软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(3):611−630 [doi: 10.13328/j.cnki.jos.005166]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

空间众包环境下的 3 类对象在线任务分配*

宋天舒, 童咏昕, 王立斌, 许 可

(软件开发环境国家重点实验室(北京航空航天大学),北京 100191)

通讯作者: 童咏昕, E-mail: yxtong@buaa.edu.cn

摘 要: 随着移动互联网技术与 O2O(offline-to-online)商业模式的发展,各类空间众包平台变得日益流行,如滴滴出行、百度外卖等空间众包平台更与人们日常生活密不可分.在空间众包研究中,任务分配问题更是其核心问题之一,该问题旨在研究如何将实时出现的空间众包任务分配给适宜的众包工人.但大部分现有研究所基于的假设过强,存在两类不足:(1) 现有工作通常假设基于静态场景,即,全部众包任务和众包工人的时空信息在任务分配前已完整获知,但众包任务与众包工人在实际应用中动态出现,且需实时地对其进行任务分配,因此,现存研究结果在实际应用中缺乏可行性;(2) 现有研究均假设仅有两类众包参与对象,即众包任务与众包工人,而忽略了第三方众包工作地点对任务分配的影响.综上所述,为弥补上述不足,提出了一类新型动态任务分配问题,即,空间众包环境下的 3 类对象在线任务分配.该问题不但囊括了任务分配中的 3 类研究对象,即众包任务、众包工人和众包工作地点,而且关注动态环境.进而设计了随机阈值算法,给出了该算法在最差情况下的竞争比分析.采用在线学习方法进一步优化了随机阈值算法,提出自适应随机阈值算法,并证明该优化策略可逼近随机阈值算法使用不同阈值所能达到的最佳效果.最终通过在真实数据集和具有不同分布人造数据集上进行的大量实验,验证了算法的效果与性能.

关键词: 空间众包;任务分配;在线算法;竞争比分析

中图法分类号: TP311

中文引用格式: 宋天舒,童咏昕,王立斌,许可.空间众包环境下的 3 类对象在线任务分配.软件学报,2017,28(3):611−630. http://www.jos.org.cn/1000-9825/5166.htm

英文引用格式: Song TS, Tong YX, Wang LB, Xu K. Online task assignment for three types of objects under spatial crowdsourcing environment. Ruan Jian Xue Bao/Journal of Software, 2017,28(3):611−630 (in Chinese). http://www.jos.org.cn/1000-9825/5166.htm

Online Task Assignment for Three Types of Objects under Spatial Crowdsourcing Environment

SONG Tian-Shu, TONG Yong-Xin, WANG Li-Bin, XU Ke

(State Key Laboratory of Software Development Environmnt (Beihang University), Beijing 100191, China)

Abstract: With the rapid development of mobile Internet techniques and Online-to-offline (O2O) business models, various spatial crowdsourcing (SC) platforms become popular. In particular, the SC platforms, such as Didi taxi and Baidu meal-ordering service, play a significant role in people's daily life. A core issue in SC is task assignment, which is to assign real-time tasks to suitable crowd workers. Existing approaches usually are based on infeasible assumptions and have the following two drawbacks: (1) Existing methods often assume to work on the static scenarios, where the spatio-temporal information of all tasks and workers is known before the assignment is

* 基金项目: 国家重点基础研究发展计划(973)(2014CB340300); 国家自然科学基金(61502021, 71531001); 软件开发环境国家重点实验室(北京航空航天大学)开放课题(SKLSDE-2016ZX-13)

Foundation item: National Program on Key Basic Research Project of China (973) (2014CB340300); National Natural Science Foundation of China (61502021, 71531001); State Key Laboratory of Software Development Environment (Beihang University) Open Program (SKLSDE- 2016ZX-13)

收稿时间: 2016-07-31; 修改时间: 2016-09-14; 采用时间: 2016-11-01; jos 在线出版时间: 2016-11-29

CNKI 网络优先出版: 2016-11-29 13:35:14, http://www.cnki.net/kcms/detail/11.2560.TP.20161129.1335.018.html

conducted. However, since both tasks and workers dynamically appear and request to be allocated in real time, therefore, existing works are impractical in real applications. (2) Existing studies usually assume that there are only two types of objects, tasks and workers, in SC and ignore the influence of workplace for task assignment. To solve the aforementioned challenges, this paper frames a novel dynamic task assignment problem, called online task assignment for three types of objects in spatial crowdsourcing, which not only includes the three types of objects, namely tasks, workers and workplaces, but also focuses on dynamic scenarios. Moreover, a random-threshold-based algorithm is designed for the new problem and a worst-case competitive analysis is provided for the algorithm. Particularly, to further optimize the algorithm, an adaptive threshold algorithm, which is always close to the best possible effectiveness of the random-threshold-based algorithm, is developed. Finally, the effectiveness and efficiency of the proposed methods are verified through extensive experiments on real dataset and synthetic datasets generated by different distributions.

Key words: spatial crowdsourcing; task allocation; online algorithm; competitive analysis

近几年,Amazon Mechanical Turks,oDesk 等众包应用快速发展,众包吸引了工业界和学术界的广泛关注.众包通常指一种把过去由专职员工执行的工作任务通过公开的 Web 平台以自愿的形式外包给非特定的解决方案提供者群体来完成的分布式问题求解模式[1].文献[2–5]对众包的研究现状和挑战进行了较好的综述.随着智能手机和移动互联网的兴起,空间众包成为众包发展的新方向.gMission[6]是在国际上被广泛认可的空间众包平台.在国内,滴滴出行、百度外卖等空间众包平台对人们的生活产生着重要影响.在空间众包的研究中,任务分配问题是其核心问题之一.然而大部分现有任务分配问题研究的假设过强,导致其存在两点不足.

(1) 大部分研究基于静态场景(也称离线场景),即,全部众包任务和众包工人的时空信息在任务分配前已完整获知.但在动态环境(也称在线场景)下,待分配对象一个接一个地动态出现,并在等待一段时间后离开.每个对象出现时,系统需要立即做出以下两种决定之一:将该对象与还未离开的对象进行匹配(即做出任务分配,匹配、分配二词在本文通用),或令该对象继续等待.匹配决定(匹配或等待)一经给出无法更改.因此,现存基于静态场景的研究结果在实际应用中缺乏可行性;

(2) 现有研究均假设空间众包中的任务分配问题仅涉及众包工人和众包任务两类对象:众包工人移动到众包任务指定地点完成任务并获取报酬.然而目前还存在着大量其他类型的空间众包应用,其任务分配不仅涉及众包任务和众包工人,还受到第三方工作地点的影响.下文将详细介绍这类应用.

本文研究一类新型空间众包应用,在这类应用中,任务分配涉及 3 类对象:众包工人、众包任务和众包工作地点.任务分配后,众包工人和众包任务发布者均需要移动到某第三方工作地点.上述 3 类对象的任务分配问题在大量 O2O 应用中广泛存在.下文将以美容美发类 O2O 应用(如南瓜车)为例进行阐述.南瓜车是具有代表性的 O2O 理发应用,其也可被视为一个空间众包平台.客人(众包任务发布者)不仅会被分配理发师(众包工人),还将被分配理发店(众包工作地点).在分配完成后,客人和理发师都要到达被分配的理发店,然后由理发师借助理发店的设备给客人理发.这类新型空间众包应用中的任务分配问题需要将 3 种类型的对象匹配起来.本文将该问题建模成最大化三维匹配[7]问题,并研究动态场景,即,3 种类型的对象都动态出现.优化目标是最大化匹配总效用.

例 1:设在众包平台上有 4 个众包任务 t_1~t_4,3 个众包工人 w_1~w_3 和 3 个众包工作地点 p_1~p_3,他们的位置如图 1 所示.众包任务和众包工人具有活动范围,如图中圆环所示,表示他们只能被分配该范围内的众包工作地点.众包工人和众包工作地点具有容量,表示其能接受的最大任务数.在本例中,众包工人 w_1~w_3 分别有容量 1,2,1,众包工作地点 p_1~p_3 分别有容量 1,2,2.任务分配的效用定义为众包任务的报酬与众包工人服务质量的乘积.众包工人、众包任务和众包工作地点的详细信息见表 1.在动态环境下,每当一个对象出现时,众包平台可以做出两种决策:(1) 将该对象与已经出现的另外两类对象进行匹配;(2) 等待,即:暂时不进行匹配,而是等待此后出现的对象再与其进行匹配.优化目标是最大化任务分配的总效用.如果对象的出现顺序是 w_1,p_1,t_1,t_2,w_2,p_2,p_3,t_3,t_4,w_3,如图 2 所示为一种可能的匹配过程.w_1 和 p_1 先出现,由于无法构成匹配,他们进入等待状态.随后 t_1 出现,如图 2(a)所示,平台可以选择将其与 w_1,p_1 构成匹配,或令其等待.假设选择做出该匹配,则获得效用 20×0.9=18.随后 t_2 出现,如图 2(b)所示.由于没有对象可与 t_2 匹配,平台令其等待.可以注意到:如果之前没有做出匹配 $\langle w_1,p_1,t_1\rangle$,则当前可做出匹配$\langle w_1,p_1,t_2\rangle$,其效用为 100×0.9=90.但在实际应用中,对于已经做出的匹配,系统无法更

改.随后,w_2, p_2,p_3,t_3,t_4 依次出现,如图 2(c)所示,可以选择做出匹配$\langle t_3,p_2,w_2\rangle$和$\langle t_4,p_2,w_2\rangle$,也可以选择等待.假设选择使这些对象都等待.最后 w_3 出现,如图 2(d)所示,并可以构成匹配$\langle t_3,p_3,w_3\rangle$和$\langle t_4,p_3,w_3\rangle$.这两个匹配的效用分别为 48 和 72,系统选择做出这两个匹配.最终的匹配结果为 $M=\{\langle t_1,p_1,w_1\rangle,\langle t_3,p_3,w_3\rangle,\langle t_4,p_3,w_3\rangle\}$.匹配的总效用为 18+48+72=138.从以上匹配过程中可以看出:在动态环境下,需要在不知道随后出现对象信息的情况下,在当前对象出现时立即做出涉及 3 类对象的匹配决定,且不能更改.

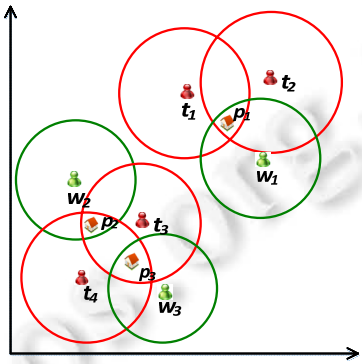


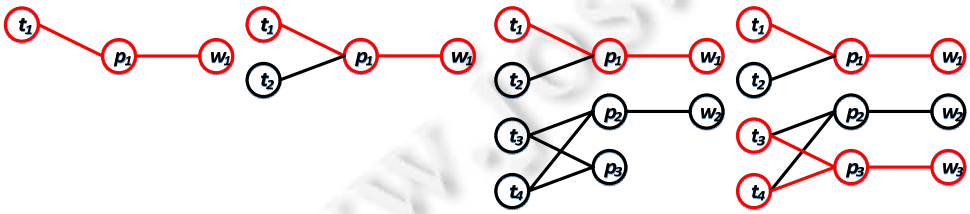
Fig.1 Locations of tasks, workers and places

图 1 任务、工人和众包工作地点位置示意图

Table 1 Information of tasks, workers and places

表 1 任务、工人和众包工作地点信息表

对象	坐标	半径	报酬	服务质量	容量
t_1	(100,140)	30	20	–	–
t_2	(140,145)	32.5	100	–	–
t_3	(80,80)	27.5	60	–	–
t_4	(55,55)	25	90	–	–
w_1	(135,110)	27.5	–	0.9	1
w_2	(50,100)	27.5	–	0.2	2
w_3	(90,50)	30	–	0.8	2
p_1	(120,125)	–	–	–	1
p_2	(60,80)	–	–	–	2
p_3	(80,60)	–	–	–	2



(a) (b) (c) (d)

Fig.2 A possible matching process

图 2 一种可能的匹配过程

本文的主要贡献有:

- 首次综合考虑第三方工作地点对任务分配的影响和任务分配的动态性,提出了空间众包环境下的 3 类对象在线任务分配问题;
- 提出随机阈值算法,并证明了该算法在最差情况下的竞争比;
- 采用在线学习方法将随机阈值算法扩展为自适应随机阈值算法,并证明该算法可逼近随机阈值算法

使用不同阈值所能达到的最佳效果;

- 在真实数据和不同分布数据集上进行实验,证明了所提出算法具有很好的近似效果与伸缩性,能够满足实时性要求的时空开销.

本文第 1 节介绍相关工作.第 2 节给出问题定义.第 3 节讨论解决方案,包括一种基于随机阈值选择的在线算法、随机阈值算法和相应的基于在线学习方法的扩展算法、自适应随机阈值算法以及相应的理论分析.第 4 节给出实验结果及分析.第 5 节对全文进行总结.

1 相关工作

本节从分配问题与空间众包两个研究方向分别回顾与本文相关的工作,具体总结如下.

1.1 分配问题

根据分配问题(assignment problem)的静态/动态特点将其各类变种研究分为离线匹配问题(又称为静态匹配问题)与在线匹配问题(又称为动态匹配问题),分别进行阐述.

1.1.1 离线匹配问题

(a) 离线二分匹配

离线二分匹配问题长期以来都是组合优化领域中的经典问题.具体而言,是指给定无向图 $G=(U\cup V, E)$,U 和 V 是互不相交的节点集合,$E\subseteq U\times V$ 是边集合,一个匹配 M 是边集 E 的子集,使得对于所有 $U\cup V$ 中的节点, M 中最多有一条边与其相连.根据二分图中边是否带有权重,又可将离线二分匹配问题分为最大二分匹配(maximum bipartite matching)和最大加权二分匹配(maximum weighted bipartite matching)[8].上述两问题均在多项式时间内可解,其中,最大二分匹配问题的优化目标是最大化匹配数,针对该问题已经提出大量经典求解算法,其中最具代表的算法如 Ford–Fulkerson 算法[8].此外,最大加权二分匹配问题的优化目标是最大化加权二分图匹配的边权加和,针对此问题也已存在众多经典算法,例如 Hungarian 算法[8].另外,Burkard 等人所撰写的专著[9]对各类离线二分匹配问题的变种进行了全面完整地介绍.

除通用的离线二分匹配问题,还有一些现存的相关研究特别关注在二维空间下两类对象的匹配问题,此类研究又被称为空间匹配.根据优化目标不同,Wong 等人[10]考虑每个对象对另一种类型对象的偏好,使得所给出的匹配是稳定匹配.余亮豪等人[11]研究在有容量约束的条件下最小化匹配的距离加和.Long 等人[12]研究最小化最大的匹配距离.注意,这些研究都基于离线情形假设.余亮豪等人[13]研究连续空间匹配,在初始匹配给出后,在保持匹配稳定性的基础上维护匹配的最优性.Gao 等人[14]研究将资源分配给动态出现的空间事件,即事件在线到达,优化目标是最小化匹配距离的加和.这两项工作所研究问题的优化目标与本文不同.

(b) 离线三维匹配

不同于离线二分匹配问题是多项式时间内可解的,Garey 等人[7]证明了最大三维匹配问题(maximum 3-dimensional matching)是 NP 难的.进一步地,Kann 等人[15]证明了最大三维匹配问题是 MAX SNP-hard[16].在该问题的近似算法研究方面,Hurkens 等人[17]证明了最大三维匹配问题对任意 $\varepsilon>0$ 是 2/3-ε可近似的,Arkin 等人[18]证明了最大加权三维匹配问题对任意 $\varepsilon>0$ 是 1/2-ε可近似的.最大三维匹配问题和最大加权三维匹配问题的近似算法基于一种称为局部搜索(local search)的思路:尝试从现有匹配中删除一些边,然后加入与剩余匹配不冲突(即没有公共节点)的边,并判断是否能够获得更好的匹配结果.下面通过一个例子介绍这一思想.

例 2:可以由表 1 建立如图 3(a)所示的三分图,并定义每条边(连接 3 个点)的权值为表 1 所示的任务报酬与工人服务质量的乘积.例如,边(t_1,p_1,w_1)的权值为 20×0.9=18,边(t_2,p_1,w_1)的权值为 100×0.9=90.对于容量大于 1 的对象,采用复制的方式.例如,p_2 的容量为 2,将其复制为 p_{21} 和 p_{22}.经过复制操作后,得到图 3(b).算法首先依次将相互不冲突的边加入临时结果集.在图 3(b)中,按照自上而下顺序,算法先将边(t_1,p_1,w_1)加入临时结果集.在考虑边(t_2,p_1,w_1)时,由于该边与(t_1,p_1,w_1)冲突,因此不将其加入临时结果集.随后,算法依次将边(t_3,p_{21},w_{21})和(t_3,p_{22},w_{22})加入临时结果集,此时,临时结果集如图 3(b)所示绿色部分.然后,算法进行局部搜索:对临时结果集中的每一条边,尝试删除该边,并引入与该边冲突但不与临时结果集中其他边冲突且权值大于被删边的边.算法首先尝试

删除(t_1,p_1,w_1),并发现可以将(t_2,p_1,w_1)加入临时结果集.由于(t_2,p_1,w_1)的权值为 90,大于(t_1,p_1,w_1)的权值 18,算法执行此次替换;类似地,算法继续删除(t_3,p_{21},w_{21}),引入(t_3,p_{31},w_{31});最后删除(t_4,p_{22},w_{22}),引入(t_4,p_{32},w_{32}).经过局部搜索后,结果集中的边如图 3(c)中所示红色部分.最后将其转化为原始图,如图 3(d)所示.局部搜索需要更改已经做出的匹配决定,因此并不适用于动态场景.

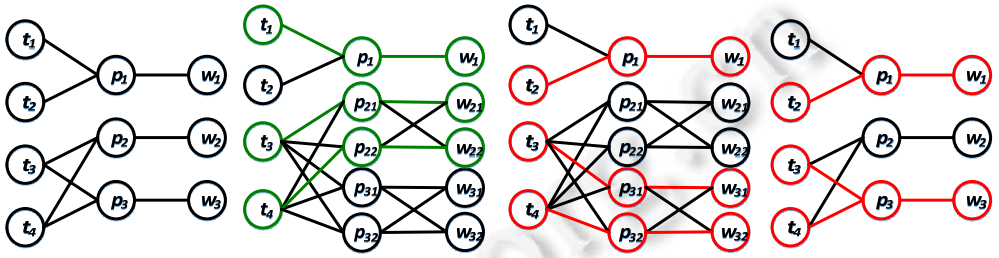


Fig.3 An example of maximum weighted 3-dimensional matching

图 3 加权最大化三维匹配问题示例图

综上所述,上述离线匹配问题要求匹配执行前必须获得输入对象的全部信息,而不可根据部分二分图或三维图信息进行匹配.因此,上述方法都不适用于文本所讨论的动态在线匹配问题.

1.1.2 在线匹配问题

Hassan 等人[19]研究在线的空间任务分配问题,其优化目标是最大化任务分配数.文献[19]和本文工作有以下区别.

- 首先,本文研究的任务分配问题涉及 3 类对象:众包任务、众包工人和众包工作地点,并且这 3 类对象都动态出现;而文献[19]研究匹配众包任务和众包工人这两类对象,并且仅允许一类对象动态出现;
- 其次,本文研究最大化任务分配总收益,而文献[19]研究最大化任务分配的数量;
- 此外,文献[19]的解决方案无法给出竞争比保证,而本文的解决方案给出了竞争比分析.

Ting 等人[20]研究单边在线(一类对象动态出现)最大加权二分匹配问题,并给出了该问题目前最好的竞争比.Tong 等人[21,22]研究双边在线(两类对象动态出现)任务分配问题,分别研究了最大化任务分配总收益和在最大化任务分配数的前提下最小化任务分配总花费.文献[20–22]与本文的区别是,本文研究涉及 3 类动态出现对象任务分配.

1.2 空间众包

本节介绍空间众包领域的 3 个主要研究问题,包括任务分配、任务规划和隐私保护.任务分配问题一直是空间众包研究中的核心问题之一,其研究角度为全局优化;任务规划问题则从众包工人的视角出发,最大化工人的收益,是一种局部优化;隐私保护主要研究在任务分配的过程中保护众包工人和众包任务发布者的位置隐私.

1.2.1 任务分配

Kazemi 等人[23]最早提出了空间众包平台上的任务分配问题,研究最大化任务分配数.Kazemi 等人[24]在文献[23]模型的基础上增加了任务完成的正确率约束.To 等人[25]扩展了文献[23]提出的模型,赋予匹配分数,优化目标是最大化匹配的总分数.Cheng 等人[26]研究在任务分配的同时最大化工人的可靠性和空间分布的差异性.She 等人[27,28]研究了在时空冲突约束下的任务分配问题,并分别研究了离线分配与在线分配两种场景.此外,对于复杂类型的空间众包任务,通常非单一众包工人所能解决,而需要一个众包团队协同完成.针对此类问题,Gao 等人[29]研究了 top-k 最佳众包团队推荐问题.上述工作或研究离线情形,或研究仅涉及两类对象的分配,因此,其解决方案均不适用于本文研究的问题.

1.2.2 任务规划

Deng 等人[30]从单一众包工人的视角研究静态路径规划问题,该问题旨在给定此众包工人出行预算成本的

条件,为其规划完成任务的路径,从而最大化其所完成任务的数量.此外,She 等人[31]研究了针对空间众包平台中全部众包工人的静态全局路径规划问题.上述两类研究只针对静态离线场景,Li 等人[32]进而研究了针对单一众包工人的动态在线路径规划问题,该问题在保证众包工人按时到达目的地的前提下,最大化工人完成任务所获得的收益.虽然上述工作也针对动态在线场景,但其仅关注单一众包工人,因此并不适用于本文研究的问题.

1.2.3 隐私保护

To 等人[33]基于差分隐私(differential privacy)给出了一种在保护众包工人隐私的同时提供有效空间众包服务的机制.Pournajaf 等人[34]研究了基于隐蔽位置(cloaked locations)进行任务分配.上述工作研究的问题与本文不同,且关注静态场景,其解决方案不适用于本文研究的问题.

2 问题描述

本节给出形式化问题描述.首先给出问题定义,然后介绍在线算法的评价模型,最后给出离线场景下问题的复杂性分析.

2.1 问题定义

首先介绍 3 个基本概念,然后给出匹配效用的定义.

定义 1(众包任务). 一个众包任务定义为 $t=\langle l_t,r_t,u_t,b_t,e_t\rangle$,表示任务的位置为欧式空间中的一个点 l_t,同意接受的移动范围是以 l_t 为圆心、以 r_t 为半径的区域,任务的回报为 u_t,任务的出现时间为 b_t,截止时间为 e_t.

定义 2(众包工人). 一个众包工人定义为 $w=\langle l_w,r_w,c_w.q_w,b_w,e_w\rangle$,表示工人的位置为欧式空间中的一个点 l_w,同意接受的移动范围是以 l_w 为圆心、以 r_w 为半径的区域,同意接受的最大请求数(即容量)为正整数 c_w,工人的服务质量 $q_w\in(0,1]$,出现时间为 b_w,截止时间为 e_w.

定义 3(众包工作地点). 一个众包工作地点定义为 $p=\langle l_p,c_p,b_p,e_p\rangle$,表示众包工作地点的位置为欧式空间中的一个点 l_p,并且能够容纳的最大任务数(即容量)为正整数 c_p,出现时间为 b_p,截止时间为 e_p.

参照文献[21],定义众包工人完成众包任务的效用:

定义 4(效用). 众包工人 w 完成众包任务 t 的效用定义为 $U(t,w)=u_t\times q_w$,即任务回报与工人服务质量的乘积.

最后,定义空间众包环境下的 3 类对象在线任务分配问题.

定义 5(空间众包环境下的 3 类对象在线任务分配问题). 给定众包任务集合 T、众包工人集合 W、众包工作地点集合 P 和一个效用函数 $U(\cdot,\cdot)$,众包任务、众包工人和众包工作地点按照某种顺序一个接一个地出现.求解目标是找到 T,W,P 的任务分配 $M\subseteq T\times W\times P$,最大化任务分配总效用 $\mathrm{Max}Sun(M)=\sum_{t\in T,w\in W}U(t,w)$ 且满足以下约束.

(1) 截止时间约束:当一个众包任务 t 或一个众包工人 w 或一个众包工作地点 p 出现时,可以给出任务分配,并要求众包工人活动时间$[b_w,e_w]$、众包任务活动时间$[b_t,e_t]$和众包工作地点活动时间$[b_p,e_p]$间两两相交;

(2) 不变性约束:任务分配一旦给出,则不能改变;

(3) 容量约束:每个任务 t 在 M 中最多出现 1 次,每个工人 w 在 M 中最多出现 c_w 次,每个众包工作地点 p 在 M 中最多出现 c_p 次;

(4) 空间约束:任务分配$\langle t,p,w\rangle$需要满足 p 在以 l_t 为中心、以 r_t 为半径的范围内和以 l_w 为中心、以 r_w 为半径的范围内.

2.2 评价模型

在线算法研究中,通常使用竞争比评价算法性能.由于本文提出的算法都是随机在线算法,下面介绍随机在线算法的竞争比:

定义 6(竞争比). 设算法 A 是空间众包环境下的 3 类对象在线任务分配问题的随机在线算法.对任意的输入 $T,W,P,U(\cdot,\cdot)$,设 $E[\mathrm{Max}Sum(A)]$是算法 A 返回匹配总效用的期望,$\mathrm{Max}Sum(OPT)$是可能获得的最大匹配总效用.

如果对于任意的输入都满足 $E[\mathrm{Max}Sum(A)]\geqslant c\mathrm{Max}Sum(OPT)$,则算法 A 的竞争比为 c.

2.3 问题复杂性

本节分析所研究空间众包环境下的3类对象在线任务分配问题离线版本的复杂性.在离线版本中,3类对象的所有信息在任务分配前已知.将离线版本称为空间众包环境下的 3 类对象任务分配问题.由于三维匹配问题可以规约到该问题,可证明该问题是 NP 难问题.

定理 1. 空间众包环境下的 3 类对象任务分配问题是 NP 难问题.

证明:考虑空间众包环境下的 3 类对象任务分配问题的一种特殊情况:众包任务数、众包工人数和众包工作地点数相等,任务回报为 1,工人服务质量为 1,工人和众包工作地点容量为 1.此时,该问题等价于三维匹配问题的优化问题,而三维匹配问题的决策问题已经被证明是NP完全问题[7].因此,空间众包环境下的3类对象任务分配问题是 NP 难问题. □

综上所述,由于空间众包环境下的 3 类对象在线任务分配问题的离线版本已经是一个 NP 难问题,因此,解决空间众包环境下的 3 类对象在线任务分配问题更加困难.

3 解决方案

首先给出一种解决该问题的基本算法,称为朴素随机算法,并对算法进行了简单讨论;随后给出一种基于随机选择阈值的在线算法,称为随机阈值算法,并给出该算法的竞争比分析;最后,结合在线学习方法扩展随机阈值算法,给出自适应随机阈值算法,并证明该算法效果接近随机阈值算法使用不同阈值所能达到的最佳效果.

3.1 基本算法

3.1.1 算法思想

朴素随机算法思想为随机地进行任务分配.具体地,每当有新对象出现,算法从能通过这一对象做出的任务分配中随机选择一个,加入匹配结果集.

3.1.2 算法伪代码

算法伪代码见表 2.

Table 2 Random algorithm

表 2 朴素随机算法

算法. Random algorithm.
Input: $T,W,P,U(\cdot,\cdot)$;
Output: M.
1: Foreach 新出现的众包任务或众包工人或众包工作地点 v
2: $Cand\leftarrow${包含 v 且满足所有约束条件的任务分配}
3: If $Cand$ 不是空集
4: 从 $Cand$ 中任意选择一个,加入到 M 中
5: Return M

算法的输入为一个接一个出现的众包任务、众包工人和众包工作地点以及一个效用函数,输出为任务分配结果集 M.在算法的执行过程中,首先,在算法的第 1 行、第 2 行,每当有新对象 v 出现,算法将能通过这一对象做出的任务分配加入集合 $Cand$.在算法的第 3 行、第 4 行,如果 $Cand$ 不是空集,则从集合 $Cand$ 中随机选择一个元素,加入结果集 M.算法的第 5 行返回结果集 M.注意:当容量为 c_w 的众包工人 w 出现,算法将其看做 c_w 个容量为 1 众包工人,并逐一处理.同样地,当容量为 c_p 的众包工作地点 p 出现时,算法将其看作 c_p 个容量为 1 的众包工作地点,并逐一处理.

复杂性分析:对每个新出现的众包任务 $t\in T$、众包工人 $w\in W$ 和众包工作地点 $p\in P$,朴素随机算法的时间和空间复杂度分别是 $O(|P||W|)$,$O(|T||P|)$和 $O(|T||W|)$.

3.1.3 运行样例

例 3:设集合 T,W,P 中对象信息见表 1.出于简单起见,在例子中忽略截止时间约束.设对象出现顺序为 w_1,p_1,

t_1,t_2,w_2,p_2,p_3,t_3,t_4,w_3,朴素随机算法运行过程如图 4 所示.如图 4(a)所示,最先出现的 3 个对象是 w_1,p_1,t_1,算法将 t_1 分配给 w_1,p_1,并获得效用 $U(t_1,w_1)$=18.在做出该分配后,w_1,p_1 的容量减为 0.如图 4(b),随后出现的对象是 t_2.由于无法通过 t_2 做出任务分配,算法使 t_2 等待.如图 4(c)所示,随后出现的 5 个对象是 w_2,p_2,p_3,t_3,t_4,算法做出任务分配 $\langle t_3,p_2,w_2\rangle$和$\langle t_4,p_2,w_2\rangle$,并获得效用 $U(t_3,w_2)$=12 和 $U(t_4,w_2)$=18.最后出现的对象是 w_4,如图 4(d)所示.通过 w_4 无法做出任务分配,算法结束.最终的任务分配结果为 M={$\langle t_1,p_1,w_1\rangle$,$\langle t_3,p_2,w_2\rangle$,$\langle t_4,p_2,w_2\rangle$},总效用为 18+12+18=48.

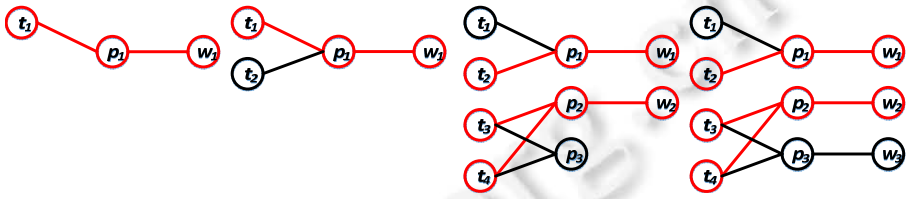


Fig.4 An example of random algorithm

图 4 朴素随机算法运行示意图

从上述例子可以看出,朴素随机算法给出任务分配结果总效用较低.原因在于:当效用较低任务分配与效用较高任务分配冲突,且效用较低任务分配的相关对象先出现时,算法将损失效用较高的任务分配.

3.2 随机阈值算法

3.2.1 算法思想

在朴素随机算法中,当两个任务分配相互冲突时,先做出的效用较小的分配,可能导致损失随后的效用较大任务分配.针对这一问题,本文扩展 Greedy-RT 算法[20],设计了一种基于随机阈值选择的在线算法,称为随机阈值算法.Greedy-RT 算法仅适用于两类对象匹配,且仅允许一类对象动态出现.本文将其扩展为适用于匹配 3 类动态出现的对象.随机阈值算法的主要思想是:首先选择一个效用阈值,在任务分配过程中,仅做出效用大于该阈值的分配.这种策略虽然可能会损失一些效用较小的分配,但在相互冲突的分配中能够选取较大者,从而规避最差情况.在随机阈值算法中,对于每个出现的众包工人 w,将其看做 c_w 个同时出现且容量为 1 的众包工人.类似地,对于每个出现的众包工作地点 p,将其看做 c_p 个同时出现且容量为 1 的众包工作地点.

3.2.2 算法伪代码

算法执行过程见表 3.

算法的输入为一个接一个出现的众包任务、众包工人和众包工作地点以及一个效用函数,输出为任务分配结果集 M.在算法的执行过程中,在第 1 行、第 2 行,随机阈值算法根据任务分配全过程中可能获得的最大效用 $U_{\max}$,随机选择一个阈值(e^k),$U_{\max}$ 可以通过历史记录学习得到.在算法的第 3 行~第 6 行,当一个新对象 v 出现时,算法从所有能通过 v 做出且效用不小于所选阈值的任务分配中任意选择一个,加入结果集 M.算法的第 7 行返回结果集 M.

复杂性分析:对每个新出现的众包任务 $t\in T$、众包工人 $w\in W$ 和众包工作地点 $p\in P$,朴素随机算法的时间和空间复杂度分别是 $O(|P||W|)$,$O(|T||P|)$和 $O(|T||W|)$.

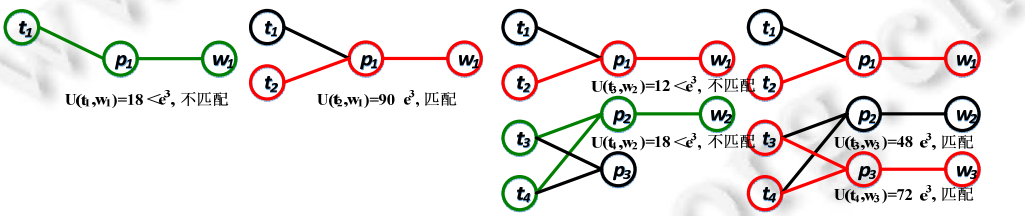
Table 3 Random-Threshold-Based algorithm

表 3 随机阈值算法

算法. Random-Threshold-Based algorithm.
Input: $T,W,P,U(\cdot,\cdot)$;
Output: M.
1: $\theta \leftarrow \lceil \ln(U_{\max}+1) \rceil$
2: 以概率 $p_0,p_1,\ldots,p_{\theta-1}$ 从 0,1,…,θ–1 中随机取值作为阈值,其中,$p_i>0$, $\sum_i p_i=1$
3: Foreach 新出现的众包任务或众包工人或众包工作地点 v
4: $Cand \leftarrow${包含 v、满足所有约束条件且效用不小于 e^k 的任务分配}
5: If $Cand$ 不是空集 Then
6: 从 $Cand$ 中任意选取一个任务分配,加入到 M 中
7: Return M

3.2.3 运行样例

例 4:设集合 T,W,P 中元素信息见表 1.出于简单起见,在例子中忽略截止时间约束.由于 $U_{\max}=100$, $\theta=\lceil \ln(U_{\max}+1) \rceil=5$,$k$ 将从 0,1,2,3,4 中随机取值.假设取 $k=4$,则阈值为 $e^4\approx 20.1$.设对象出现顺序 $w_1,p_1,t_1,t_2,w_2,p_2,p_3,t_3,t_4,w_3$.随机阈值算法运行过程如图 5 所示.在图 5(a)中,最先出现的 3 个对象是 w_1,p_1,t_1,可以通过他们做出任务分配.但由于该分配效用 $U(t_1,w_1)=18<e^3$,算法不做出该分配.随后出现对象是 t_2,如图 5(b)所示.通过 t_2 可以做出分配$\langle t_2,p_1,w_1\rangle$,该分配效用 $U(t_2,w_1)=90\geq e^3$,算法做出该分配.随后出现的 5 个对象是 w_2,p_2,p_3,t_3,t_4,如图 5(c)所示,可做出分配$\langle t_3,p_2,w_2\rangle$和$\langle t_4,p_2,w_2\rangle$.由于 $U(t_3,w_2)=12<e^3$,$U(t_4,w_2)=18<e^3$,算法不做出任务分配.最后出现的对象是 w_3,如图 5(d)所示.通过 w_3 可做出分配$\langle t_3,p_3,w_3\rangle$和$\langle t_4,p_3,w_3\rangle$.由于 $U(t_3,w_3)=48\geq e^3$,$U(t_4,w_3)=72\geq e^3$,算法将这两个分配加入结果集 M.最终分配为 $M=\{\langle t_2,p_1,w_1\rangle,\langle t_3,p_3,w_3\rangle,\langle t_4,p_3,w_3\rangle\}$,总效用为 90+48+72=210.由于该算法是随机算法,其期望总效用为(48+48+48+210+162)/5=103.2.



(a) (b) (c) (d)

Fig.5 An example of random-threshold-based algorithm

图 5 随机阈值算法运行过程

与朴素随机算法给出结果相比,可以发现随机阈值算法给出了效用更高的分配结果.原因在于,随机阈值算法能从冲突的任务分配中选择效用较高者.但该算法存在的问题是:选择不同阈值时,算法获得的总效用差异较大,算法表现不稳定.

3.2.4 理论分析

定理 2. 随机阈值算法的竞争比是 $p_{\min}/3e$,其中,$p_{\min}=\min\{p_0,p_1,\ldots,p_{\theta-1}\}$.

证明:设在离线场景下算法的输入对应的三分图是 G.设 $G_{[e^i,e^{i+1})}$ 是 G 的子图,该子图中所有连接 3 类点的边的效用都在区间$[e^i,e^{i+1})$内.设 $OPT_{[e^i,e^{i+1})}$ 和 $M_{[e^i,e^{i+1})}$ 分别为在图 $G_{[e^i,e^{i+1})}$ 上的最优匹配结果和随机阈值算法的匹配结果.对 $OPT_{[e^i,e^{i+1})}$ 中任意一条连接 3 类点的边,其所连接的 3 个点至少有一个在 $M_{[e^i,e^{i+1})}$ 被某条边连接.因此,$|M_{[e^i,e^{i+1})}|\geqslant|OPT_{[e^i,e^{i+1})}|/3$.因此,

$$\begin{aligned}E[\text{Max}Sum(M)] &= \sum_{i=0}^{\theta-1} p_i \text{Max}Sum(M_{[e^i,\infty)})\\ &\geqslant p_{\min}\sum_{i=0}^{\theta-1}\text{Max}Sum(M_{[e^i,e^{i+1})})\\ &\geqslant p_{\min}\sum_{i=0}^{\theta-1} e^i \mid M_{[e^i,e^{i+1})} \mid\\ &\geqslant p_{\min}\sum_{i=0}^{\theta-1} e^i \mid OPT_{[e^i,e^{i+1})} \mid /3\\ &\geqslant \frac{p_{\min}}{3e}\sum_{i=0}^{\theta-1}\text{Max}Sum(OPT_{[e^i,e^{i+1})})\\ &\geqslant \frac{p_{\min}}{3e}\text{Max}Sum(OPT).\end{aligned}$$

因此,随机阈值算法的竞争比为

$$CR = \frac{p_{\min}}{3e}.$$ □

3.3 自适应随机阈值算法

随机阈值算法性能受 k 值影响较大.为对其进行优化,本节介绍结合 Randomized Weighted Majority 算法[35,36]的自适应随机阈值算法.Blum 等人[37]使用 Randomized Weighted Majority 算法解决了一种具有内部状态的专家问题.本文使用文献[37]的证明框架,证明了自适应算法效果接近随机阈值算法的最佳效果.

3.3.1 基本思想

自适应随机阈值算法本质上采用了一种在线学习的策略.在随机阈值算法中,使用不同阈值将导致不同的匹配结果,具有不同的总效用.自适应随机阈值算法在算法运行过程中动态调整选择不同 k 值的概率 p_k,增加效果较好 k 值被选择的概率,以提高算法的性能.

3.3.2 算法伪代码

自适应随机阈值算法伪代码见表 4.

Table 4 Adaptive random-threshold-based algorithm

表 4 自适应随机阈值算法

算法. Adaptive random-threshold-based algorithm.

Input: $T,W,P,U(\cdot,\cdot)$;
Output: M.
1: 对每个 i=0,1,…,θ–1,设 w_i=1, $W=\sum_i w_i$,p_i=w_i/W
2: Foreach 新出现的众包任务、众包工人或众包工作地点 v
3: 依概率 p_i 从 0,1,…,θ–1 随机选择一个作为 k 值
4: *Cand*←{包含 v、满足所有约束条件且效用不小于 e^k 的任务分配}
5: If *Cand* 不是空集 Then
6: 从 *Cand* 中任意选取一个任务分配,加入 M
7: $w_k = w_k(1+\delta)^{u_k/U_{\max}}$,其中,u_k 是随机阈值算法一直使用 e^k 为阈值时获得的效用
8: 更新 W 和 p_i
9: Return M

算法的输入为一个接一个出现的众包任务、众包工人和众包工作地点以及一个效用函数,输出为任务分配结果集 M.在算法的执行过程中,第 1 行,设 k 的每个可能取 0,1,…,θ–1 权重为 w_i=1,并设 $W=\sum_i w_i$,p_i=w_i/W,即:在还没有任何对象出现前,k 将从 0,1,…,θ–1 中均匀随机选取.在算法的第 2 行、第 3 行,每当有新对象 v 出现,依照 p_k 随机选取 k 值.当使用新选取 k 值完成任务分配后,算法使用公式 $w_k = w_k(1+\delta)^{u_k/U_{\max}}$ 更新 w_k,并相应地更新 W 和 p_k.符号 u_k 表示如果一直使用 e^k 作为阈值,随机阈值算法通过对象 v 获得的效用.

复杂性分析:对每个新出现的众包任务 $t\in T$、众包工人 $w\in W$ 和众包工作地点 $p\in P$,朴素随机算法的时间和空间复杂度分别是 $O(\lceil\ln(U_{\max}+1)\rceil|P||W|)$,$O(\lceil\ln(U_{\max}+1)\rceil|T||P|)$和 $O(\lceil\ln(U_{\max}+1)\rceil|T||W|)$.

3.3.3 理论分析

定理 3. 设$\delta=\varepsilon U_{\max}/2D$.$D$ 是变化 k 值导致的损失上界(变化 k 值可能导致一直使用某一 k 值时的分配由于冲突无法给出),且满足 $D\geqslant U_{\max}$,则自适应随机阈值算法的总效用至少是

$$OPT(1-\varepsilon)-\frac{2D}{\varepsilon}\ln(\theta),$$

其中,OPT 是选取不同的 k 值时随机阈值算法给出任务分配总效用的最大值.

证明:设算法当前的权重为 $w_0,w_1,\ldots,w_{\theta-1}$,使用不同 k 值的效用为 $u_0,u_1,\ldots,u_{\theta-1}$.更新后的权重为

$$w'_0,w'_1,\ldots,w'_{\theta-1},$$

其中,$w_k=w_k(1+\delta)^{u_k/U_{\max}}$.设 $W=\sum_i w_i, W'=\sum_i w'_i, p_i=w_i/W, p'_i=w'_i/W'$.

当某个对象 v 出现时,算法通过分配 v 获得效用的期望为 $\sum_i p_i u_i$,将其称为 E_v.从概率分布 p_i 转移到概率分布 p'_i 的损失最多为

$$D\sum_{i:p'_i>p_i}(p'_i-p_i)\leqslant\frac{D}{W}\sum_{i:p'_i>p_i}(w'_i-w_i)\leqslant\frac{D}{W}(W'-W)=D\left(\frac{W'}{W}-1\right).$$

由于对任意 $x\leqslant 1$,有$(1+\alpha)^x\leqslant 1+x\alpha$,则有:

$$W'\leqslant\sum_i w_i\left(1+\frac{\delta u_i}{U_{\max}}\right)=W\sum_i p_i\left(1+\frac{\delta u_i}{U_{\max}}\right)=W\left(1+\frac{\delta}{U_{\max}}E_v\right).$$

因此,

$$D\left(\frac{W'}{W}-1\right)\leqslant D\frac{\delta}{U_{\max}}E_v.$$

因此,算法从对象 v 获得的效用至少为

$$E_v\left(1-\frac{D\delta}{U_{\max}}\right).$$

设 W_f 是算法结束时的总权重,则有:

$$(1+\delta)^{OPT/U_{\max}}\leqslant W_f\leqslant\theta\prod_v\left(1+\frac{\delta}{U_{\max}}E_v\right).$$

对任意的 $x\in[0,1]$,$\ln(1+x)\in[x-x^2/2,x]$,则有:

$$\frac{OPT}{U_{\max}}(\delta-\delta^2/2)\leqslant\frac{OPT}{U_{\max}}\ln(1+\delta)\leqslant\ln\theta+\sum_v\ln\left(1+\frac{\delta}{U_{\max}}E_v\right)\leqslant\ln\theta+\frac{\delta}{U_{\max}}\sum_v E_v.$$

因此,

$$\sum_v E_v\geqslant OPT\left(1-\frac{\delta}{2}\right)-\frac{U_{\max}}{\delta}\ln\theta.$$

因此,算法的总效用为

$$G\geqslant OPT\left(1-\frac{\delta}{2}\right)\left(1-\frac{\varepsilon}{2}\right)-\left(1-\frac{\varepsilon}{2}\right)\frac{2D}{\varepsilon}\ln\theta\geqslant OPT(1-\varepsilon)-\frac{2D}{\varepsilon}\ln\theta.$$ □

4 实验分析

4.1 实验设置

实验环境:处理器为 2.5GHz Inter(R) Core(TM) i7-4710MQ,内存为 8GB,操作系统为 Windows 10,编程语言为 C++.

通过在真实数据和具有不同分布数据上进行实验,验证算法的运行效果.实验数据的参数设置见表 5,加黑的值为具有多种取值参数的默认值.

Table 5 Statistics of experimental data

表 5 实验数据参数表

参数	取值
$n=\|T\|=\|W\|=10\|P\|$	1k,2k,**3k**,4k,5k,10k
众包任务报酬的均值(正态分布)	10,30,**50**,70,90
众包任务报酬的标准差(正态分布)	5,15,**25**,35,45
众包任务报酬的 Shape(幂律分布)	1,3,**5**,7,9
众包任务和众包工人的范围	5,7,**10**,13,15
众包众包工作地点的容量	5,6,**7**,8,9
众包工人的服务质量均值(正态分布)	0.5,0.6,**0.7**,0.8,0.9
众包工人的服务质量标准差(正态分布)	0.1
$U_{\max}$	100
对象的坐标	100×100 的网格内均匀分布
对象的出现时间	在[0,480]均匀分布
众包任务、众包工人和众包工作地点的等待时间	10

自适应随机阈值算法显然优于随机阈值算法,因此在实验中比较朴素随机算法(实验图中为 Random)和自适应随机阈值算法(实验图中为 Adaptive RT).人造数据依据表 5 生成,最大效用 $U_{\max}$ 设置为 100.在实际应用中,最大效用可以通过历史记录估计得到.在自适应随机阈值算法中,设δ=0.01.真实数据从空间众包平台 gMission 上收集得到.

4.2 实验结果

4.2.1 改变分布

在这组实验中,改变任务报酬分布情况,其他参数使用表 5 中的默认值.当任务的报酬服从正态分布时,改变均值的实验结果如图 6(a)所示.

可以看出:

- 在效用方面,自适应随机阈值算法始终优于朴素随机算法.随着均值的增加,两种算法返回结果的效用都增加,这是由于整体上任务的回报在增加.另一方面,随着均值的增加,两种算法总效用的差距在变大.可能的原因是:当均值增加,越来越多任务的回报集中在一个较大值附近,而自适应随机阈值算法在运行的过程中会过滤掉效用较小的边,从而增加了选择效用较大边的机会;
- 在时间和空间消耗方面,自适应随机阈值算法的运行时间要长于朴素随机算法的运行时间,内存消耗要略高于朴素随机算法.原因是,自适应随机阈值算法需要维护每种单一阈值的运行历史.注意:虽然自适应随机阈值算法运行时间较长,但处理每个对象所消耗的时间是毫秒级,因此可以满足实时性的要求.

当任务的报酬服从正态分布时,改变标准差的实验结果如图 6(b)所示.

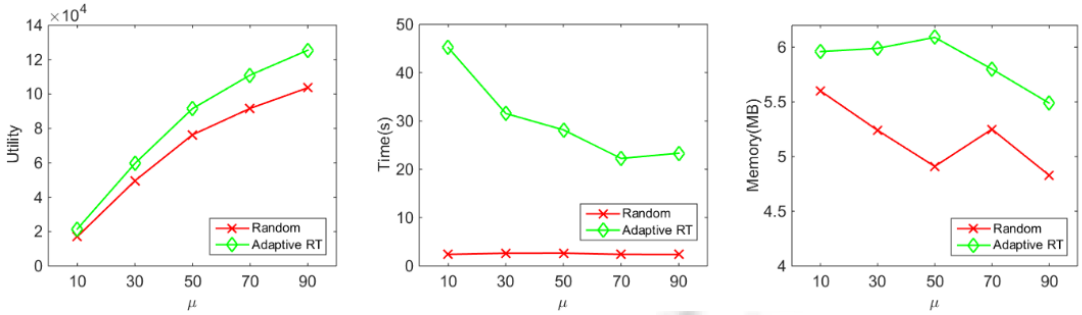
可以看出:

- 在效用方面,自适应随机阈值算法始终优于朴素随机算法.随着标准差的增加,两种算法返回结果的效用有所波动但变化不大,这是由于任务报酬的均值不变;
- 在时间和空间消耗方面,自适应随机阈值算法的运行时间要长于朴素随机算法的运行时间,内存消耗要略高于朴素随机算法.

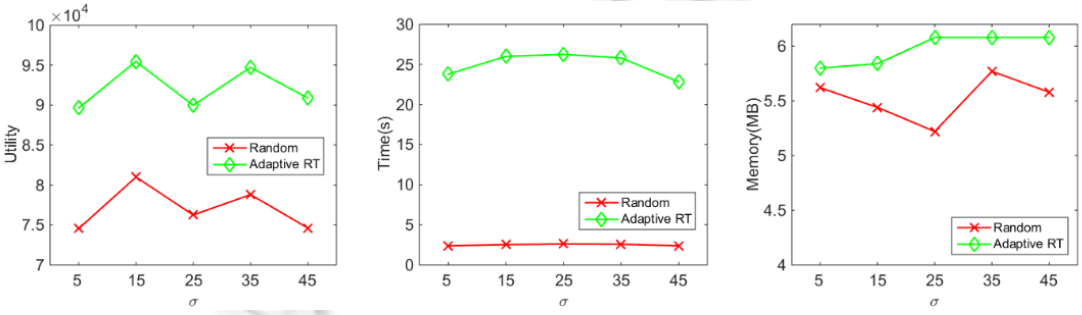
当任务的报酬服从幂率分布时,改变幂率分布 Shape 参数的实验结果如图 6(c)所示.注意:本文调整了幂率分布的集中方向,使得当参数变大时,分布向较大值集中.

可以看出:

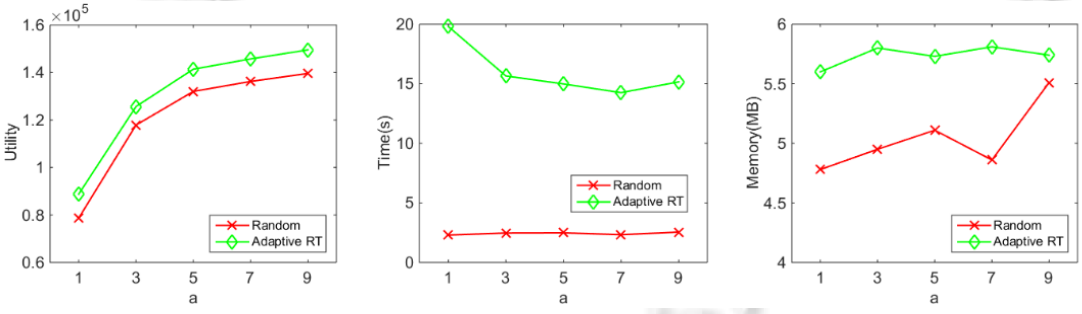
- 在效用方面,自适应随机阈值算法始终优于朴素随机算法.随着 Shape 参数的增加,两种算法返回结果的效用都在增大,这是由于任务的报酬越来越向 $U_{\max}$ 集中;
- 在时间和空间消耗方面,自适应随机阈值算法的运行时间要长于朴素随机算法的运行时间,内存消耗要略高于朴素随机算法.



(a) 改变任务报酬的均值(正态分布)



(b) 改变任务报酬的标准差(正态分布)



(c) 改变任务报酬的 Shape(幂率分布)

Fig.6 Varying the distribution of tasks' reward

图 6 改变任务报酬的分布

4.2.2 改变数量

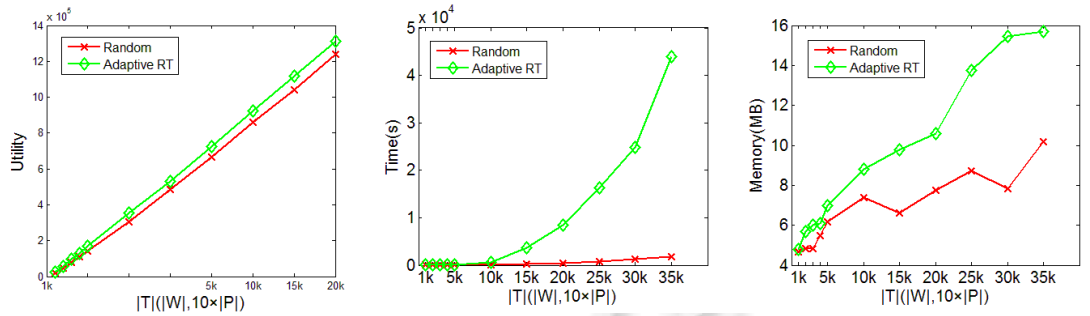
在这组实验中,改变任务分配对象的数量,其他参数使用表 5 中的默认值.改变对象数量的实验结果如图 7 所示.注意:为了验证所提出算法的可扩展性,在图 7(a)中增加了|T|分别取 15k,20k,25k,30k 和 35k 的实验结果.

可以看出:

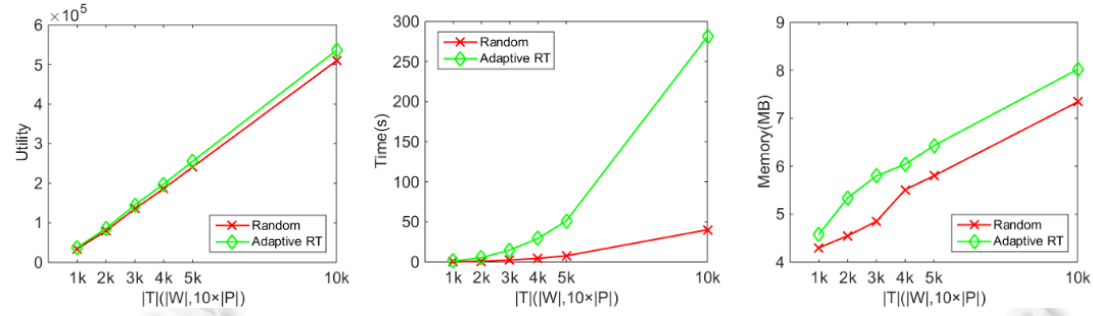
- 在效用方面,自适应随机阈值算法始终优于朴素随机算法.随着对象数量的增加,自适应随机阈值算法和朴素随机算法返回结果的效用都在增大,这是由于可达成的任务分配数增加;
- 在时间和空间消耗方面,自适应随机阈值算法的运行时间要长于朴素随机算法的运行时间,内存消耗要略高于朴素随机算法.

值得注意的是:在图 7(a)中,随着对象数量的增加,时间消耗的增加速度快于任务分配对象数量的增加速度.其原因在于:单位时间内到达对象的数量增加,导致在对每个对象进行分配决策时需要考虑更多的对象.但即使

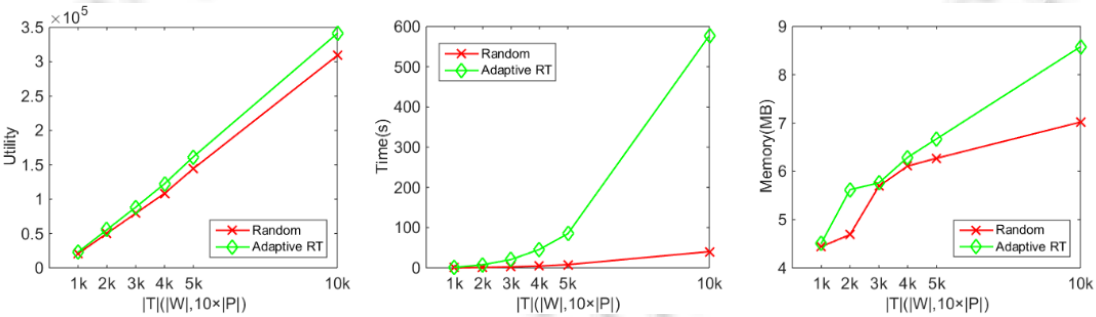
在数据量较大的情况下,每个对象的平均处理时间依然能够满足实时性的要求.



(a) 正态分布



(b) 幂率分布



(c) 均匀分布

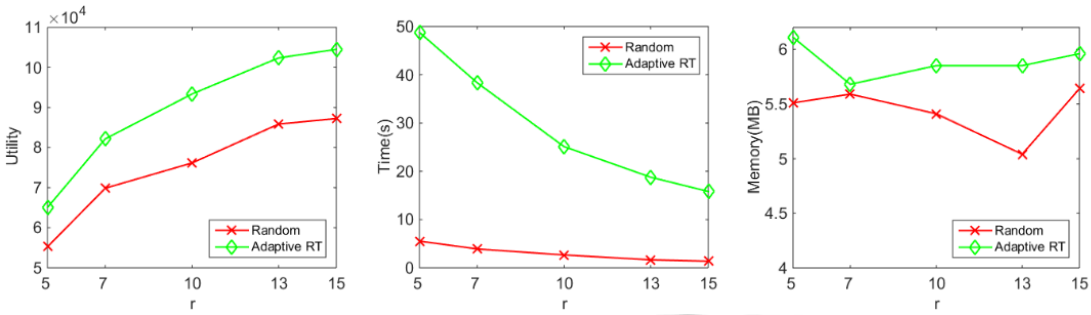
Fig.7 Varying the number of objects

图 7 改变对象数量

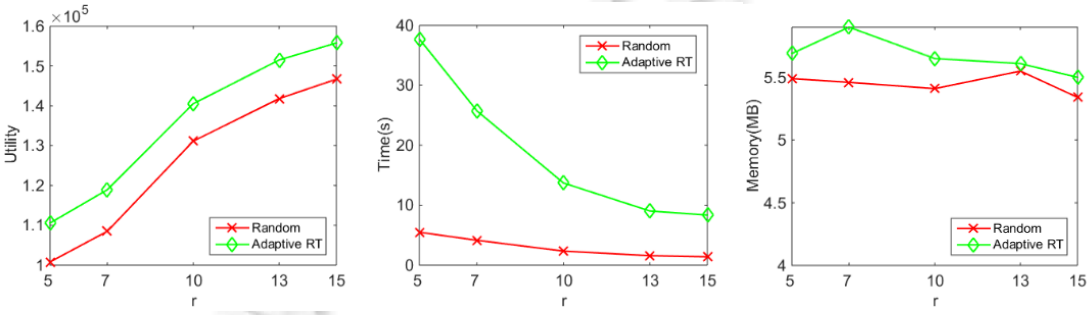
4.2.3 改变半径

在这组实验中,改变任务的半径,其他参数使用表 5 中的默认值.改变任务半径的实验结果如图 8 所示.可以看出:

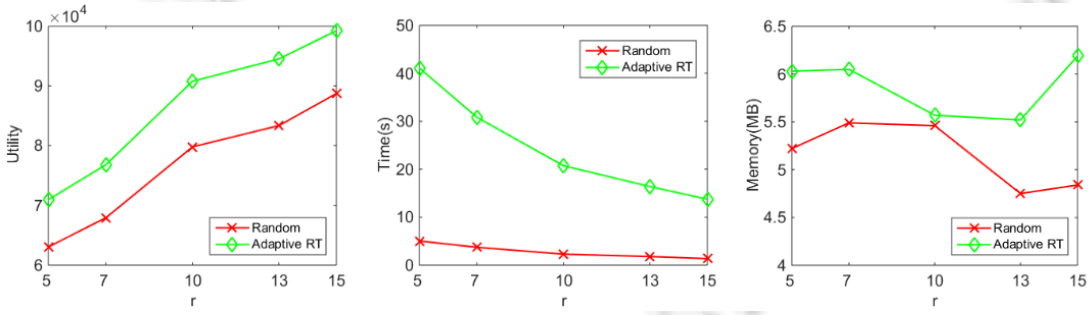
- 在效用方面,自适应随机阈值算法始终优于朴素随机算法.随着任务半径的增加,两种算法返回结果的效用都在增大,这是由于可达成的任务分配数增加;
- 在时间和空间消耗方面,自适应随机阈值算法的运行时间要长于朴素随机算法的运行时间,内存消耗要略高于朴素随机算法.



(a) 正态分布



(b) 幂率分布



(c) 均匀分布

Fig.8 Varying the number of objects

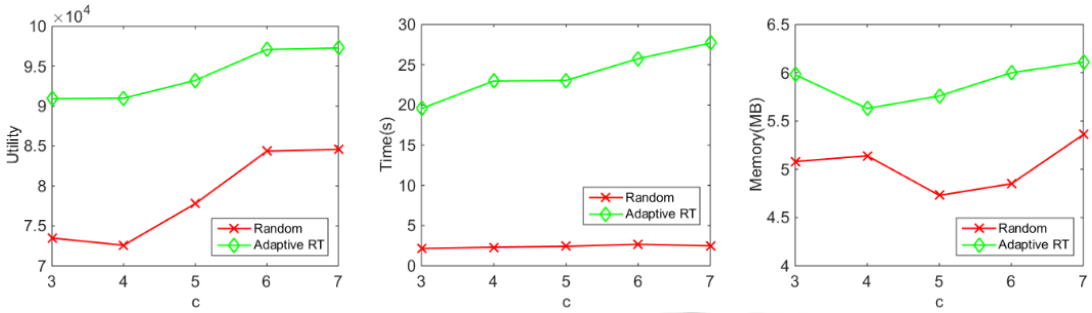
图 8 改变任务半径

4.2.4 改变容量

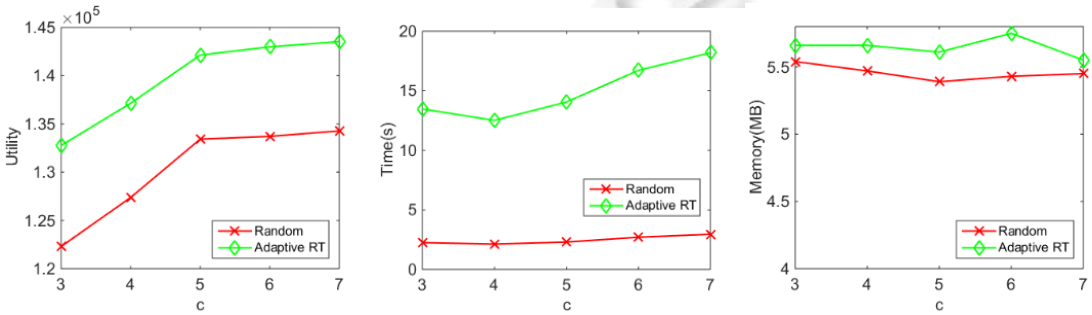
改变众包工作地点容量的实验结果如图 9 所示.

可以看出:

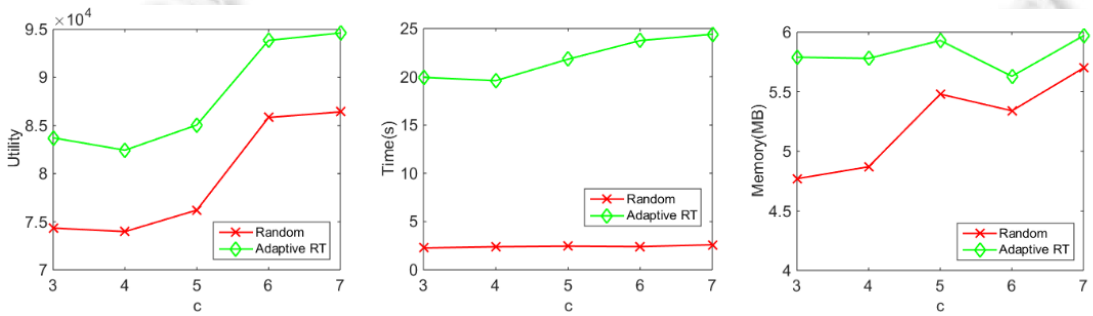
- 在效用方面,自适应随机阈值算法始终优于朴素随机算法.随着众包工作地点容量的增加,两种算法返回结果的效用先较快增长,后缓慢增长.这是由于随着容量增加,可达成的任务分配数增加,导致分配总效用增加.但当容量增长到一定程度,任务分配数趋于饱和,总效用只有少量的增长;
- 在时间和空间消耗方面,自适应随机阈值算法的运行时间要长于朴素随机算法的运行时间,内存消耗要略高于朴素随机算法.



(a) 正态分布



(b) 幂率分布



(c) 均匀分布

Fig.9 Varying the capacity of places

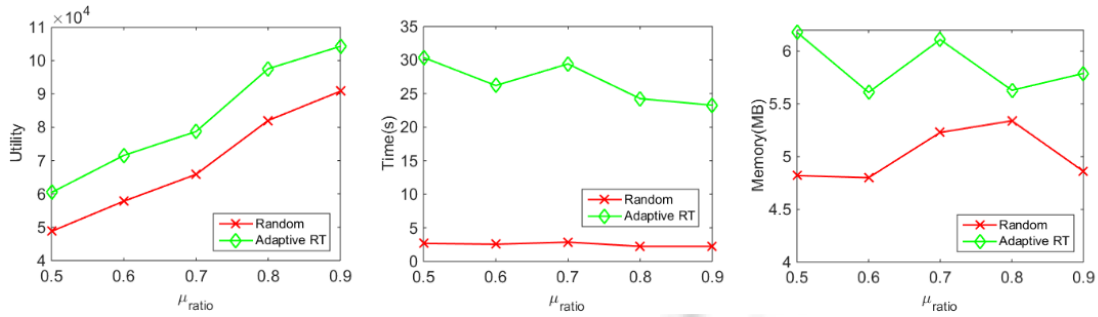
图 9 改变众包工作地点容量

4.2.5 改变服务质量均值

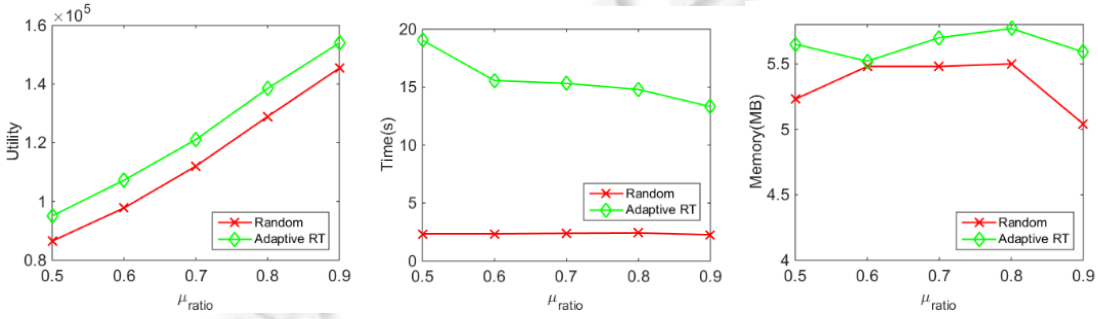
在这组实验中,改变工人服务质量(服从正态分布)的均值,其他参数使用表 5 中的默认值.改变服务质量均值的实验结果如图 10 所示.

从图中可以看出:

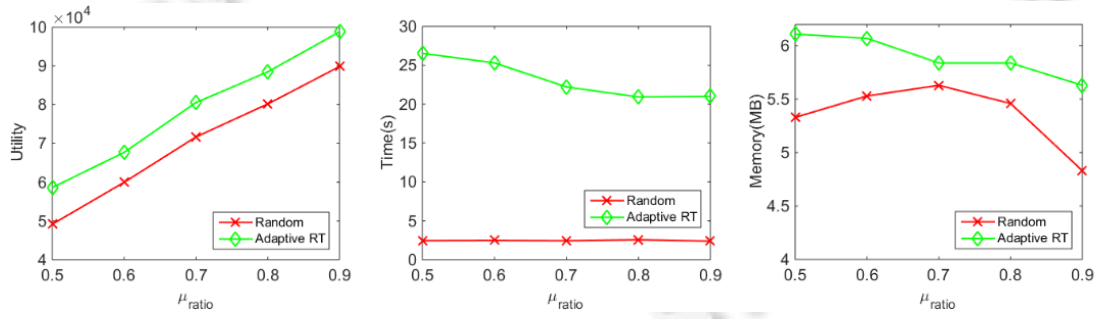
- 在效用方面,自适应随机阈值算法始终优于朴素随机算法,且差距变化不大.随着服务质量均值的增加,两种算法返回结果的效用都增加,这是由于服务质量的增加导致了分配效用的增加;
- 在时间和空间消耗方面,自适应随机阈值算法的运行时间要长于朴素随机算法的运行时间,内存消耗要略高于朴素随机算法.



(a) 正态分布



(b) 幂率分布



(c) 均匀分布

Fig.10 Varying the mean of workers' ratio (normal distribution)

图 10 改变服务质量均值(正态分布)

4.2.6 真实数据

使用空间众包平台 gMission 上收集到的数据进行实验.在 gMission 平台上指定了 300 个工作地点,设置了 3 000 个任务,并收集了前 3 000 个工人的数据.工人和任务的范围设置为 1000m~3000m 不等.实验结果如图 11 所示.

从图中可以看出:

- 在效用方面,自适应随机阈值算法始终优于朴素随机算法;
- 在时间和空间消耗方面,自适应随机阈值算法的运行时间要长于朴素随机算法的运行时间,内存消耗要略高于朴素随机算法.

上述结果显示,所提出算法具有较好的实际应用价值.

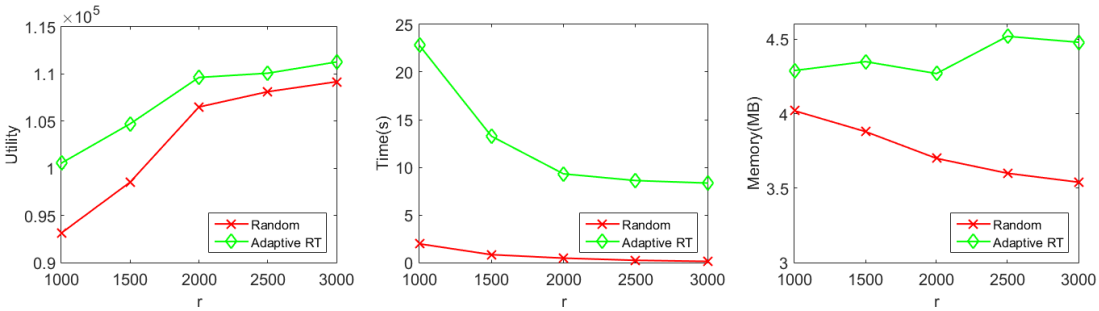


Fig.11 Experiment results on gMission dataset

图 11 gMission 数据集实验结果

4.3 实验总结

通过在真实数据和不同分布数据集上进行实验,发现自适应随机阈值算法在所有情况下都优于朴素随机算法.自适应随机阈值算法的时间消耗相比朴素随机算法较长,但其处理单个对象的时间消耗完全能够满足真实应用的实时性要求.在空间消耗方面,自适应随机阈值算法的内存消耗略高于朴素随机算法.

5 结 论

本文研究了空间众包中的一类新型动态任务分配问题,称为 3 类对象在线任务分配问题.由于在大量现存的空间众包应用中,众包工人与众包任务发布者均需要移动到某第三方工作地点来完成任务,因此本文聚焦于基于 3 类对象的在线任务分配问题.由于该问题所对应的离线问题为经典的难解问题——三维匹配问题,因此本文设计了一种在线算法——随机阈值算法,并给出了该算法在最差情况下的竞争比分析.此外,为了进一步优化随机阈值算法,本文采用在线学习方法将随机阈值算法扩展为自适应随机阈值算法,并证明此优化算法的效果接近随机阈值算法使用不同阈值所能达到的最佳效果.最终,本文通过在真是数据和大量不同分布数据集上的实验,证明了所提出算法不但具有很好的近似效果与伸缩性,而且拥有能够满足实时性要求的时空开销.

References:

[1] Jeff H. Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business. Crown Business, 2009.

[2] Li GL, Wang JN, Zheng YD, Franklin JM. Crowdsourced data management: A survey. ACM Trans. on Knowledge and Data Engineering, 2016,28(9):2296−2319. [doi: 10.1109/TKDE.2016.2535242]

[3] Feng JH, Li GL, Feng JH. A survey on crowdsourcing. Chinese Journal of Computers, 2015,38(9):1713−1726 (in Chinese with English abstract).

[4] Chittilappilly AI, Chen L, Amer-Yahia S. A survey of general-purpose crowdsourcing techniques. ACM Trans. on Knowledge and Data Engineering, 2016,28(9):2246−2266. [doi: 10.1109/TKDE.2016.2555805]

[5] Garcia-Molina H, Joglekar M, Marcus A, Parameswaran AG, Verroios V. Challenges in data crowdsourcing. ACM Trans. on Knowledge and Data Engineering, 2016,28(4):901−911. [doi: 10.1109/TKDE.2016.2518669]

[6] Chen Z, Fu R, Zhao ZY, Liu Z, Xia LH, Chen L, Cheng P, Cao CC, Tong YX, Zhang CJ. gMission: A general spatial crowdsourcing platform. In: Jagadish HV, ed. Proc. of the 40th Int'l Conf. on Very Large Data Bases. Hangzhou: ACM Press, 2014. 1629−1632.

[7] Garey MR, Johnson DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.

[8] Cormen TH, Leiserson CE, Revest RL, Stein C. Introduction to Algorithms. MIT Press, 2009.

[9] Burkard RE, Dell'Amico M, Martello S. Assignment Problems. SIAM, 2009.

[10] Wong RC, Tao YF, Fu AW, Xiao XK. On efficient spatial matching. In: Koch C, ed. Proc. of the 33rd Int'l Conf. on Very Large Data Bases. University of Vienna: ACM Press, 2007. 579−590.

[11] U LH, Yiu ML, Mouratidis K, Mamoulis N. Capacity constrained assignment in spatial databases. In: Wang JT, ed. Proc. of the 27th Int'l Conf. on Management of Data. Vancouver: ACM Press, 2008. 15−28. [doi: 10.1145/1376616.1376621]

[12] Long C, Wong RC, Yu PS, Jiang MH. On optimal worst-case matching. In: Ross KA, ed. Proc. of the 32nd Int'l Conf. on Management of Data. New York: ACM Press, 2013. 845−856. [doi: 10.1145/2463676.2465321]

[13] U LH, Mouratidis K, Mamoulis N. Continuous spatial assignment of moving users. VLDB Journal, 2016,19(2):141−160. [doi: 10.1007/s00778-009-0144-3]

[14] Gao J, Guibas LJ, Milosavljevic N, Zhou DP. Distributed resource management and matching in sensor networks. In: Proc. of the 8th ACM/IEEE Int'l Conf. on Information Precessing in Sensor Networks. San Francisco: ACM Press, 2009. 97−108.

[15] Kann V. Maximum bounded 3-dimensional matching is MAX SNP complete. Information Process. Letter, 1991,37(1):27−35. [doi: 10.1016/0020-0190(91)90246-E]

[16] Papadimitriou C, Yannakakis M. Optimization, approximation, and complexity classes (extended abstract). In: Simon J, ed. Proc. of the 20th Annual Symp. on Theory of Computing. Chicago: ACM Press, 1988. 229−234. [doi: 10.1145/62212.62233]

[17] Hurkens CAJ, Schrijver A. On the size of systems of sets every *t* of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. SIAM Journal on Discrete Mathematics, 1989,2(1):68−72. [doi: 10.1137/0402008·Source: DBLP]

[18] Arkin EM, Hassin R. On local search for weighted *k*-set packing. Mathematics of Operations Research, 1998,23(3):640−648. [doi: 10.1287/moor.23.3.640]

[19] Hassan UU, Curry E. A multi-armed bandit approach to online spatial task assignment. In: Proc. of the 11th IEEE Int'l Conf. on Ubiquitous Intelligence and Computing. Bali: IEEE Computer Society, 2014. 212−219. [doi: 10.1109/UIC-ATC-ScalCom.2014.68]

[20] Ting HF, Xiang XZ. Near optimal algorithms for online maximum edge-weighted *b*-matching and two-sided vertex-weighted *b*-matching. Theoretical Computer Science, 2015,607:247−256. [doi: 10.1016/j.tcs.2015.05.032]

[21] Tong YX, She JY, Ding BL, Wang LB, Chen L. Online mobile micro-task allocation in spatial crowdsourcing. In: Proc. of the 32nd IEEE Int'l Conf. on Data Engineering. Helsinki: Computer Society, 2016. 49−60. [doi: 10.1109/ICDE.2016.7498228]

[22] Tong YX, She JY, Ding BL, Chen L, Wo TY, Xu K. Online minimum matching in real-time spatial data: Experiments and analysis. In: Proc. of the 42nd Int'l Conf. on Very Large Data Bases. New Delhi: ACM Press, 2016. 1053−1064. [doi: 10.14778/2994509.2994523]

[23] Kazemi L, Shahabi C. Geocrowd: Enabling query answering with spatial crowdsourcing. In: Proc. of the 20th Int'l Conf. on Advances in Geographic Information Systems. Redondo Beach: ACM Press, 2012. 189−198. [doi: 10.1145/2424321.2424346]

[24] Kazemi L, Shahabi C, Chen L. Geotrucrowd: Trustworthy query answering with spatial crowdsourcing. In: Proc. of the 21st Int'l Conf. on Advances in Geographic Information Systems. Orlando: ACM Press, 2013. 304−313. [doi: 10.1145/2525314.2525346]

[25] To H, Shahabi C, Kazemi L. A server-assigned spatial crowdsourcing framework. ACM Trans. on Spatial Algorithms and Systems, 2015,1(1):2. [doi: 10.1145/2729713]

[26] Cheng P, Lian X, Chen Z, Fu R, Chen L, Han JS, Zhao JZ. Reliable diversity-based spatial crowdsourcing by moving workers. Proc. of the VLDB Endowment, 2015,8(10):1022−1033. [doi: 10.14778/2794367.2794372]

[27] She JY, Tong YX, Chen L, Cao CC. Conflict-Aware event-participant arrangement and its variant for online setting. ACM Trans. on Knowledge and Data Engineering, 2016,28(9):2281−2295. [doi: 10.1109/TKDE.2016.2565468]

[28] She JY, Tong YX, Chen L, Cao CC. Conflict-Aware event-participant arrangement. In: Proc. of the 31st Int'l Conf. on Data Engineering. 2015. 735−746. [doi: 10.1109/ICDE.2015.7113329]

[29] Gao DW, Tong YX, She JY, Song TS, Chen L, Xu K. Top-*k* teams recommendation in spatial crowdsourcing. In: Proc. of the 17th Int'l Conf. on Web-Age Information Management. 2016. 191−204. [doi: 10.1007/978-3-319-39937-9_15]

[30] Deng DX, Shahabi C, Demiryurek U. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In: Proc. of the 21st Int'l Conf. on Advances in Geographic Information Systems. Orlando: ACM Press, 2013. 314−323. [doi: 10.1145/2525314.2525370]

[31] She JY, Tong YX, Chen L. Utility-Aware social event-participant planning. In: Proc. of the 34th ACM SIGMOD Int'l Conf. on Management of Data. 2015. 1629−1643. [doi: 10.1145/2723372.2749446]

[32] Li Y, Liu LM, Xu WJ. Oriented online route recommendation for spatial crowdsourcing task workers. In: Proc. of the 14th Int'l Symp. on Spatial and Temporal Databases. Hong Kong: Springer-Verlag, 2015. 137−156. [doi: 10.1007/978-3-319-22363-6_8]

[33] To H, Ghinita G, Shahabi C. A framework for protecting worker location privacy in spatial crowdsourcing. Proc. of the VLDB Endowment, 2014,7(10):919−930. [doi: 10.14778/2732951.2732966]

[34] Pournajaf L, Xiong L, Sunderam VS, Xu XF. Spatial task assignment for crowd sensing with cloaked locations. In: Proc. of the 15th Int'l Conf. on Mobile Data Management. Brisbane: IEEE Computer Society, 2014. 73−82. [doi: 10.1109/MDM.2014.15]

[35] Littlestone N, Warmuth MK. The weighted majority algorithm. Information and Computation, 1994,108(2):212−261. [doi: 10.1006/inco.1994.1009]

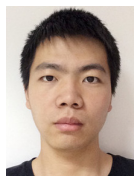
[36] Freund Y, Schapire R. Game theory, on-line prediction and boosting. In: Proc. of the 9th Annual Conf. Conputational Learning Theory. New York: ACM Press, 1996. 325−332. [doi: 10.1145/238061.238163]

[37] Blum A, Burch C. On-Line learning and the metrical task system problem. Machine Learning, 2000,39(1):35−58. [doi: 10.1023/A:1007621832648]

附中文参考文献:

[3] 冯剑红,李国良,冯建华.众包技术研究综述.计算机学报,2015,38(9):1713−1726.

宋天舒(1994－),男,江苏徐州人,CCF 学生会员,主要研究领域为众包数据管理,时空数据管理与挖掘.



王立斌(1995－),男,本科生,主要研究领域为众包数据管理,时空数据管理与挖掘.

童咏昕(1982－),男,博士,副教授,CCF 会员,主要研究领域为众包数据管理,不确定数据管理与挖掘,时空数据管理与挖掘,社交网络分析.

许可(1971－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为算法,数据挖掘与网络.