

OTAP: 基于预测的机会群智感知多任务在线分配算法

李卓^{1,2}, 徐哲², 陈昕², 李淑琴²

(1.北京信息科技大学 网络文化与数字传播北京市重点实验室, 北京 100101; 2.北京信息科技大学 计算机学院, 北京 100101)

摘要:机会群智感知网络中,不同节点间的相遇间隔各异,任务由不同节点执行时的时间成本有较大差异性。为最小化任务平均完成时间,设计并实现了一种基于预测的多任务在线分配算法(online multi-task assignment based on prediction, OTAP)。基于真实移动轨迹数据集,分析了节点间相遇间隔分布,设计了节点相遇规律发现子算法;利用对节点间的相遇间隔的预测,每次给执行节点分配在与任务分发者下次相遇间隔内能完成的最大任务量。针对4个不同的真实移动轨迹数据集,利用ONE模拟器,对OTAP算法性能进行了验证与分析。结果显示,相比于已有的NTA算法,OTAP在4个不同数据集中平均任务完成时间分别缩短了50.49%、45.34%、32.71%、32.23%,任务完成率在其中两个移动轨迹数据集中也有所提高。

关键词:机会群智感知;多任务分配;在线算法

中图分类号:TP393.01

文献标志码:A

文章编号:2096-3246(2018)05-0176-07

OTAP: Online Multi-task Assignment Algorithm with Prediction for Opportunistic Crowd Sensing

LI Zhuo^{1,2}, XU Zhe², CHEN Xin², LI Shuqin²

(1.Beijing Key Lab. of Internet Culture and Digital Dissemination Research, Beijing Info. Sci. & Technol. Univ., Beijing 100101, China;
2.School of Computer Sci., Beijing Info. Sci. & Technol. Univ., Beijing 100101, China)

Abstract: Inter-contact time between different node pairs in opportunistic crowd sensing networks is different, which results in different makespan of tasks when the tasks are taken by different nodes. In order to minimize the average makespan of tasks, an online multi-task assignment algorithm based on prediction (OTAP) was proposed and implemented. Based on the real mobility trace datasets, the distribution of the inter-contact time between nodes was analyzed and a subroutine for discovering the contact law between nodes was developed. Relying on the prediction for inter-contact time between nodes, each time the node responsible for task execution was assigned the most amount of task that could be accomplished before the next contact with the task allocator. Using four different real mobility trace datasets, the performance of OTAP was verified and analyzed by ONE simulator. Experiment results showed that OTAP outperformed the existing algorithm NTA on the average makespan of tasks by 50.49%, 45.34%, 32.71% and 32.23% individually on the four different datasets. Completion ratio of tasks was also improved on two mobility datasets.

Key words: opportunistic crowd sensing; multi-task assignment; online algorithm

传统物联网受限于成本、灵活性和管理上的问题,大规模部署受到挑战。随着配备了多种嵌入式传感器的移动智能设备得到普及,出现了一种新的感知模式——群智感知^[1]。不同于由专用固定传感器组成的无线传感器网络^[2],群智感知设备可利用其移动性以较低成本实现大规模、大范围感知任务。因此,

群智感知受到广泛关注和研究^[3-6]。移动终端基本都具备短距离无线通信技术,如蓝牙(Bluetooth)或Wi-Fi等,为避免传统移动蜂窝网络信号强度覆盖不均匀及在特殊场景中不可用的问题,可利用短距离通信技术进行数据机会传输,这种特殊群智感知模式称为机会群智感知^[7]。

收稿日期:2017-09-10

基金项目:国家自然科学基金资助项目(61502040;61370065);北京市属高校高水平教师队伍支持计划青年拔尖人才培养计划资助项目(CIT&TCD201804055);网络文化与数字传播北京市重点实验室资助项目(ICDDXN001);北京信息科技大学“勤信英才”培养计划资助项目

作者简介:李卓(1983—),男,副教授,博士。研究方向:移动无线网络、分布式计算。E-mail: lizhuo@bistu.edu.cn

网络出版时间:2018-08-31 20:06:11 网络出版地址: <http://kns.cnki.net/kcms/detail/51.1773.tb.20180831.2006.002.html>

在机会群智感知网络中,由于任务的分配与完成都依靠着两节点在相遇时建立的通信链接进行数据通信,所以节点间的相遇时间对任务分配的决策有很大影响。节点的移动具有随机性,任务分发节点在分发任务时,与执行节点之间没有持续连接的通信链路,加之节点的具体信息难以准确获取,所以传统并行机调度^[8]领域的研究无法直接应用于机会群智感知网络。

为减少机会网络中完成感知任务的总时间,国内外已开展了多项研究。移动感知任务可能与任务执行节点的位置相关,针对这种情况,He等^[3]认为,制定位置相关的任务分配解决方案时需要考虑感知任务的地理特征和移动用户的移动性,并提出了两个有效的近似算法,即LRBA和DLBA,但是,文中并未考虑任务的属性,比如任务的质量要求,这在分配期间是必不可少的。Tang等^[9]研究了车载移动网络中的数据收集问题,为了解决智能设备中的存储有限问题,提出了选择策略的优先级概念,并提出了一种基于模拟退火的优先级分配算法,但是文中并没有考虑电池容量的限制。Maia等^[10]在求解优化任务完成时间的问题时,加入了电池容量的限制。Shi等^[11]提出了Water Filling算法用于任务分配,以最小化最长任务执行时间。该算法即不考虑移动节点的实时移动性,也不考虑移动节点的历史移动轨迹。Yao等^[12]以最小化最长任务完成时间为目标,提出了相遇概率敏感的离线和在线任务分配算法。Xiao等^[13]以最小化平均任务完成时间为优化目标,提出了NTA算法,并证明了其近似比与平均任务工作量和节点期望相遇时间相关,但要求分配节点掌握所有节点的信息。徐哲等^[14]针对最小化任务平均完成时间的多任务分发问题,提出一种基于中枢节点的多任务分发算法HTA,与文献^[13]相同,该算法要求任务请求者掌握网络全局信息,即所有节点间的相遇概率,但是在实际应用中较难满足该条件。文献^[13]中,针对移动节点间的相遇时间间隔采用了指数分布的假设,通过分析真实数据集可发现,指数分布描述的相遇间隔规律与实际情况之间仍有一定偏差,如人类的移动存在间隙性、人类的夜间活动较少甚至停止等情况下难以利用指数分布反映出来。综上所述,文献^[11,13-15]等研究中需要掌握整个节点网络的全局信息,这对算法的应用环境提出了较高的要求,限制了算法的实用性。

作者利用局部信息,研究一种面向最小化任务平均完成时间、支持任务动态增加的在线分配算法。经过对真实数据集的分析发现,在机会群智感知网络中,虽然节点的移动具有自主性和随机性,但节点间在不同时刻的相遇仍然有一定的内在联系。基于

节点的历史相遇记录可以寻找相遇间隔分布规律,进而对节点的相遇时间进行预测,由此,设计一种基于预测的在线多任务分配(online multi-task assignment based on prediction, OTAP)算法。OTAP算法的核心思想是每次给执行节点分配在与任务分发者下次相遇间隔内能完成的最大任务量,以降低感知节点在完成感知任务后及与任务发布者相遇提交任务结果之间的等待时间。基于unicalsocialblue^[16]、cambridge_city^[17]、infocom2005^[17]、intel^[17]等4个不同数据集,利用ONE机会网络模拟器^[18]进行充分实验,验证提出的节点相遇预测模型的准确性,该算法有效降低了机会群智感知网络中平均任务完成时间。

1 问题描述

1.1 网络模型

假设机会群智感知网络由 n 个节点组成,节点用 $v_i(i=0,1,2,3,\dots,n-1)$ 表示。每个节点均携带了具有一定计算、存储、通信能力的智能感知设备。节点间只通过短距离无线通信技术(如Wi-Fi、蓝牙等)建立连接并传输数据。假设节点间的数据传输可在一次相遇机会中完成。

1.2 任务模型

假设上述网络中有一个任务分发者,用 v_0 表示。它有 m 个相互独立的感知任务需要其他 $n-1$ 个节点中的部分或全部节点执行。任务分配方案由 v_0 确定。 v_0 在与其它节点 v_i 相遇时,将根据确定的任务分配策略,将若干任务分发给 v_i 。 v_i 花费一定量的时间完成被分配的任务。在完成感知任务后, v_i 将一直等到与 v_0 再次相遇时,再将所得到的任务结果返回给 v_0 。使用 $J_i(1 \leq i \leq n-1)$ 表示节点 v_i 执行的任务集合, $J_i = \{j_1^i, j_2^i, \dots, j_k^i\}$, v_i 按照 j_1^i, j_2^i 一直到 j_k^i 的顺序依次执行 J_i 中的任务。

约束条件:1)每个时刻每个节点只能执行一个任务,每个任务只能被一个节点执行,且任务一旦开始执行不能被中断;2)不考虑任务的优先权。

定义(任务完成时间makespan) 一个感知任务 j_k 的完成时间为 $M(j_k) = t_{k1} + t_{k2} + t_{k3}$,其中, t_{k1} 为任务分发者开始分发任务 j_k 到节点 v_i 接收到任务 j_k 的时长, t_{k2} 为任务 j_k 从节点 v_i 接收到执行完成的时长, t_{k3} 为节点 v_i 将任务结果传递回任务分发者的时长。

1.3 问题描述

作者研究的问题是如何确定一种任务分配策略,使感知任务平均完成时间(average makespan, AM)最小化。用 $J = \{j_1, j_2, \dots, j_m\}$ 表示 m 个独立的感知任务, $\Pi = (J_1, J_2, \dots, J_{n-1})$ 表示一种任务分配策略。当且仅当 $J_i \cap J_j = \emptyset, i \neq j$ 且 $\sum_{i=1}^{n-1} \cup J_i = J$, v_0 将 J_i 中的所有

任务全部分配给用户 v_i ; 如果 J_i 为 \emptyset , 任务分发者 v_0 不分配任何任务给用户 v_i 。

问题 (最小化平均完成时间的在线多任务分配问题) 在机会群智感知网络中, 给定一个节点集合 $V = (v_1, v_2, \dots, v_n)$ 、一个感知任务集合 $J = (j_1, j_2, \dots, j_m)$, 每个感知任务 j_j 有相应的任务工作量 p_j 和发布时间 r_j 。在时间 r_j 之前, 任务 j_j 并不存在, 任务分发者无法获知其任何信息。则最小化平均完成时间的在线多任务分配问题就是确定一个任务分配策略 $\Pi = \{J_1, J_2, \dots, J_n\}$, 使任务平均完成时间最小, 即

$$\min AM(\Pi) = \frac{1}{m} \sum_{i=1}^m M(j_i) \quad (1)$$

式中, $M(j_i)$ 表示任务 j_i 的完成时间, $AM(\Pi)$ 表示任务分配策略 Π 下所有任务的平均完成时间。

定理 1 在清楚所有节点彼此相遇时间的情况下, 问题 1 为 NP-完全问题。

证明: 针对单机的最小化平均完成时间的在线多任务分配问题已被证明为 NP-完全问题^[19-20]。而单机情况为问题 1 的特例, 故问题 1 为 NP-完全问题。

证毕

2 基于预测的在线任务分配算法

2.1 节点移动模型

以往对移动机会网络中任务分配问题的研究均假设节点相遇间隔时间服从指数分布^[20], 基于对多个真实数据集中记录的分析, 发现这些数据集中普遍存在一个规律是: 节点间的相遇间隔往往存在周期性, 即两个节点在长度为 T_1 的一段时间内频繁相遇, 在之后一段长度为 T_2 的时间内又几乎不相遇, 如此反复。若将上述过程定义为一个相遇周期, 则周期为 $T = T_1 + T_2$ 。以 cambridge_city 数据集中 2 号节点为例, 图 1 展示了其与其他节点的相遇情况统计, 横轴为时间, 纵轴为相遇节点的序号。由图 1 可以看出 2 号节点与其他节点间相遇存在周期性。

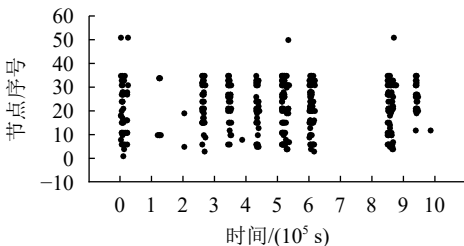


图 1 cambridge_city 数据集中的节点 2 与其他节点的相遇记录

Fig. 1 Encounter records between node 2 and other nodes in cambridge_city dataset

2.2 节点相遇规律发现算法

根据第 2.1 节中对节点移动规律的分析, 可将两

个节点的相遇序列划分为交替的持续相遇阶段和长间隔阶段, 其中, 持续相遇阶段由短持续时长 t_s^{θ} 所在序列组成, 长间隔阶段由长相遇间隔 t_l^{θ} 所在序列组成。根据历史相遇记录, 判断当前相遇时刻是否处于持续相遇阶段, 预测未来相遇时间间隔。短持续时长 t_s^{θ} 和长相遇间隔 t_l^{θ} 的定义如下:

对于两个节点的一段相遇时间序列 $\{T_1, T_2, \dots, T_n\}$, 若存在一段最长子序列 $\{T_i, T_{i+1}, \dots, T_j\}$, 满足 $\forall i \leq k < j, T_{k+1} - T_k < \theta$ 且 $T_{j+1} - T_j \geq \theta$, 则称 $T_j - T_i$ 为 θ 参数的短持续时长, 记为 t_s^{θ} ; 称 $T_{j+1} - T_j$ 为 θ 参数的长相遇间隔, 记为 t_l^{θ} 。

节点相遇规律发现算法的基本思路是:

1) 遍历所有相遇记录, 将两次相邻相遇间隔 $T' < \theta$ 的记录合并, 并将合并后的新记录的持续时间增加 T' 。

2) 将新生成的记录序列的平均时间间隔记为长相遇间隔 t_l^{θ} , 持续时间的平均值为短持续时长 t_s^{θ} 。

具体实现步骤见算法 1。

算法 1 节点相遇规律发现算法 (getHistoryRule)

输入: 两节点间相遇记录序列 R 、阈值 θ 。

输出: 两节点相遇的长时间间隔 t_l^{θ} 、短持续时长 t_s^{θ} 。

1. 遍历 R , 若相邻两次相遇记录 r_1 和 r_2 的时间间隔小于 θ , 则对 r_2 做标记;
2. 根据未被标记的记录将 R 分割为若干子序列 a_1, a_2, \dots, a_x , 未被标记的记录作为首元素;
3. 统计所有子序列 a_1, a_2, \dots, a_x 中的第 1 条记录和最后一条记录之间的时间间隔, 并求平均值记为 t_s^{θ} , 若子序列长度为 1, 则时间间隔为零;
4. 提取每个子序列 a_i 的第 1 条记录组成新的子序列 b , 并统计这些记录间的时间间隔, 求其平均值记为 t_l^{θ} , 若子序列 b 的长度为 1, 则 t_l^{θ} 等于无穷大;
5. 返回 t_l^{θ} 、 t_s^{θ} 。

2.3 任务分配子算法

要使平均任务完成时间最小, 即所有任务的完成时间的总和最小。根据第 1.2 节对任务完成时间的定义, 由于任务的工作量固定不变, 即执行阶段所消耗的时间是定值。因此, 应使分发任务和返回任务结果两个阶段的时间最小。

制定任务分配策略: 预测节点间的相遇间隔, 每次给执行节点分配在与任务分发者下次相遇间隔内能完成的最大任务量。任务分配子算法的基本思路是:

1) 根据相遇历史记录, 通过算法 1 计算节点间相遇规律, 得到短持续时长 t_s^{θ} 和长相遇间隔 t_l^{θ} 。

2) 获取最近一次长相遇间隔后的第 1 次相遇时刻 t' 和空闲时刻 $idle_i$ 。

3) 根据 t' 和 t_s^{θ} 预测本次持续相遇阶段结束时间

$t_e = t' + t_s^0$ 。

4) 如果 $t_e - t$ 大于 $t_s^0 \times \omega$, 将预期分配任务量 Δ 设置为 $t_e - idle_i$; 否则, $\Delta = t_e + t_s^0 - idle_i$ 。其中: t 为当前时刻; ω 为分配因子, 控制分配持续相遇阶段所执行任务的时间窗口在持续相遇阶段所占比例。

5) 将待分配任务集中的任务依次分配给节点, 直到本次已分配任务的总任务量之和大于 Δ 为止。

该算法具体步骤见算法2。算法2在获取待分配任务集时, 并未要求获取所有的任务信息; 同时, 利用已有节点间相遇历史对下一次相遇间隔进行预测, 因此能够适应任务在线分配的要求。

算法2 任务分配子算法(assignTasks)

输入: 待分配任务集 $J = (j_1, j_2, \dots, j_m)$ 、相遇节点 v_i 、当前时刻 t 、相遇节点 v_i 的空闲时刻 $idle_i$ 、最近一次长时间间隔后的第1次相遇时刻 t' 、长相遇间隔 t_s^0 、短持续时长 t_s^0 、分配因子 $\omega (0 < \omega)$ 。

输出: 相遇节点分配到的任务集 Π_i 。

1. 初始化: 将结果置为 \emptyset , $\Delta = 0$

2. $t_e = t' + t_s^0$;

3. if $t' == -1$ 或 $t_e - t > t_s^0 \times \omega$ do

4. $\Delta = t' + t_s^0 - idle_i$;

5. else if $t_e - t \leq t_s^0 \times \omega$ do

6. $\Delta = t' + t_s^0 - idle_i$;

7. if $\Delta > 0$ do

8. 将 J 中的任务按任务量从小到大的顺序分配给节点 v_i , 直到新分配任务的总任务量之和大于 Δ , 或已分配完成所有任务;

9. 将新分配的任务加入结果中;

10. return 结果。

2.4 整体任务分配框架

第3.2、3.3节的算法是作者所设计算法的核心内容, 下面介绍算法的整体框架, 详见算法3。算法3使用算法1进行节点相遇规律计算得到两节点的长相遇间隔和短持续时长用于预测节点相遇时间, 使用算法2作为任务分配算法计算得到分配策略。

由于分配到每个节点的任务数一般为多个, 所以还涉及到单机任务调度问题, 依据最小化平均完成时间的求解目标可知节点应按照任务工作量从小到大的顺序执行。

算法3 在线多任务分配算法(OTAP)

输入: 待分配任务集 $J = (j_1, j_2, \dots, j_m)$ 、候选节点集 $V = (v_1, v_2, \dots, v_n)$ 、候选节点与任务分发节点的相遇历史记录 H 、阈值 θ 、分配因子 $\omega (0 < \omega)$ 。

输出: 分配策略 Π_i 。

1. 将 H 按 V 中节点分组, 并按时间先后排序得 $R = (r_1, r_2, \dots, r_k)$;

2. for $i \in [1, n-1]$ do

3. $J_i = \emptyset$;

4. $idle_i = 0$;

5. $\tau_i = 0$; /*记录 v_0 与 v_i 最后一次相遇的时刻*/

6. $\Pi = \{J_1, J_2, \dots, J_n\}$;

7. for $r_i \in R$ do

8. $(t_{s,i}^0, t_{s,i}^0) = \text{getHistoryRule}(r_i, \theta)$; /*调用算法1*/

9. 对 J 中任务按任务量从小到大排序得有序数组 J' ;

10. While 每次与某节点 v_i 相遇 and 任务未分配完毕 do

11. 令 t 为当前时刻;

12. if $idle_j < t$ do

13. $idle_j = t$;

14. if $\tau_j == 0$ 或 $t - \tau_j > t_{s,i}^0$ do

15. $\tau_j = t$;

16. $J'_j = \text{assignTasks}(J', v_j, t, idle_j, \tau_j, t_{s,i}^0, t_{s,i}^0, \omega)$; /*调用算法2*/

17. 将 J'_j 中的全部任务分配给 v_j 执行, $J' = J' - J'_j$, $J_i + = J'_j$;

18. $idle_j + = \text{sum}(J'_j)$;

19. return Π 。

3 算法分析

对算法各部分的复杂度和可行性进行分析。

定理2 算法3的复杂度为 $O(nk + m \log m)$, 其中, n 为网络中节点个数, k 为相遇记录个数, m 为待分配任务个数。

证明: 首先对算法1的复杂度进行证明。算法1将相遇记录序列划分为两个子序列, 再分别遍历每个子序列作加减操作, 最多做 $2k$ 次基本操作, 因此复杂度为 $O(k)$ 。

算法2中, 对于7~8行的语句, 最坏情况下的执行次数为 m 次, 因此算法2的复杂度为 $O(m)$ 。

算法3中, 首先, 对 $n-1$ 个节点调用 $n-1$ 次算法1, 因此该部分的复杂度为 $O(nk)$; 接着, 对所有任务进行排序, 设任务个数为 m , 则排序任务的复杂度为 $O(m \log m)$; 然后, 调用若干次算法2, 直至所有任务分配完毕, 由于最多分配 m 个任务, 因此该部分的复杂度为 $O(m)$; 所以, 算法3的复杂度为 $O(nk + m \log m)$ 。
证毕

算法3在对节点 v_i 进行任务分配时只用到节点 v_i 自身的历史相遇记录, 不涉及其他节点, 即不需要了解全局信息。由于算法2使用渐进式分配策略, 并非一次性将任务分配给节点 v_i , 算法3中的待分配任务集可以是动态增加的, 因此, 算法3具有实时在线分配的能力。与离线多任务分配算法相比, 算法3的竞争比是由节点间相遇间隔的预测准确度决定的。

4 性能评估

4.1 实验环境

采用ONE (opportunistic networking environment) 机会网络模拟器^[18], 使用Crawdad网站提供的4个移动轨迹数据集cambridge_city、infocom2005、intel和uncialsocialblue对OTAP算法的性能进行评估。两个对比算法为: 作者实现的NTA算法, 及依据文献^[13]给出的计算方法得到了最优任务分配策略, 前提是将节点间未来的相遇时刻作为已知条件。选择任务平均完成时间、任务完成率作为性能评价指标。任务完成率定义为是 $\frac{|J_r|}{|J|}$, 其中, $|J_r|$ 表示截止到移动轨迹数据集的最后一条相遇记录时任务分发者成功接收到执行结果的任务数量, $|J|$ 表示待分配任务的总数。

因为OTAP、NTA算法均需要根据一定数量的历史记录对节点间的相遇规律进行分析, 因此实验中需要划分数据集为训练集和测试集。训练集用于算法初始化阶段的规律分析, 测试集作为任务分发过程中节点间的相遇记录。数据集的详细介绍及参数配置见表1。此外, 使用的感知任务集为通过程序以均匀分布的统计特性随机生成, 任务集规模为50、100、150、200、250、300时, 最小工作量为100 s, 最大工作量为2 000 s。

根据对数据集中各节点的相遇间隔进行采样统计分析后, 可知所有节点在相遇时间间隔分布上以小于1h和大于等于2 h为主要分布点; 并且根据对部分数据的初步试验, 发现 ω 设置为0.75时, 算法在各

表 1 移动轨迹数据集配置

Tab. 1 Statistics of mobile trace datasets

数据集	场景	节点数	时间长度/s	总记录数	请求者数
uncialsocialblue	校园	30	709 597	21 920	1 579
cambridge_city	校园	51	987 529	10 873	529
infocom2005	会议	41	254 150	22 459	922
intel	室内	9	359 190	1 364	241

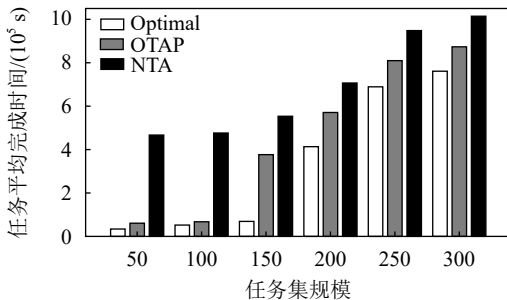
数据集的性能最稳定。因此, 将算法1中参数 θ 设置为1 h, 算法2中参数 ω 设置为0.75。

4.2 实验结果

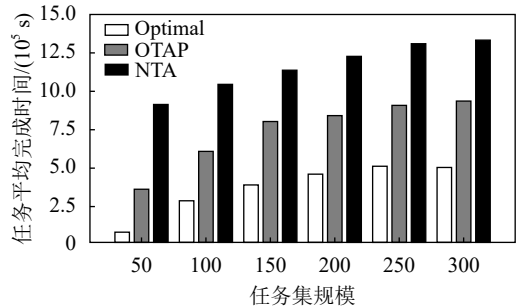
首先, 评估算法的任务平均完成时间。图2对比了在4种不同移动轨迹数据集中, 采用OTAP、NTA算法以及最优策略对不同配置的任务集进行任务分配时所得到的任务平均完成时间。

从图2可以看出, OTAP算法在4个不同移动轨迹数据集中分配各任务集时所得到的平均完成时间均低于NTA算法, 但比最优任务分配策略高。在4个不同移动轨迹数据集uncialsocialblue、cambridge_city、infocom2005、intel中, OTAP算法在不同任务集规模中所得到的平均任务完成时间比NTA算法平均缩短了50.49%、45.34%、32.71%、32.23%。

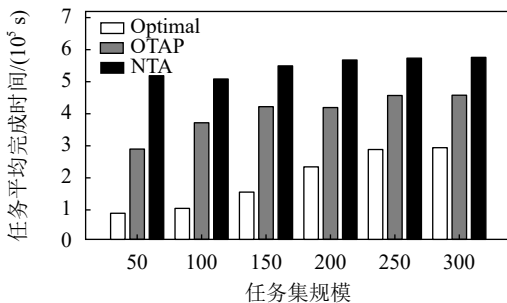
评估算法性能的另一个指标是任务完成率。图3对比了使用4个不同移动轨迹数据集进行实验时, OTAP、NTA算法、最优任务分配策略在任务完成数与任务集规模的关系 (即任务完成率)。如图3(a)所示, OTAP和NTA算法所得任务完成率均达到了100%。



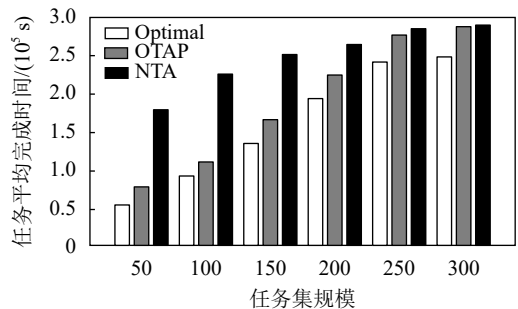
(a) 移动轨迹数据集 uncialsocialblue



(b) 移动轨迹数据集 cambridge_city



(c) 移动轨迹数据集 infocom2005



(d) 移动轨迹数据集 intel

图 2 OTAP与NTA、最优解在平均任务完成时间上的性能对比

Fig. 2 Performance comparisons among OTAP, NTA and the Optimal solution on the average makespan

如图3(b)和(d)所示,在cambridge_city和intel数据集中,OTAP算法的任务完成率高于NTA算法,平均分别高了16.55%、4.57%。如图3(c)所示,在infocom2005数据集中,除任务集规模为50时以外,OTAP的任务

完成率均低于NTA算法,平均低了2.75%。其具体原因在于:infocom2005数据集的活动时间较短;OTAP算法在分析和利用节点移动规律时,没有充分发挥出其优势。

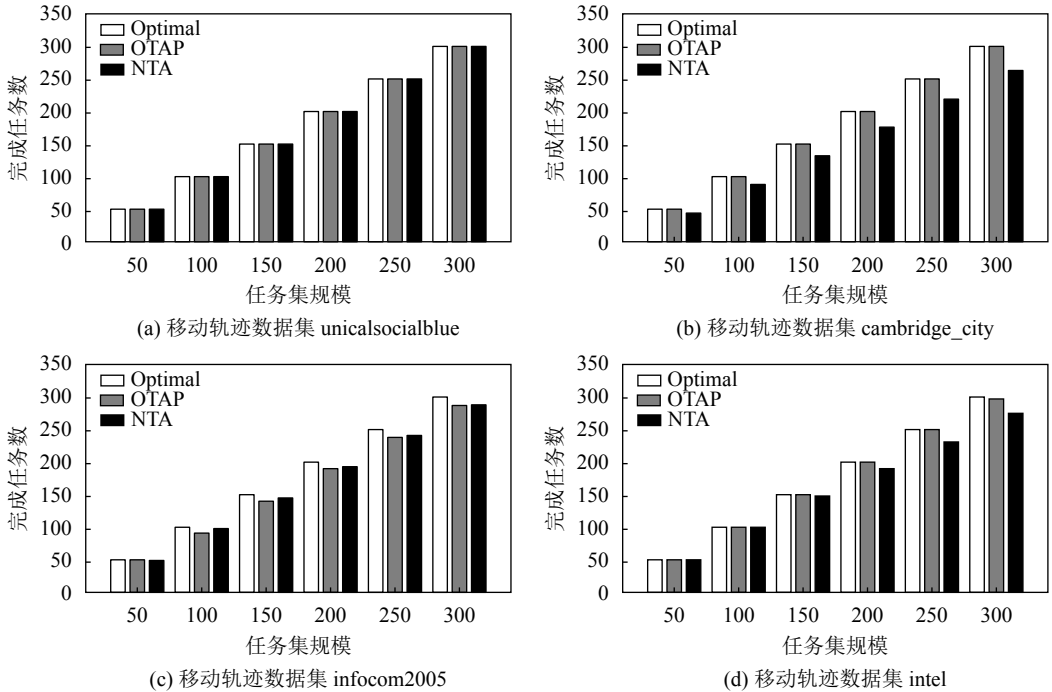


图3 OTAP与NTA、最优解在最终任务完成数上的性能对比

Fig. 3 Performance comparisons among OTAP, NTA and the Optimal solution on the number of completed tasks

图4对比了4个不同数据集中,任务集规模为300时,OTAP、NTA算法以及最优任务分配策略的任

务完成率与时间的关系。从图4中可以看出OTAP算法比NTA算法的任务策略完成更快。

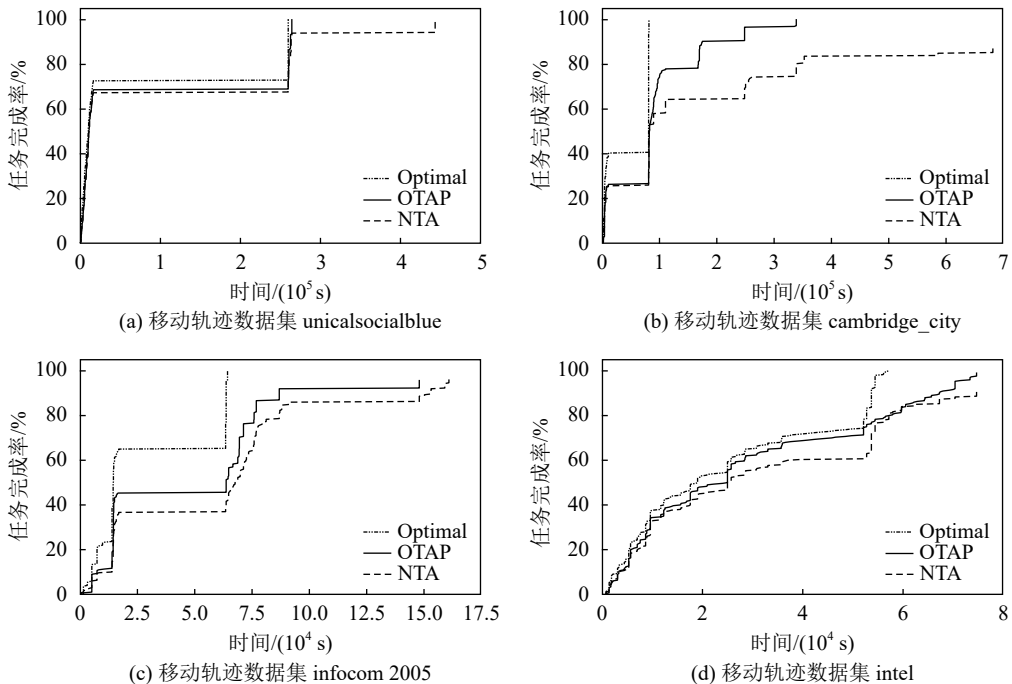


图4 OTAP与NTA、最优解在时间与任务完成率关系上的对比

Fig. 4 Comparisons among OTAP, NTA and the Optimal solution on the relationship between time and task completion rate

5 结 论

针对机会群智感知网络中的多任务在线分配问题开展研究。基于真实数据集分析了移动节点的相遇规律,将节点间相遇过程分为若干周期,每个周期包含持续相遇阶段和长间隔阶段。据此规律设计了一种基于预测的在线任务分配(OTAP)算法,以缩短任务平均完成时间。OTAP算法的核心思想是对节点间的相遇间隔进行预测,每次给执行节点分配在与任务分发者下次相遇间隔内能完成的最大任务量。在4个不同数据集上进行实验,结果显示本文提出的OTA算法比NTA算法缩短了至少32.23%的任务平均完成时间,同时,任务完成率在其中两个移动轨迹数据集中也有所提高。在未来的研究工作中,将基于不同感知任务的相关性分析、感知任务分布与用户移动性的时空关联分析等对机会群智感知任务分配机制做进一步优化。

参考文献:

- [1] Zhao Dong, Ma Huadong. Development and challenges of crowd sensing networks[J]. *Information and Communications Technologies*, 2014(5):66–70. [赵东, 马华东. 群智感知网络的发展及挑战[J]. *信息技术*, 2014(5):66–70.]
- [2] Li Jianzhong, Gao Hong. Survey on sensor network research[J]. *Journal of Computer Research and Development*, 2008,45(1):1–15. [李建中, 高宏. 无线传感器网络的研究进展[J]. *计算机研究与发展*, 2008,45(1):1–15.]
- [3] He Shibo, Shin D H, Zhang Junshan, et al. Near-optimal allocation algorithms for location-dependent tasks in crowdsensing[J]. *IEEE Transactions on Vehicular Technology*, 2017, 66(4):3392–3405.
- [4] Wu Yao, Zeng Juru, Peng Hui, et al. Survey on incentive mechanisms for crowd sensing[J]. *Journal of Software*, 2016,27(8):2025–2047. [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.]
- [5] Zeng Juru, Chen Hong, Peng Hui, et al. Privacy preservation in mobile participatory sensing[J]. *Chinese Journal of Computers*, 2016,39(3):595–614. [曾菊儒, 陈红, 彭辉, 等. 参与式感知隐私保护技术[J]. *计算机学报*, 2016,39(3):595–614.]
- [6] Zhao Dong, Ma Huadong, Li Qi, et al. A unified delay analysis framework for opportunistic data collection[J]. *Wireless Networks*, 2018,24(4):1313–1325.
- [7] Xiong Yongping, Liu Wei, Liu Zhuohua. Key technologies of opportunistic crowd sensing network[J]. *ZTE Technology Journal*, 2015,21(6):19–22. [熊永平, 刘伟, 刘卓华. 机会群智感知网络关键技术[J]. *中兴通讯技术*, 2015,21(6):19–22.]
- [8] Cheng T C E, Sin C C S. A state-of-the-art review of parallel-machine scheduling research[J]. *European Journal of Op-*

erational Research, 1990,47(3):271–292.

- [9] Tang Zhipeng, Liu Anfeng, Huang Changqin. Social-aware data collection scheme through opportunistic communication in vehicular mobile networks[J]. *IEEE Access*, 2016,4:6480–6502.
- [10] Maia S L F, Silva É R, Guardieiro P R. A new optimization strategy proposal for multi-copy forwarding in energy constrained dtns[J]. *IEEE Communications Letters*, 2014, 18(9):1623–1626.
- [11] Shi C, Lakafosis V, Ammar M H, et al. Serendipity: enabling remote computing among intermittently connected mobile devices[C]//Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing. *New York:ACM*, 2012:145–154.
- [12] Yao Hong, Xiong Muzhong, Liu Chao, et al. Encounter probability aware task assignment in mobile crowdsensing[J]. *Mobile Networks and Applications*, 2017,22(2):275–286.
- [13] Xiao Minjun, Wu Jie, Huang Liusheng, et al. Multi-task assignment for crowdsensing in mobile social networks[C]//Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM). *Hong Kong:IEEE*, 2015:2227–2235.
- [14] Xu Zhe, Li Zhuo, Chen Xin. Multi-task assignment algorithm for mobile crowdsensing[J]. *Journal of Computer Applications*, 2017,37(1):18–23. [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.]
- [15] Liu Yan, Guo Bin, Wu Wenle, et al. Multitask-oriented participant selection in mobile crowd sensing[J]. *Chinese Journal of Computers*, 2017,40(8):1872–1887. [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]. *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.]
- [16] Antonio C, Annalisa S, Floriano D R. CRAWDAD: The unicat/socialblueconn dataset[EB/OL]. (2015-02-08)[2017-05-07]. <http://crawdada.org/unicat/socialblueconn/20150208>.
- [17] Dimitrios G, Akestoridis, CRAWDAD: The uoi/haggle dataset[EB/OL]. (2016-08-28)[2017-05-07]. <http://crawdada.org/uo/haggle/20160828>.
- [18] Keranen A, Ott J, Käykkäinen T. The ONE simulator for DTN protocol evaluation[C]//Proceedings of the 2nd International Conference on Simulation Tools and Techniques. *Brussels:Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2009:55.
- [19] Lenstra J K, Kan A H G R, Brucker P. Complexity of machine scheduling problems[J]. *Annals of discrete Mathematics*, 1977,1:343–362.
- [20] Musolesi M, Mascolo C. A community based mobility model for ad hoc network research[C]//Proceedings of the 2nd International Workshop on Multi-hop Ad-Hoc Networks: From Theory to Reality. *New York:ACM*, 2006:31–38.

(编辑 赵 婧)

引用格式: Li Zhuo, Xu Zhe, Chen Xin, et al. OTAP: Online multi-task assignment algorithm with prediction for opportunistic crowd sensing[J]. *Advanced Engineering Sciences*, 2018,50(5):176–182. [李卓, 徐哲, 陈昕, 等. OTAP: 基于预测的机会群智感知多任务在线分配算法[J]. *工程科学与技术*, 2018,50(5):176–182.]