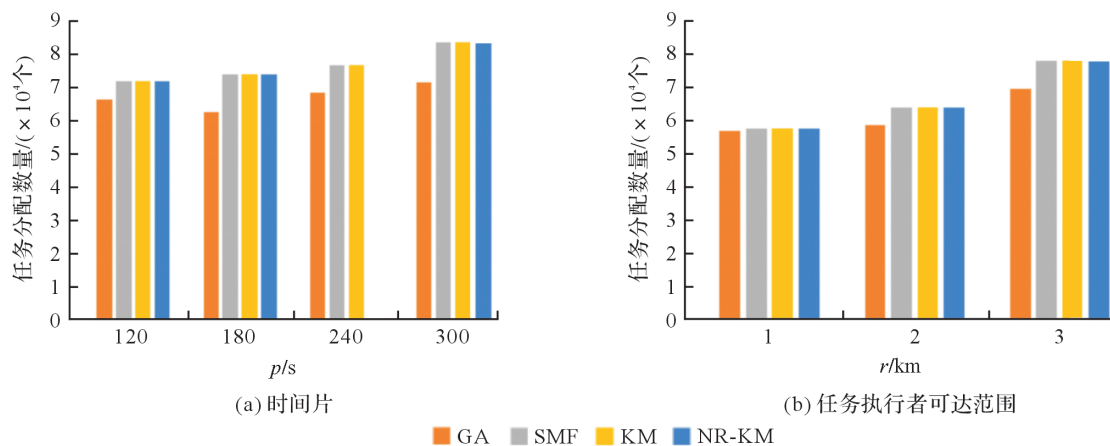


图 4 时间片和任务执行者可达范围对算法任务分配数量的影响

Fig. 4 Effect of time slices and range of executor on number of tasks assigned efficiency of NR-KM

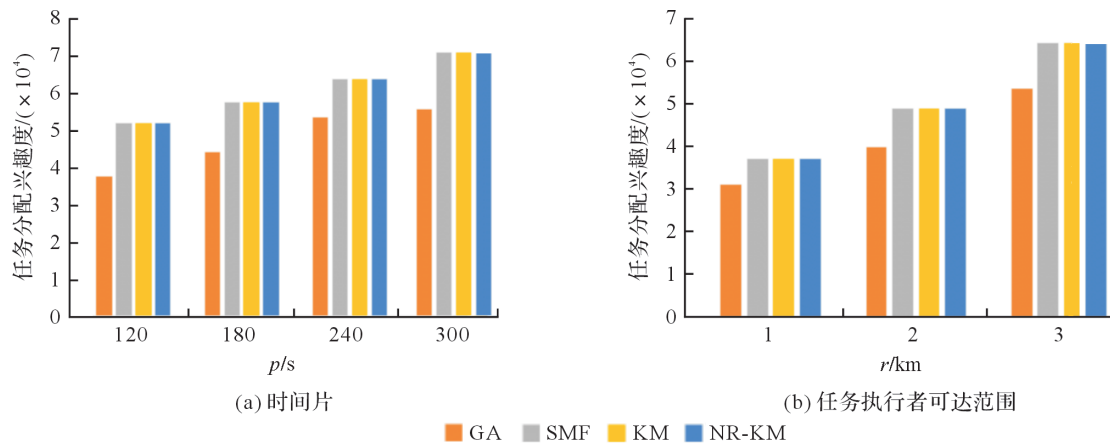


图5 时间片和任务执行者可达范围对算法任务分配兴趣度的影响

Fig. 5 Effect of time slices and range of executor on interest of task assignment of NR-KM

4 结 语

本文研究可拒绝情况下空间众包任务在线分配问题,提出动态可拒绝的空间众包处理方法,具有较强的扩展性。为减少任务执行者拒绝的概率,运用 PCA 方法计算任务执行者对任务的兴趣度;为实现高效、准确的任务分配,提出 SMF 算法和 NR-KM 算法寻找最优任务分配方案,并通过试验比较 GA、KM、SMF 和 NR-KM 4 种算法的分配性能。试验结果验证了本文算法的有效性,它是一种在任务执行者和任务动态变化情况下的人性化分配。下一步我们将致力于研究不同分配思路和应用场景的在线任务分配算法,以及不同场景下团队合作的空间众包任务分配算法,以更有效地进行任务分配。

参考文献:

- [1] LIU J X, XU K. Budget-aware online task assignment in spatial crowdsourcing[J]. World Wide Web, 2020, 23(1): 289.
- [2] TONG Y X, ZHOU Z M, ZENG Y X, et al. Spatial crowdsourcing: a survey[J]. The International Journal on Very Large Data Bases, 2020, 29(1): 217.
- [3] CHEN L, SHAHABI C. Spatial crowdsourcing: challenges and opportunities[J]. IEEE Data Engineering Bulletin, 2016, 39(4): 14.
- [4] 顾立秋. 众包中基于空间动态的任务分配技术[D]. 南京: 南京邮电大学, 2018.
- [5] GUO B, LIU Y, WANG L, et al. Task allocation in spatial crowdsourcing: current state and future directions[J]. IEEE Internet of Things Journal, 2018, 5(3): 1749.
- [6] MENG F, ZHANG S, ZHENG X, et al. Task assignment method in spatial crowdsourcing based on graph search [C]//2019 IEEE 21st International Conference on High Performance Computing and Communications. Zhangjiajie: IEEE, 2019: 2623.
- [7] TO H, SHAHABI C, KAZEMI L. A server-assigned spatial crowdsourcing framework[J]. ACM Transactions on Spatial Algorithms and Systems (TSAS), 2015, 1(1): 1.
- [8] CHENG P, LIAN X, CHEN L, et al. Prediction-based task assignment in spatial crowdsourcing [C]//2017 IEEE 33rd International Conference on Data Engineering (ICDE). San Diego: IEEE, 2017: 997.
- [9] COSTA C F, NASCIMENTO M A. In-route task selection in spatial crowdsourcing [J]. ACM Transactions on Spatial Algorithms and Systems (TSAS), 2019, 6(2): 1.
- [10] TONG Y, WANG L, ZHOU Z, et al. Dynamic pricing in spatial crowdsourcing: a matching-based approach [C]// Proceedings of the 2018 International Conference on Management of Data. Houston: Association for Computing Machinery, 2018: 773.

(下转第 247 页)