

## 移动群智感知中基于社区的任务分发算法

龙浩<sup>1,2</sup>, 张书奎<sup>1</sup>, 张洋<sup>1</sup>, 张力<sup>1</sup>

(1. 苏州大学计算机科学与技术学院, 江苏 苏州 215006;  
2. 徐州工业职业技术学院信息与电气工程学院, 江苏 徐州 221002)

**摘要:** 针对移动群智感知 (MCS) 任务分发的有效性和精准性问题, 提出了基于社区的任务分发算法。为了解决所提问题, 该方法首先通过计算移动节点间的最小生成树、连接参量、社区融合度, 抽象和识别出节点的行为模式, 将用户合理划分成不同的社区, 然后计算感知任务与社区行为模式特征值的匹配度, 根据匹配度由社区的中心节点完成任务的分发。仿真结果表明, 与其他算法相比, 所提方法有效地提高了任务分发的精确性和任务完成率, 节约了任务完成的时间成本。

**关键词:** 移动群智感知; 社区; 行为模式; 任务分发; 匹配度

**中图分类号:** TP393

**文献标识码:** A

**doi:** 10.11959/j.issn.1000-436x.2019213

## Task distribution algorithm based on community in mobile crowd sensing

LONG Hao<sup>1,2</sup>, ZHANG Shukui<sup>1</sup>, ZHANG Yang<sup>1</sup>, ZHANG Li<sup>1</sup>

1. School of Computer Science and Technology, Soochow University, Suzhou 215006, China  
2. School of Information and Electrical Engineering, Xuzhou College of Industrial Technology, Xuzhou 221002, China

**Abstract:** A community-based task distribution algorithm was proposed to solve the problem of the validity of mobile crowd sensing (MCS) task distribution. By calculating the minimum spanning tree (MST), the connection parameter (CP) and the community convergence degree (CI) between the mobile nodes, the behavior patterns of the users were abstracted and identified to rationally divide the nodes into different communities. Then, the eigenvalue matching degree of the community behavior patterns with the sensing task was calculated. According to the matching degree, the distribution of the corresponding tasks was completed by the central node of the community. The simulation results show that the proposed method can effectively improve the accuracy of the task distribution and the task completion rate, and save the time cost of the task completion.

**Key words:** mobile crowd sensing, community, behavior pattern, task distribution, matching degree

收稿日期: 2019-05-20; 修回日期: 2019-08-23

通信作者: 张书奎, zhangsk@suda.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61201212); 江苏省高等学校自然科学研究面上项目 (No.19KJB520061); 徐州市应用基础研究计划基金资助项目 (No.KC17074); 苏州市融合通信重点实验室 (No.SKLCC2013XX); 江苏省青蓝工程人才培养计划 (No.102508999008); 苏州市重点产业技术创新前瞻性应用研究项目 (No.SYG201730)

**Foundation Items:** The National Natural Science Foundation of China (No.61201212), The Natural Science Research Project (General Program) of Jiangsu Higher Education Institution (No.19KJB520061), Science and Technology Project of Xuzhou of China (No.KC17074), Suzhou Key Laboratory of Converged Communication (No.SKLCC2013XX), Blue Project of Jiangsu of China (No.102508999008), Prospective Application Foundation Research of Suzhou of China (No.SYG201730)

## 1 引言

移动群智感知 (MCS, mobile crowd sensing) 通过移动终端携带者实现对现实世界的实时感知, 利用用户的移动性和社会性等特性完成感知任务, 并保证了感知服务的实时性和可靠性<sup>[1]</sup>。在目前的应用中, MCS 管理系统通过分析用户之间的移动性、社会性和信任关系, 按照一定的约束机制将感知任务进行合理分发, 以此实现感知任务的有效分发, 并最终提高感知数据质量和效率。MCS 系统结构如图 1 所示。

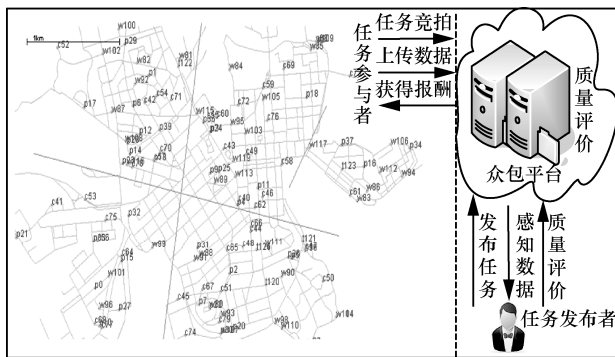


图 1 MCS 系统结构

目前, 人们已经开发了许多群智感知技术的应用, 成为实时监测城市环境 (如空气质量<sup>[2]</sup>、噪声水平<sup>[3]</sup>、交通状况<sup>[4]</sup>等) 的有效手段。在这些应用中, 感知任务分配方法主要是基于基础网络的集中式分发, 系统中的移动终端大多是通过蜂窝网络连接到互联网, 将感知任务分发到参与者的手机中。该方式一方面将耗用户较多的网络流量, 导致收集成本高, 降低用户参与热情, 影响系统可用性; 另一方面也给蜂窝网络带来较大的负载压力<sup>[5]</sup>。

当前的移动终端基本都具备短距离无线通信技术如 Bluetooth 或 Wi-Fi, 当设备进入彼此通信范围时, 可以利用短距离无线技术交换数据, 即支持机会计算模式。将机会网络的特性应用到群智感知中, 形成机会群智感知网络, 该方式实现的核心是众包数据, 任务的分发和数据的收集通过众包来实现, 目前的众包技术通过激励机制进行任务分发, 一般包括集中式或分布式 2 种模式, 主要的焦点集中在任务自身, 而任务自身融合多维环境信息和社会属性, 却很少考虑到用户的社会化信息, 忽略了任务与节点的匹配度。由于人们的参与, 使 MCS 具有强烈的社会属性, 且已经成为 MCS 的首要属

性。另外由于人们之间社会关系相对稳定且具有一定的依赖性, 网络中会出现节点的聚集现象, 从而形成不同的社区。在社区内部, 节点密度较高, 相遇频繁, 移动缓慢, 而处于不同社区的节点相遇概率较低。因此, 针对 MCS 系统这一属性, 以移动节点社区划分、节点社会属性、节点行为模式为基础, 研究基于可信交互的服务众包方法, 提出了一种基于社区的任务分发算法, 该方法通过计算移动节点间的最小生成树、连接参量、社区融合度等参数, 将节点划分成不同的社区, 然后通过节点行为模式和感知任务匹配度选取符合任务需求的社区, 最后选取一个社区中心节点完成感知任务的分发与数据的回收。本文主要的贡献包括以下 3 个方面。

1) 针对现有 MCS 应用中社区划分算法存在划分社区的特征因子单一以及缺乏对社会关系的具体量化等问题, 本文通过计算移动节点间的最小生成树、连接参量、社区融合度, 抽象和识别出节点的行为模式, 将用户合理划分成不同的社区, 该方法能有效地满足 MCS 系统任务分发需求。

2) 设计了社区行为模式特征值与任务特征值的匹配方法, 并选取社区中心节点通过改进的自适应层次喷射等待路由算法完成感知任务的分发。

3) 通过大量的实验, 首先验证了社区划分算法的效果, 与现有算法相比, 所提的方法能够更加准确地检测到网络中具有相似行为特征的社区结构; 然后对比现有的任务分配算法, 验证了所提的方法在任务平均完成时间、任务匹配率和任务分配总效用值等方面都要优于对比算法。

## 2 相关工作

当前随着嵌入大量传感器的移动智能终端设备的技术进步, MCS 应用得到快速发展。在 MCS 研究中, 任务分配是研究的核心点之一。根据考虑角度和研究内容的不同, 任务分配方法主要分为 2 类: 以平台为中心的优化算法和以用户为中心的激励机制。安健等<sup>[6]</sup>提出了一种基于公交系统的任务差异化分发方法, 采用覆盖差异和扩散差异的方法分发任务, 然而该方法在任务分发准确性和节点移动距离方面还有待改进。刘琰等<sup>[7]</sup>提出了一种多任务参与者优选方法, 选出最佳的参与者集合, 使参与者完成任务所移动的总距离最短以降低成本, 然而该方法中任务平均完成时间有待进一步提高。Xiao 等<sup>[8]</sup>根据用户的历史移动数据预测用户的移动

轨迹, 设计了平均完成时间敏感的在线任务分配算法, 然而该方法中等待用户加入的时间过长, 从而拉长了完成任务的平均时间。Yu 等<sup>[9]</sup>提出了一种多任务多用户分配的解决方案, 以用户的活动时间为分组依据, 将用户划分为多个协作组。然后, 利用匈牙利算法从每个任务的协作组中选择每个模块的最优工作者分配任务。然而该方法中用户划分的特征过于单一, 另外任务分配的匹配方式缺乏依据。徐哲等<sup>[10]</sup>提出了一种基于中枢节点的多任务分发 (HTA, hub-based multi-task assignment) 算法, 该算法利用节点在移动网络中社交关系属性不同的特点, 通过中枢节点选择算法将部分节点作为中枢节点, 并将其用于协助任务请求节点分发任务, 然而该方法并未考虑节点的社交属性与任务属性的匹配度。杨玉仁等<sup>[11]</sup>提出一种基于社交属性和有效用户计算的群智感知应用任务分发算法, 针对感知数据质量较低和参与者不足等问题, 通过社交网络的有效传递, 动态调整实时有效用户完成任务分发。该方法在数据质量和任务分发效率方面有较大提高, 然而该方法采用贪婪算法选取可信度较高用户转发任务, 存在一定的局限性, 可能导致任务扩散较慢, 从而降低任务分发效率。Wang 等<sup>[12]</sup>提出了一种基于时空相关性的群智感知多任务分配方法, 利用任务间的隐式时空相关性, 采用两阶段任务分配方法, 根据用户的偏好搜索对应的用户分发任务。然而该方法任务匹配搜索对应的用户需要花费较大的时间成本。张君涛等<sup>[13]</sup>提出了一种渐进式的任务分发方法, 该方法将任务绑定到任务点, 当移动节点到达任务点时自动接收任务, 并渐进式地增加任务点的任务数量。该方法虽然提高了任务分发成功率, 但是数据收集的质量有待提高。目前机会网络路由算法的提出为 MCS 网络中任务的分发和数据的收集提供了途径,

本文针对 MCS 网络特点设计一种基于层次的喷雾等待路由算法来实现任务的分发。

### 3 社区划分

现有社区划分算法存在划分社区的特征因子单一以及缺乏对社会关系的具体量化等问题。针对这些问题, 本文提出了一种基于多维社会关系特征的社区划分算法。通过量化节点之间的多维社会关系, 将节点聚类成具有真实社会关系的社区网络。针对网络中的各个节点, 首先构造出节点的最小生成树, 形成初始社区, 然后计算相邻节点间的连接参量, 通过连接参量来判断节点是否处于同一社区, 完成最小生成树的合并, 形成明显的社区结构, 最后用社区融合度来检验社区划分的紧密度, 衡量社区结构划分效果, 并进行适当调整, 从而完成网络中社区的最佳划分, 实现用户接受任务的最佳优化。该方法量化了节点间的多维社会关系和行为特征, 并衡量了节点间的社会关联度, 考虑了节点间的实际位置关系, 通过社区融合度有效地处理了重叠社区的节点划分问题。所提的方法主要适用于 MCS 系统中任务与用户群的匹配分发问题。具体社区划分过程如图 2 所示。

#### 3.1 最小生成树

假设  $G(V,E)$  是无向加权网络, 节点集合  $V$  包含了  $n$  个节点, 边集合  $E$  包含了  $e$  条边, 每一条边有 2 个顶点  $(v_i, v_j)$  在  $V$  中。社区划分的目标就是要将  $G$  划分成  $k$  个社区:  $C = \{c_1, c_2, \dots, c_k\}$ , 其中  $c_i \neq \emptyset, c_i \cap c_j = \emptyset, (i=1:k, j=1:k), V = \bigcup_{i=1}^k c_i$ 。

定义 1 从网络  $G$  中的节点  $v_i$  开始, 直到包含所有的节点集合  $V$  截止, 以任意节点  $v_i$  为开始的最小生成树路径值的计算式为

$$J_{ij} = \min \{ \varphi(i, j) + \varphi(j, x) + \dots + \varphi(y, n) \} \quad (1)$$

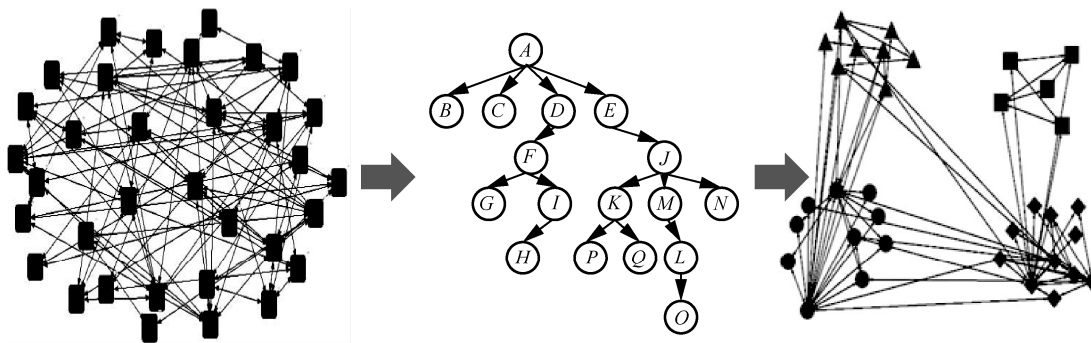


图 2 社区划分过程

其中,  $j, x, y$  为中间相邻节点, MST 的构造是通过所有相邻节点的权值  $\varphi$  最大值求和<sup>[14]</sup>,  $\varphi(i, j)$  为计算两相邻节点的权重值, 称作两节点的社会关系值, 具体计算式为

$$\varphi(i, j) = \frac{|\Gamma(v_i)| + |\Gamma(v_j)|}{|\Gamma(v_i) \cap \Gamma(v_j)|} \quad (2)$$

其中,  $\Gamma(v_i)$  表示节点  $i$  的邻居集。

MST 的建立过程类似最小生成树算法 Prim 的原理, 是以网络中的任意节点  $v_i \in V$  作为根节点, 将各连接路径上最大权值相加。网络  $G(V, E)$  的一棵最小生成树设为  $MST(V', E')$ ,  $V' = V$ ,  $|V'| = |V|$ ,  $E' \subset E, E' = V' - 1$ 。在 MST 中, 本文选择边权值最大的  $\frac{n-1}{2}$  条边, 设为集合  $E^*$ , 然后将 MST 中在集合  $E^*$  中的边移除, 得到了  $\frac{n+1}{2}$  个相互独立的部分, 本文把它称为初始社区, 记为  $C = \{c_1, c_2, \dots, c_h\}$ ,  $h = \frac{n+1}{2}$ 。

### 3.2 连接参量

目前大多数社区划分算法通过计算节点间的相似度和多特征值<sup>[15]</sup>来实现, 然而由于聚类方法的不同, 节点间相似度和特征的定义在不同的算法中各不相同。考虑到 MCS 的应用场景, 划分的最终目标是根据节点区域来分发不同的任务, 因此社区的划分主要是根据节点间的社会关系和位置聚类形成的, 本文社区划分方法提出节点连接参量的定义, 将节点的连接参量定义为节点间社会关联度和位置特征, 通过连接参量来实现初始社区的合并, 形成明显的社区结构。本文将连接参量化为 2 个值: 第一个值是根据节点历史相遇记录引入的社会压力度量值 (SPM, social pressure metric), 用来探测节点间连接质量  $\pi_{i,j}$ <sup>[16]</sup>, 衡量节点间的社会关联度; 第二个值是通过 GPS 获取移动用户的位置特征 (经度和纬度) 并采用欧氏距离公式计算两节点的距离作为节点位置特征。社区的合并需要根据节点间的联系紧密度来计算, 一般来说 2 个节点属于朋友或者邻居, 可以认为这 2 个节点联系紧密, 具有较高的连接质量。然而, 存在朋友关系的节点对可能会处在距离比较远的 2 个区域, 根据感知任务区域划分的特点, 这 2 个节点不能划分到同一社区, 因此, 本文需要考虑采用节点的位置特征值来进行校验。下面具体对连接参量进行定义。

**定义 2** 连接参量包括节点间社会关联度和位置特征, 节点之间的社会关联度定义为 SPM 的倒数。节点间的位置特征值由欧氏距离公式<sup>[17]</sup>计算获得。

$$\begin{cases} \pi_{i,j} = \frac{1}{\text{SPM}} = \frac{2T}{\sum_{i=1}^r t_{\text{inter},i}^2} \\ d(v_i, v_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \end{cases} \quad (3)$$

其中,  $T$  为任务执行周期,  $t_{\text{inter}}$  为节点相遇间隔时间,  $r$  为间隔次数。下面介绍具体合并过程。首先计算初始社区中任意 2 个社区节点对的连接质量是否大于阈值  $\psi$ , 如果大于阈值则计算节点间距离是否小于阈值  $\varepsilon$ , 2 个条件都满足则建立 2 个节点的连接, 并将 2 个节点保存到同一社区  $c_i$  中, 直到所有的社区不能再合并, 形成的  $\{c_1, c_2, \dots, c_k\}$  即为合并的社区划分。为了降低节点间连接参量的计算和判别, 本文考虑如果 2 个初始社区中任意节点对之间的连接质量和距离满足阈值, 则将 2 个社区进行合并。

在社区合并过程中, 可能会有以下问题。首先, 假设节点在一定时间内处于有限的迁移场景中。事实上, 节点的活动具有一定的特点, 在某个时间段, 节点的活动范围一般并不会超出所属社区的范围, 尤其在接收感知任务期间。第二个问题是有部分节点移动到别的区域或者有新的节点加入。首先可以根据移动节点的历史信息计算是否可以加入对应社区, 如果不满足条件, 本文计算移动节点或新节点与邻居节点的距离, 如果小于阈值  $\varepsilon$  则加入邻居节点社区。第三个问题是在某个时间段后, 大部分节点进行了移动, 社区需要进行重新划分。社区的变化特征超出了本文的范围, 本文将其作为未来的工作。

### 3.3 社区融合度

考虑到在实际网络中社区结构是自然形成的, 社区内部各个节点的连接比社区间节点的连接要紧密得多, 然而这只是一个定性的指标。本文提出了社区融合度的概念, 从定量角度来衡量社区划分的质量, 对一些社区间的重叠节点做进一步优化, 重叠节点即节点  $i \in u_k$ , 同时  $i \in c_j$ , 优化的目的是将重叠节点  $i$  划分到对应社区中。

**定义 3** 社区融合度通过任意节点  $i$  的度来衡量社区划分质量。如果  $\forall i \in c_k$ , 社区  $c_k$  应满足

$D_i^{\text{in}}(c_k) \geq D_i^{\text{out}}(c_k)$ 。其中,  $D_i^{\text{in}}(c_k)$  表示节点  $i$  与社区  $c_k$  内中其他节点连接边的条数,  $D_i^{\text{out}}(c_k)$  表示节点  $i$  与社区  $c_k$  外其他节点连接的边数。如果社区内任意节点与社区外部节点的连接, 比它与社区内部节点的连接更紧密, 那么就将其节点调节到对应社区。

### 3.4 社区划分算法

社区划分算法实现过程主要包括 3 个步骤: 1) 根据式(2)计算各节点的权值, 构造一棵最小生成树, 然后删除最小生成树中集合  $E^*$  中的边, 将最小生成树分裂成  $h$  个初始社区; 2) 根据式(3)计算相邻节点对的连接参量, 判断连接参量是否满足阈值, 将初始社区进行合并, 重复该步骤, 直到所有的初始社区全部合并; 3) 判断社区融合度指标, 衡量社区划分质量, 调节不满足定义 3 的相关节点到对应社区。

#### 算法 1 社区划分算法

输入 网络  $G(V,E)$ , 相邻节点连接参量阈值  $\psi$  和  $\varepsilon$

输出 划分完整的社区结构集合  $C=\{c_1,c_2,\dots,c_k\}$

- 1) for  $V$  中的每个节点对  $(v_i,v_j)$
- 2) 基于式(2)计算  $\varphi_{i,j}$
- 3) end for
- 4) 构建 MST  $T$
- 5) 从  $E$  中移除  $\frac{n-1}{2}$  条最大权重的边
- 6) 产生  $\frac{n+1}{2}$  个社区  $C=\{c_1,c_2,\dots,c_h\}$
- 7) for  $C$  中每个社区对  $(c_i,c_j)$
- 8) if  $\forall \pi_{i,j} > \psi$  和  $d(v_i,v_j) < \varepsilon$  then
- 9) 合并  $c_i$  和  $c_j$  为  $c_i$
- 10) 删除  $c_j$
- 11) end if
- 12) if 所有的社区不能再合并 then
- 13) break;
- 14) end if
- 15) end for
- 16) 得到合并后的社区集合  $C=\{c_1,c_2,\dots,c_k\}$
- 17) while  $i$  from 1 to  $k$  do
- 18) if  $D_i^{\text{in}}(u_k) \geq D_i^{\text{out}}(u_k)$
- 19) continue

- 20) else
- 21) 调整  $i$  到对应社区
- 22) end if
- 23) end while
- 24) return  $C=\{c_1,c_2,\dots,c_k\}$

算法 1 中节点对的连接质量判断阈值  $\psi$  需要根据任务的时间周期来决定, 连接距离判定阈值  $\varepsilon$  需要根据感知任务区域的半径来决定, 一般  $\varepsilon \in (50, 250) \text{ m}^{[18]}$ 。算法复杂度分析, 首先最小生成树的建立类似于 Prim 算法, 其运行时间是  $O(|E|+|V|\log|V|)=O(m+n\log n)$ , 然后将初始社区合并时间复杂度为  $O(n^2)$ , 社区融合度检验时间复杂度为  $k$ 。因此整个社区划分算法的时间复杂度为  $O(n^2+n\log n+m)$ 。

## 4 社区中心节点选取与任务分发

MCS 系统中, 感知任务的执行主要由移动终端的携带者——人来完成, 尽管人的移动性是不可控的, 但也不是随机的。大部分人都有自己的活动范围, 比如家、公司、商场等。因此感知系统能够提前根据参与者历史数据信息将参与者划分到某个社区。一个感知任务可以定义成一个元组(任务内容、说明、位置、时间), 其中任务内容指的是各种感知操作, 比如感知噪音, 拍一张照片, 拍一段视频等; 说明指的是感知任务的要求和参数, 比如录音时长、照片的像素、视频格式等; 位置包括感知任务区域的中心坐标和感知半径; 时间指的是感知任务执行的时效性。本文考虑根据感知任务的位置属性来决定感知任务的分发, 具体方法是将感知任务位置属性与划分好的社区节点位置进行匹配, 判断是否在感知半径范围内, 如果位于感知半径范围内的节点数达到了任务需求的节点数, 则将该任务分发给该社区中节点。当新到达的参与者想要参与感知任务时, 社区节点也可以将感知任务转发给他, 实现新参与者的动态加入。

本文方法中将 MCS 任务分配给对应的社区, 从而节省网络开销, 保证感知任务完成的质量。为了进一步加快任务的分发速度, 需要选取一个社区的中心节点来实现社区任务的分发。具体流程如图 3 所示。

### 4.1 社区中心节点的选取

感知任务的分发主要分 2 个步骤: 首先计算社区行为模式特征值与任务的匹配度, 然后确定社区

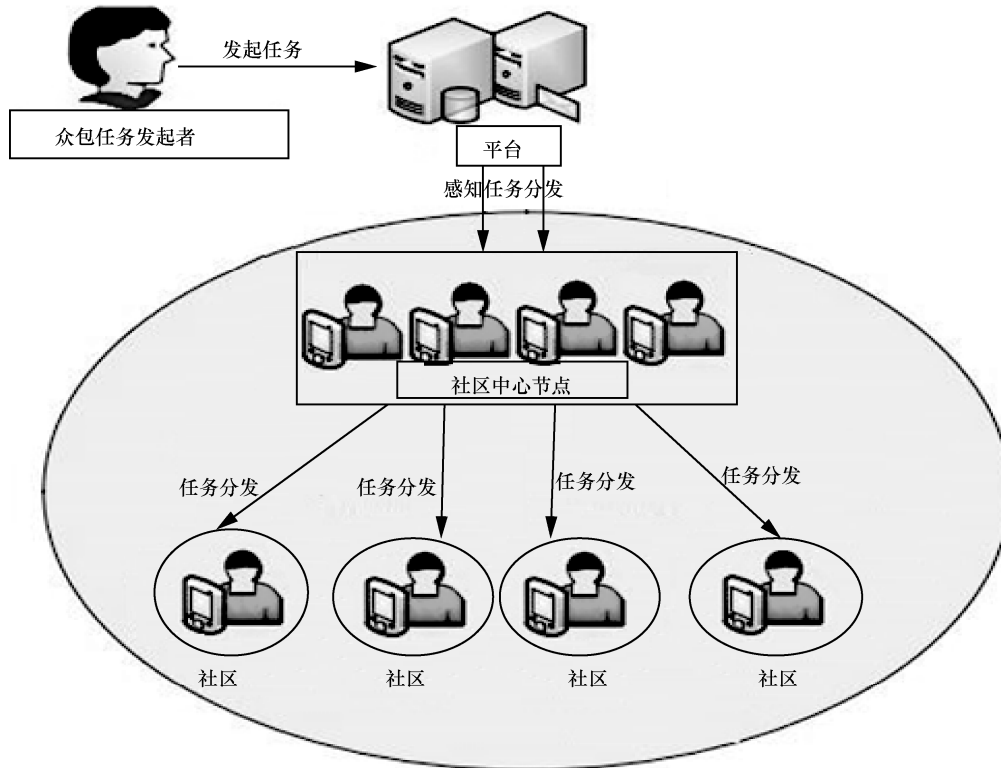


图 3 众包任务节点选取与任务分发流程

的中心节点，将感知任务传递给中心节点。利用机会网络自适应层次喷射等待路由算法，中心节点将任务分发给社区其他节点。为了实现任务分发首先进行如下定义。

**定义 4** 社区行为模式特征值  $O(c_i)$  主要体现社区内部节点的个体特征，通过计算社区节点对连接参量  $CP(v_i, v_j)$  的平均值来获得，计算式为

$$O(c_i) = \frac{\sum_{i,j=1}^{|c_i|} CP(v_i, v_j)}{|c_i|}, \quad i, j \in c_i \quad (4)$$

其中，子社区  $c_i = \{v_1, v_2, \dots, v_n\}$ ， $|c_i|$  为社区节点数。

**定义 5** 感知任务在选择社区的过程中根据任务自身的特征值和需求，寻找匹配的社区行为模式，应满足社区行为模式特征值  $O(c_i)$  与任务特征值匹配度不小于  $K$ ，即满足以下条件

$$O(c_i) > K$$

其中， $K$  为感知任务自身特征值，理想情况下，等于  $K$  也能满足任务的需求，但是感知任务自身的特征以及社区行为模式特征存在一定的随机性和不确定性，因此要求  $O(c_i)$  大于  $K$ ，以维持一定的冗余和错误容忍性。

**定义 6** 社区中心节点是通过各节点与社区其他节点的边的条数来确定的，社区中某节点与其他节点的边的条数越多，说明其与其他节点的连接越紧密。具体计算式为

$$y = \max y = \max(k_i^{in}(c_i)), \quad \forall c_i \in C, \forall i = 1, 2, \dots, n \quad (5)$$

其中， $k_i^{in}(c_i)$  表示节点  $i$  连接社区  $c_i$  中其他节点的边的条数，当  $k_i^{in}(c_i)$  达到最大时， $v_i$  即为社区中心节点。

#### 4.2 感知任务分发

社区中心节点负责将感知任务分发给社区内的其他节点，由于节点的相遇是随机的，通信的链路时常中断，因此节点间任务分发以“存储—携带—转发”的路由模式实现相互通信。不同于机会网络，MCS 网络并不需要知道感知任务分发的目的节点，其目标可以是任意一个或多个移动节点。本文任务分发提出了一种新的自适应层次喷射等待路由算法，具体实现步骤如下。

**喷雾过程。**社区网络  $c_i$  设定社区中心节点为网络的第一层，拥有感知任务副本的中心节点  $v_1$  将副本传递给第二层没有获得任务副本的节点，直到整个社区全部拥有任务副本。喷雾过程如图 4 所示。

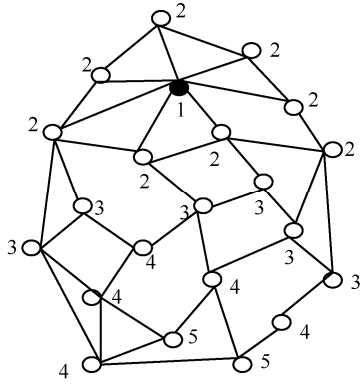


图 4 节点喷雾过程

等待过程。进入等待过程的节点有 2 类：一类是在节点所处的那层其他节点还没有获得任务副本；另一类是该节点已经完成将任务副本传递给它的下一层。

层次喷雾等待算法与机会网络喷雾等待算法不同的地方在于，后者在寻找到目的节点的过程中，需要考虑增加消息副本数量、缓存容量、路由代价、传递时延。而 MCS 网络中，目的是将任务副本传递给社区所有节点，在传递副本时通过层次传递的方法，路由开支已达到最低。因此层次喷雾等待路由算法的性能评估主要是传递时延，传递时延主要由两部分组成：相遇时间和等待时间。具体通过以下定义来计算。

**定义 7** 社区网络中由于人的社会属性和行为特征基本类似，节点间的相遇比较有规律，因此节点间相遇间隔时间可以根据历史的多次相遇间隔时间的平均来计算。图 5 展示了节点与邻居的历史相遇间隔时间。具体的计算式为

$$T_{\text{average}} = \frac{\sum_{i=1}^{|v_i|} \sum_{j=1}^n T_j}{n|v_i|}, v_i \in c_i \quad (6)$$

其中， $T_{\text{average}}$  为平均相遇间隔时间， $n$  为平均相遇次数， $|v_i|$  为  $i$  节点邻居节点的个数。

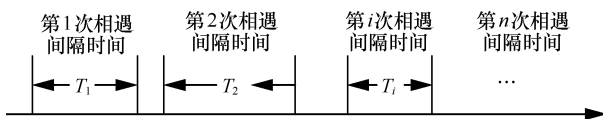


图 5 节点与邻居节点相遇间隔历史

当节点与邻居节点的平均相遇间隔时间与节点等待时间的总和大于或等于感知任务目标时延时，上层节点需要考虑重传感知任务，具体判定

式为

$$T_{\text{average}} + T_{\text{wait}} < T_{\text{delay}} \quad (7)$$

其中， $T_{\text{wait}}$  为节点等待同一层邻居节点获得任务副本的等待时间， $T_{\text{delay}}$  为感知任务自带的目标时延。

### 4.3 算法实现

感知节点发现与任务分发算法主要包括 3 个步骤：1) 根据社区行为模式特征值  $O(c_i)$  与任务特征值匹配度来确定符合感知任务需求的社区；2) 根据式(5)确定社区中心节点，由中心节点负责整个社区任务分发；3) 由于社区内部节点相遇符合机会网络特征，但是 MCS 网络与机会网络在很多方面又存在不同，因此提出一种层次喷雾等待路由算法，以中心节点为起始节点，逐层往外进行喷雾任务，直到社区所有节点全部获得任务副本。

#### 算法 2 任务分发算法

输入 划分好的社区网络  $C = \{c_1, c_2, c_3, \dots, c_n\}$

输出 完成子社区内全部节点的任务分发

- 1) for  $c_i$  ( $i$  from 1 to  $n$ ) do
  - 2) for  $c_i$  中的每一个节点对  $(v_i, v_j)$
  - 3) 计算  $a[i] = O(c_i)$
  - 4) end for
  - 5) end for
  - 6) for  $i$  from 1 to  $n$  do
  - 7) if  $a[i] > k$
  - 8) 选择  $c_i$
  - 9) end if
  - 10) end for
  - 11) for  $c_i$  中的每一个节点对  $v_i$
  - 12) 根据式(5)计算出  $y$ ，确定出中心节点  $v_i$
  - 13) end for
  - 14) for  $j$  from 1 to  $h$  do //  $h$  为节点传输的跳数
  - 15) for  $h$  层节点中每一个节点  $v_{j,k}$
  - 16) 发送数据给  $v_{j,k}$
  - 17) if 其他邻居没有获得任务
  - 18) 等候
  - 19) end if
  - 20) if 式(7) is false
  - 21) 重传数据
  - 22) end if
  - 23) end for
  - 24) end for
- 算法时间和空间复杂度分析，步骤 1)~步骤 9)

计算每个子社区的平均特征值，并与感知任务特征进行比较，时间复杂度为  $O(n^2+n)$ ，空间复杂度为  $O(n)$ ；步骤 10)~步骤 12) 获取社区中心节点计算，运行时间为  $O(n)$ ，空间复杂度为  $O(1)$ ；步骤 13)~步骤 23) 为层次喷雾等待路由算法，按层次将感知任务传递给社区其他节点，时间复杂度为  $O(n^2)$ ，空间复杂度为  $O(n)$ 。因此整个算法的时间复杂度为  $O(n^2+n)$ ，空间复杂度为  $O(n)$ 。

## 5 仿真实验结果与分析

仿真实验分成 2 个部分，第一部分验证社区划分算法的效果；第二部验证基于社区的任务分发算法的性能。所有实验结果至少进行 1 000 轮后取平均值。

### 5.1 社区划分算法性能分析

#### 5.1.1 实验参数设置

为了验证社区划分算法的划分效果，本文采用 LFR 作为测试网络进行社区发现实验，LFR 是目前社区发现算法实验的基准测试网络，也是最为常用的模拟数据集，它模拟了真实网络中节点度和社区大小的无标度性质，用来评价算法发现社区的质量。通过设置不同的混合参数，能够生成不同类型的模拟网络，节点间的权重随机生成。使用 Java 语言在 myclipse 编程实现。本文算法与常见典型社区发现算法进行了对比实验，采用的对比算法主要包括：基于增量密度的社区发现算法 (IncOrder)<sup>[19]</sup>，基于最优生成树和模块的社区划分算法 (CDMSTD)<sup>[14]</sup>。采用归一化互信息 (NMI, normalized mutual information)、调整兰德系数 (ARI, adjusted rand index) 和模块度量  $Q$  值这 3 种评价方法评价社区发现算法的效果，验证社区结构的有效性、合理性和正确性。归一化互信息用来度量通过算法划分社区和真实社区之间的差异程度。当算法划分的社区与原始社区完全一致时， $NMI=1$ ；当完全不一致时， $NMI=0$ 。调整兰德系数<sup>[19]</sup>可以用来评价社区发现算法的效率和划分社区与真实社区的吻合程度。ARI 越接近于 1，表示社区之间的关系越独立；ARI 越接近于 0，表示社区之间的关系越紧密。实验中 LFR 基准网络设置参数如表 1 所示。

表 1 LFR 基准测试网络参数设置

网络	$n$	$m$	$k$	minc	maxc	mu
a	10 000	98 325	20	20	100	0.1~0.8
b	10 000	98 135	20	10	50	0.1~0.8

在表 1 中，本文列出 a 和 b 这 2 种综合网络的参数，其中 a 代表网络划分的社区数相对较少，b 代表网络划分的社区数相对较多。 $n$  表示节点的总数； $m$  表示节点之间边的总数； $k$  表示网络中节点间平均权重；minc 表示最小社区包含的节点的数量；maxc 表示最大社区包含的节点的数量；mu 表示节点与社区外部连接的概率，定义为重叠参数。LFR 基准网络中，mu 的值越大，社区发现的难度就越大。

在社会网络社区划分算法研究中，模块度量  $Q$  值是目前常用的一种衡量社区结构稳定度的方法，社区内部的节点连接紧密，社区之间的节点连接稀疏。在社区中，同一社区内的节点交互相对社区之间的节点交互比较频繁，各个社区均是相对稳定的结构，社区结构越稳定表明社区划分结果越好。因此，本文用模块度量  $Q$  值作为社区划分好坏的衡量标准。模块度量  $Q$  值本是为了用于衡量无权社会网络结构的划分结果，由于考虑到连接强度  $S_{i,j}$ ，本文采用改进的模块度量值，具体计算式为

$$Q = \frac{1}{2S} \sum_{i,j} \left[ S_{i,j} - \frac{S_i S_j}{2S} \right] \delta(C_i, C_j) \quad (8)$$

其中， $S = \sum_{i \in N, j \in N} S_{i,j}$  是所有节点间的连接强度和；

$S_i = \sum_{j \in N} S_{i,j}$ ， $S_j = \sum_{i \in N} S_{i,j}$  分别是社区节点  $i$  和  $j$  与其所有邻居节点的连接强度和； $S_{i,j} = \alpha \varphi(i, j) + (1 - \alpha) \pi_{i,j}$ ， $\alpha$  为社会关系影响因子。当网络具有明显的社区结构时， $Q$  接近于 1，当网络中社区结构不明显时， $Q$  接近于 0。在实际社区划分算法中  $Q$  通常在 [0.3, 0.7] 范围，本文通常认为  $Q > 0.3$  时算法划分具有较好的社区结构。

#### 5.1.2 性能评价与分析

在 LFR 基准网络中，针对每个重叠参数 mu 值，在实验中随机生成 20 个模拟网络，首先针对 3 个算法分别在 20 个模拟网络上运行，分别得到它们的 NMI 和 ARI 的平均值。算法对比分别如图 6 和图 7 所示。

从图 6(a) 和图 7(a) 中可以观察到本文算法相比其他 2 种算法具有较优的社区检测结果。尤其当  $mu < 0.4$  时，本文算法的 NMI 和 ARI 值接近于 1，此时的网络社区结构比较清晰，能够很准确地检测到网络中的社区结构，得到的划分结果几乎完全接近正确的划分结果，而检测结果最差的是 CDMSTD。当  $0.4 < mu < 0.8$  时，各个算法的 NMI 和 ARI 值都随着 mu 的增大迅速下降，此时的网络结

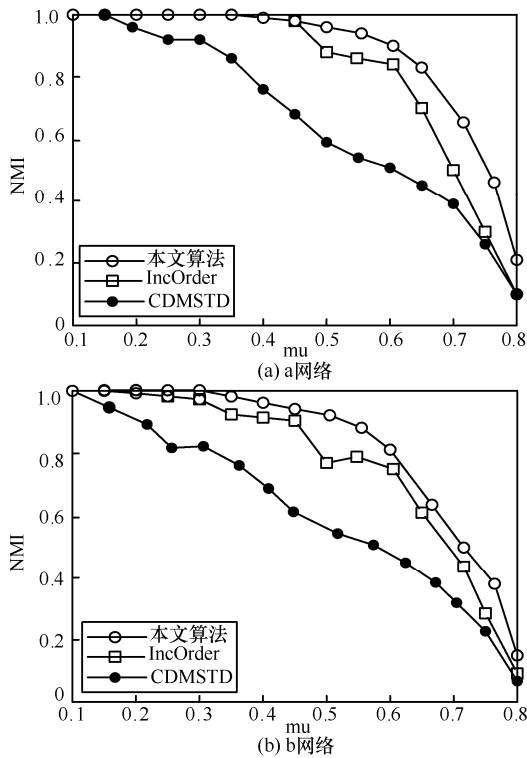


图6 LFR 基准网络中 NMI 值的算法对比

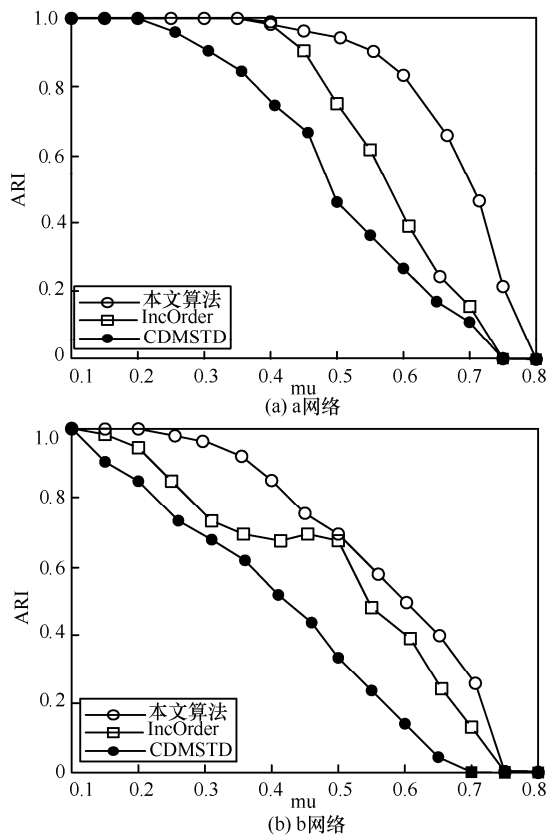


图7 LFR 基准网络中 ARI 值的算法对比

构中, 社区划分比较混杂, 社区结构内部的紧密度较小, 而社区之间的联系反而较大, 社区划分效果

都较差, 然而, 总体来看本文的算法得到的社区划分结果相比其他 2 种算法都较优。在图 6(b) 和 7(b) 中, 同样可以观察到本文的算法能够取得较优的社区划分结果。特别地, 当  $\mu < 0.3$  时, 本文算法的 NMI 和 ARI 值接近于 1, 得到的划分结果也都是几乎接近完全正确的划分结果。当  $0.4 < \mu < 0.8$  时, 本文的算法得到的划分结果相比其他 2 个算法也都较优。针对 a 网络中  $0 < \mu < 0.6$  计算平均 ARI, 本文的算法准确性平均达到 97.3%, 平均高于其他 2 种算法 15.2%。

接下来, 通过网页抓取百度地图获取某个城市各个社区节点的基本属性以及节点之间的实际地理位置距离, 作为划分算法的数据依据来验证社区划分的模块度量  $Q$  值。由于实验数据是从真实的城市社区获取, 为验证所提算法的准确性, 可以将实验结果与某城市结构进行比较。已知真实社区的模块度量值为 0.616 5, 采用遍历社会关系影响因子  $\alpha$  的方法得到当  $\alpha = 0.252 3$  时, 本文算法所得模块度量值与真实社区的模块度量值很接近, 具体如图 8 所示。且当参数在区间  $[0.252, 0.542]$  时, 本文算法不仅能达到真实的社区结构, 并且优于现有实际社区结构的模块度量  $Q$  值。

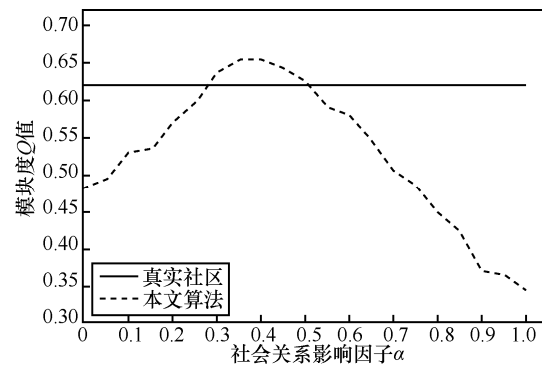


图8 模块度  $Q$  值的比较

### 5.2 任务分发算法性能分析

#### 5.2.1 实验参数设置

任务分发算法相关性性能采用 ONE 平台对其进行仿真验证<sup>[20]</sup>, 选用基于中枢节点的多任务分发算法 (HTA)<sup>[10]</sup>和在线任务分配算法 (NTA, online task assignment)<sup>[21]</sup>进行对比实验。使用现实世界数据集 ParticipAct<sup>[22]</sup>和 MDC Nokia<sup>[23]</sup>分别设计 MCS 应用系统任务分发实验, 通过数据集中节点的历史记录信息, 抽象出节点的行为特征, 然后采用社区划分算法将节点合理划分成几个社区, 并根据感知任务的特征分配给节点任务, 从而验证社区划分后对

任务分发的相关影响。不同任务集的任务平均完成时间提升百分比如图 9 所示。

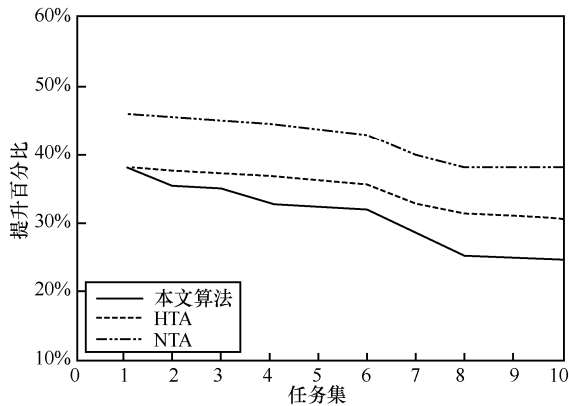


图 9 不同任务集的任务平均完成时间提升百分比 (ParticipAct)

ParticipAct 数据集是一个关于 MCS 系统的现实实验数据集, 涉及超过一年时间的博洛尼亚大学 170 名学生彼此相遇记录和社交关系网络数据。本文使用 869 725 个消息样本进行抽象和识别出节点的行为模式, 使用 536 216 个消息样本用于模拟实验, 将用户节点的数量设置为  $L=170$ 。MDC Nokia 数据集是从 2009 年到 2011 年 3 月收集日内瓦湖地区 185 名用户数据, 在实验中用户携带了一个诺基亚 N95 手机和应用程序定期收集数据, 包括蓝牙、GPS、电话呼叫、短信等传感器数据。所有采集数据 (包括 GPS 和蓝牙) 的采样周期为 600 s。本文从 MDC Nokia 数据集中选择了 1 120 000 个数据点的时间序列, 然后根据 32 122 个样本进行抽象, 识别出节点的行为模式, 6 538 个样本用于模拟实验, 用户节点的数量设为  $L=185$ 。本文将参与者分成不同的社区来分配任务, 并将一些传感信息混合到数据集中来实现本文的方法。这 2 个数据集提供了用户参与 MCS 活动的真实结果, 它具有 4 个维度的特征: 1) 接触频率的迁移率较高; 2) 接触的异质性较低; 3) 接触时间连通性较低; 4) 可利用的传感器数据丰富。这些数据很好地捕捉到了人们在一个城市之间的偶遇, 模拟用户全天使用智能手机的情况。因此, 本文使用不同的任务分配方法进行实验, 其中数据集混合了不同的信息来源来评估本文提出的方法的性能。

为了充分验证任务分发算法的性能, 本文将每个数据集中的每个节点都分配一次感知任务集, 利用节点所在不同的社区来展现算法的性能表现。另外, 实验中本文通过程序随机生成任务集, 分发的任务集由从小到大的工作量任务组成, 任务集编号

为 1~10, 其中任务数量设为 10, 对应平均任务工作量为 500 s、600 s、...、1 000 s、2 000 s、...、5 000 s。为了简化操作, 任务工作量定义为感知时长。实验的评价指标包括任务平均完成时间和平均任务完成率。一个感知任务的完成时间等于  $t_1+t_2+t_3$ , 其中,  $t_1$  为任务分发者开始分发任务到用户接收到该任务的时长,  $t_2$  为用户采集感知数据的时长,  $t_3$  为用户将感知数据提交给任务分发者的时长。实验中每个用户都会分配一个任务集, 实验的目的是分析哪种算法的分配策略可以使任务集中任务平均完成时间最小。本次实验不考虑任务的时效性问题, 即任务不会过期, 只要任务的感知数据能够传回任务分发者即视为任务完成。平均任务完成率体现的是用户接收任务集中的任务后, 数据回收情况。另外本文还从任务匹配率和任务分配总效用这 2 个方面对 3 种算法进行实验分析。任务匹配率定义为通过计算任务的位置与用户所在的位置的匹配程度。任务分配总效用定义为用户分配的任务集中任务回报值 and 用户任务完成率的乘积, 其中任务回报值为  $[0,100]^{[24]}$ , 即用户提交数据量与发布任务的说明中任务要求数据量的比值。

### 5.2.2 性能评价与分析

首先本文给出 3 种算法时间和空间复杂度的对比。HTA 分为 2 个步骤, 首先计算数据集中的中枢节点, 算法中需要计算每一个节点是否为中枢节点, 该步骤时间复杂度为  $O(n^3)$ ; 然后进行任务分配, 其时间复杂度为  $O(n^2)$ , 因此 HTA 的整体时间复杂度为  $O(n^3+n^2)$ , 空间复杂度为  $O(n)$ 。NTA 分为 3 个部分, 计算任务分配时间为  $O(n)$ ; 计算任务的候选匹配集为  $O(n^2)$ ; 更新分配结果集为  $O(n)$ 。因此 NTA 时间复杂度为  $O(n^2+n)$ , 空间复杂度也为  $O(n)$ 。通过以上分析可以看出, HTA 的时间复杂度最大, 本文算法和 NTA 一样都优于 HTA, 空间复杂度 3 个算法都相同。

本文从图 9 中可以观察到, 在 ParticipAct 数据集在不同任务集下, 本文的算法与 HTA、NTA 在任务平均完成时间方面的性能比较结果。总的来说, 由于  $t_1$  和  $t_3$  时长较短, 本文算法任务完成的平均时间明显优于对比算法, 完成任务所需的平均时间缩短了 42.2%。与此同时, 如图 10 所示, 与其他 2 个算法相比, 由于用户行为特征与任务特征的较高吻合, 本文算法实现了任务完成率的显著提高, 整体而言, 平均任务完成率提高了 53.1%。从

图 10 中可以观察到，任务集的平均工作量越小，本文的算法平均完成时间缩短的百分比越大，任务完成率所提高的百分比也越大。

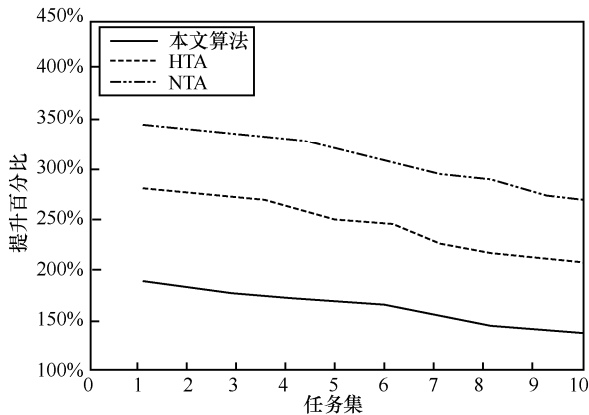


图 10 不同任务集的任务完成率提升百分比 (ParticipAct)

下面对 MDC Nokia 数据集进行性能分析。在所有任务集下，使用本文算法相比其他 2 种算法得到的任务平均完成时间平均缩短 48.1%，详细的任务平均完成时间性能比较如图 11 所示。同时任务完成率平均有 119.3% 的提高，详细的任务完成率性能比较如图 12 所示。

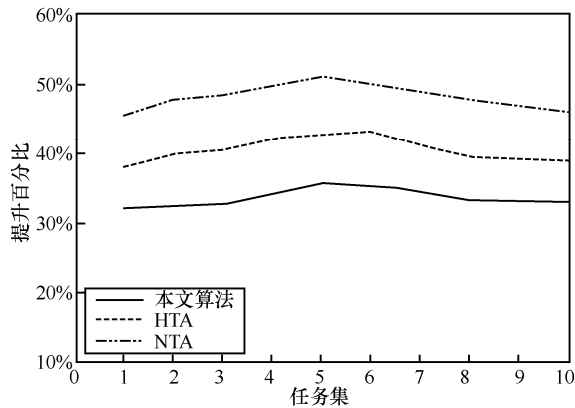


图 11 不同任务集的任务平均完成时间提升百分比 (MDC Nokia)

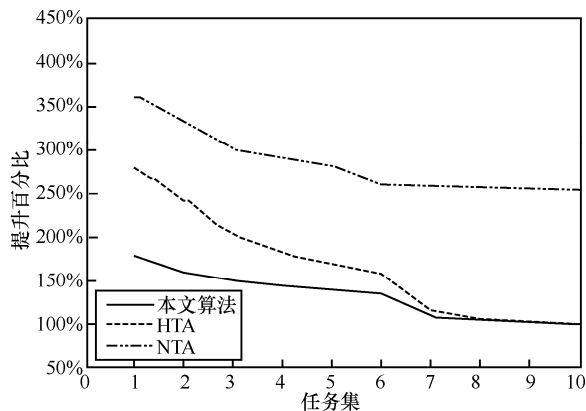


图 12 不同任务集的任务完成率提升百分比 (MDC Nokia)

从图 9 和图 11 还可以观察到，按任务集的工作量分组后，对不同工作量的任务集进行对比，结果显示本文算法对任务的平均完成时间所实现的改进百分比随任务集平均工作量的增加而有所降低，但缩短的绝对时间长度是增加的。通过分析得出，这一改进百分比变小的现象是由于任务工作量增加，任务分发阶段消耗的时间与任务结果收集阶段消耗的时间所占比重减小，任务执行阶段消耗的时间所占比重增加的原因造成的。因此，可以得出结论，本文所提算法对工作量越小的感知任务集较适用。

图 13 和图 14 分别为以 ParticipAct 为例，不同任务集下 3 种算法的任务匹配率和任务分配总效用的对比。由图 13 可知，本文算法由于将用户根据其行为模式划分成社区，并采用任务特征值与社区行为特征相匹配的方法分发任务，因此任务匹配度最高，且随着任务集中工作量的增多，匹配度变化不大。而 NTA 由于采用在线任务分发的方式，任务匹配度要高于 HTA，并且 2 种算法的任务匹配度随着任务集中工作量的增多，任务匹配度下降比较明显。由图 14 可知，随着任务集中任务工作量的增加，3 种算法的任务分配总效用值都随之降低了。本文算法的下降速度比较缓慢，总效用值明显优于对比算法。

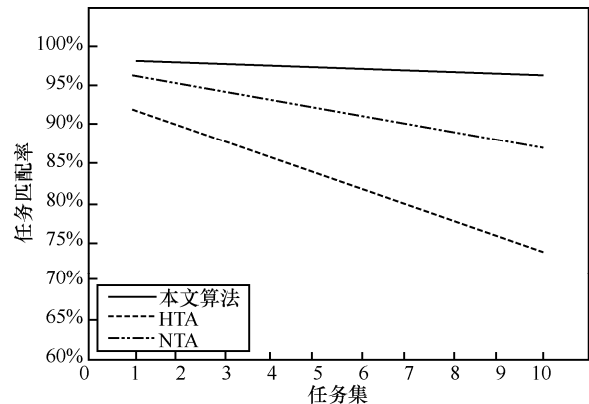


图 13 不同任务集的任务匹配率对比

根据实验结果分析发现，本文算法对任务平均完成时间性能的改进程度有差异，从数据集的具体特点分析得出：因为本文算法依赖于移动机会网络中各节点之间的稳定强关系，而作为任务分发者的节点遇到的其他节点在移动机会网络中的活跃程度各不相同，在社区中心节点与其他普通节点的相遇次数较多时，不能体现本文路由算法的优势；另

外, 某些节点作为社区中心节点时, 与其他节点的相遇次数过少或过于集中, 不能满足分配感知任务和接收感知任务结果的需要, 换句话说, 就是受限于采用的数据集。分析结果表明, 当移动节点规模越大时, 越能发挥本文算法在缩短任务分发时间消耗的优势。

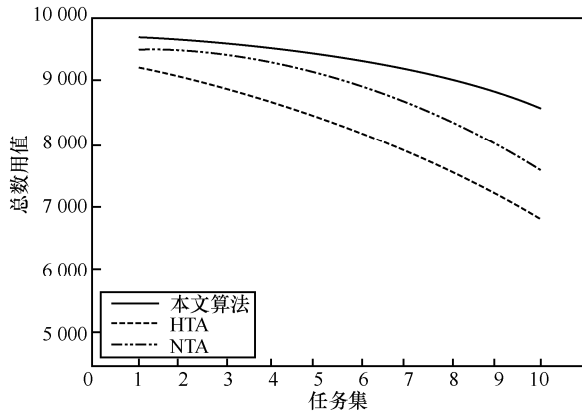


图 14 不同任务集的任务分配总效用值对比

## 6 结束语

在 MCS 的任务分发问题中, 需要海量的用户来完成复杂的感知任务。如何确定高效的任务分发与数据回收策略是 MCS 数据收集的核心问题之一。本文分析移动终端携带者人的社会属性和行为特点, 将社区的概念与任务分配策略相结合, 提出了基于社区的任务分发算法, 通过计算移动节点间的最小生成树、连接参量、社区融合度, 抽象和识别出节点的行为模式, 将节点划分成不同的社区, 并计算社区的中心锚点, 由锚点辅助分发和收集任务数据, 降低了任务在数据分发和收集上的时间消耗, 也提高了 MCS 任务用户群的匹配度。实验结果显示, 本文提出的算法与其他 2 个算法相比任务平均完成时间缩短了 45.1%, 任务平均完成率提高了 86.2%。进一步的实验分析表明, 尤其当任务数量多、用户分布不均、任务涉及的位置更多时, 本文算法更能体现其优势。考虑到不同节点的活跃度和完成任务的经验, 后续工作主要针对在保证完成任务的平均时间成本和完成率的同时, 如何保证感知数据完成质量符合量化要求。

## 参考文献:

[1] 吴垚, 曾菊儒, 彭辉, 等. 群智感知激励机制研究综述[J]. 软件学报, 2016, 27(8):2025-2047.

WU Y, ZENG J R, PENG H, et al. Survey on incentive mechanisms for crowd sensing[J]. Journal of Software, 2016, 27(8): 2025-2047.

[2] OMOKARO O, PAYTON J. FlySensing: a case for crowdsensing in the air[C]//IEEE International Conference on Pervasive Computing & Communication Workshops, 2014: 545-550.

[3] ZAPPATORE M, LONGO A, BOCHICCHIO M A, et al. A crowdsensing approach for mobile learning in acoustics and noise monitoring[C]//ACM Symposium on Applied Computing. 2016:219-224.

[4] HU S. SmartRoad: smartphone-based crowd sensing for traffic regulator detection and identification[J]. ACM Transactions on Sensor Networks, 2015, 11(4):1-27.

[5] WEI G, ZHANG B, CHENG L. Task assignment in mobile crowdsensing: present and future directions[J]. IEEE Network, 2018, PP (99):1-8.

[6] 安健, 彭振龙, 桂小林. 群智感知中基于公交系统的任务分发机制研究[J]. 计算机学报, 2019(2):295-308.

AN J, PENG Z L, GUI X L. Research on task distribution mechanism based on public transit system in crowd sensing[J]. Chinese Journal of Computers, 2019(2):295-308.

[7] 刘琰, 郭斌, 吴文乐, 等. 移动群智感知多任务参与者优选方法研究[J]. 计算机学报, 2017, 40(8):1872-1887.

LIU Y, GUO B, WU W L, et al. Multitask-oriented participant selection in mobile crowd sensing[J]. Chinese Journal of Computers, 2017, 40(8):1872-1887.

[8] XIAO M J, WU J, HUANG L S, et al. Online task assignment for crowdsensing in predictable mobile social networks[J]. IEEE Transactions on Mobile Computing, 2017, 16(8): 2306-2320.

[9] YU D H, ZHOU Z, WANG Y. Crowdsourcing software task assignment method for collaborative development[J]. IEEE Access, 2019, 10(7): 35743-35754.

[10] 徐哲, 李卓, 陈昕. 面向移动群智感知的多任务分发算法[J]. 计算机应用, 2017, 37(1): 18-23.

XU Z, LI Z, CHEN X. Multi-task assignment algorithm for mobile crowd sensing[J]. Journal of Computer Applications, 2017, 37(1): 18-23.

[11] 杨玉仁, 张书奎, 龙浩, 等. 群智感知中基于社交属性及有效用户计算的任务分发机制[J]. 计算机应用研究, 2019(5):1493-1499.

YANG Y R, ZHANG S K, LONG H, et al. Task distribution mechanism based on social attribute and effective user calculation in crowd sensing[J]. Application Research of Computers, 2019(5):1493-1499.

[12] WANG L, YU Z, ZHANG D, et al. Heterogeneous multi-task assignment in mobile crowdsensing using spatiotemporal correlation[J]. IEEE Transactions on Mobile Computing, 2018, 18(1):84-97.

[13] 张君涛, 赵智慧, 周四望. 矢量任务地图: 群智感知任务渐进式分发方法[J]. 计算机学报, 2017, 40(8):1946-1960.

ZHANG J T, ZHAO Z H, ZHOU S W. Vector task map: progressive task allocation in crowd-sensing[J]. Chinese Journal of Computers, 2017, 40(8):1946-1960.

[14] SAOUD B, MOUSSAOUI A. Community detection in networks based on minimum spanning tree and modularity[J]. Physica A: Statistical Mechanics & Its Applications, 2016(460):230-234.

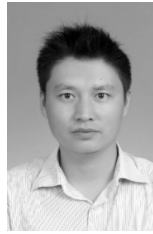
[15] 吕少卿, 张玉清, 刘东航, 等. 在线社交网络中 Spam 相册检测方法[J]. 通信学报, 2016, 37(9):82-91.

LYU S Q, ZHANG Y Q, LIU D H, et al. Detecting Spam albums in

online social network[J]. Journal on Communications, 2016, 37(9): 75-91.

- [16] BULUT E, SZYMANSKI B K. Exploiting friendship relations for efficient routing in mobile social networks[J]. IEEE Transactions on Parallel & Distributed Systems, 2012, 23(23):2254-2265.
- [17] XIONG P, ZHU T, NIU W, et al. A differentially private algorithm for location data release[J]. Knowledge and Information Systems, 2016, 47(3):647-669.
- [18] REDDY S, ESTRIN D, SRIVASTAVA M. Recruitment framework for participatory sensing data collections[C]// Pervasive Computing, International Conference.ACM, 2010:138-155.
- [19] SUN H, HUANG J, ZHANG X, et al. IncOrder: incremental density-based community detection in dynamic networks[J]. Knowledge-Based Systems, 2014(72):1-12.
- [20] HOSSEN M S, AHMED M T, RAHIM M S. Effects of buffer size and mobility models on the optimization of number of message copies for multi-copy routing protocols in scalable delay-tolerant networks[C]// International Conference on Innovations in Science. IEEE,2017:1-4.
- [21] 张兴盛, 余敦辉, 张万山, 等. 时空众包环境下时效均衡的在线任务分配算法[J]. 计算机应用, 2019, 39(5):1357-1363.  
ZHANG X S, YU D H, ZHANG W S, et al. utility balanced online task assignment algorithm under spatial crowdsourcing environment[J]. Journal of Computer Applications, 2019, 39(5):1357-1363.
- [22] PAOLO B, ANTONIO C, LUCA F, et al. Scalable and cost-effective assignment of mobile crowdsensing tasks based on profiling trends and prediction: the participact living lab experience[J]. Sensors, 2015, 15(8):18613-18640.
- [23] LAURILA J K, GATICA-PEREZ D, AAD I, et al. The mobile data challenge: big data for mobile computing research[C]// Mobile Data Challenge by Nokia Workshop. IEEE,2012:1-8.
- [24] 刘辉, 李盛恩. 时空众包环境下基于统计预测的自适应阈值算法[J]. 计算机应用, 2018, 38(2): 415-420.  
LIU H, LI S G. Adaptive threshold algorithm based on statistical prediction under spatial crowdsourcing environment[J]. Journal of Computer Applications, 2018, 38(2): 415-420.

### [作者简介]



龙浩（1984-），男，湖南长沙人，苏州大学博士生、讲师，主要研究方向为群智感知计算、数据挖掘、分布式计算。



张书奎（1966-），男，内蒙古呼和浩特人，博士，苏州大学教授、博士生导师，主要研究方向为物联网、无线传感器网络、信息安全、群体感知计算、智能信息处理等。



张洋（1989-），男，江苏苏州人，苏州大学硕士生，主要研究方向为物联网、信息安全、群体感知计算、智能信息处理等。



张力（1983-），男，江苏镇江人，苏州大学博士生，主要研究方向为群智感知、数据挖掘、并行计算。