

第3章 移动群智感知任务单目标优化分配

任务分配是移动群智感知系统中的一个基础问题,其主要解决群智感知任务与用户资源之间的适配问题,即针对定制化的感知任务,基于任务需求与用户感知能力之间的匹配计算挑选最合适的用户执行任务。一般而言,移动群智感知任务分配的基本流程是:从任务的需求多样性、多点并发性、动态变化性等特征出发,根据任务的时空特征、技能需求及用户个人偏好、移动轨迹、移动距离、激励成本等设定优化目标和约束,进而实现群智感知任务的优化分配。可以说,移动群智感知任务分配对数据采集的全面性、任务完成率和数据采集质量等都具有重要影响^[1-2]。

在任务分配阶段,往往涉及多个不同的优化目标。例如,在保证任务质量的前提下,通过最小化任务参与者的数量,可以有效节省群智感知的数据采集成本^[3-4];在给定用户激励成本的约束下,最大化移动群智感知的时空覆盖率,可以有效改善群智感知任务的完成质量^[5];在给定需要完成的任务总量前提下,通过最小化参与者完成任务移动的总距离,可以降低群智感知数据采集的用户成本等^[6]。按照任务分配中所考虑的优化目标的个数,将这部分工作分为基于单目标优化的任务分配和基于多目标优化的任务分配,分别在第3章、第4章进行详细介绍。

3.1 面向即时任务的多任务工作者选择

3.1.1 问题分析

为移动群智感知任务选择合适的执行者是任务能否顺利完成的关键。对于多任务而言,移动群智感知通用任务平台中的所有任务共享平台“候选用户池”中的资源,考虑到不同任务在时间-空间以及技能需求方面的差异性需求,在众多的工作者中选择一个最合适的用户子集无疑是一个极具挑战性的问题。面向单任务的工作者选择已有较多研究且相对简单,但对于一个通用化的任务平台而言,任意时刻可能存在多个并发的感知任务需要处理,因此对多任务场景下的工作者选择研究十分必要。由于任务之间在时空上可能相邻,因此一次性分配多个任务给工作者不仅可以提高工作者的收益,也能提高任务的完成效率,提升平台的整体效能。

移动群智感知通用任务平台中有一部分任务需要在短时间内完成,本文将这类任务称为“即时任务”。例如,暴风雨过后,收集积水街道的信息;观测重要路口的交通状况,报告交通事故信息等^[7-8]。这些任务都具有高实时性这一特点。在这种场景下,通用感知平台需要选择一组工作者,让他们有目的地前往任务点进行感知数据的收集,如图3-1所示。为了最大化地利用工作者资源,并且提升感知数据的质量,本文规定:一个工作者可以完成多个任务,同时一个任务需要多个工作者完成。假设任务的激励与工作者移动的总距离成正比,那

么,为了最小化平台成本,工作者选择的目标在于选择一组工作者,使得所有任务被完成时所有参与工作者的移动总距离最小。

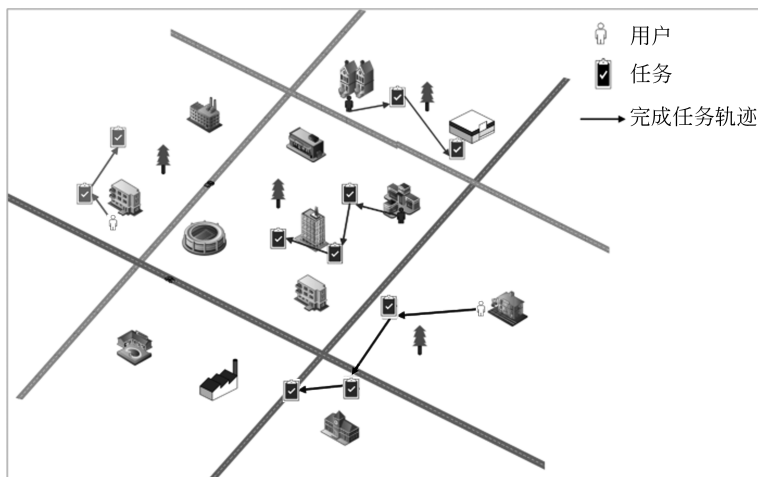


图 3-1 面向即时任务的多任务工作者选择图示

为了解决面向即时任务的多任务工作者选择问题(Worker Selection for Time-sensitive Tasks, WSTS),本文将对问题进行形式化描述,进而根据问题的特点设计有效的算法进行求解。

3.1.2 问题定义

假设待分配的任务集合为 $T = \{t_1, t_2, \dots, t_n\}$, 对于第 $i (1 \leq i \leq n)$ 个任务, 其任务地点为 tl_i , 需求工作者数量为 p_i 。对于即时任务来说, 需要获取备选工作者的当前位置。备选的工作者全集为 $W = \{w_1, w_2, \dots, w_m\}$, 每个工作者 w_j 的当前位置是 $wl_j (1 \leq j \leq m)$ 。为保证任务的完成质量, 设每个工作者一次最多可以分配到 q 个任务。WSTS 就是寻求一个最佳的任务分配方式, 使得所有任务被完成时工作者移动的总距离最小。

本文使用工作者当前的地理位置以及任务的地理位置计算工作者完成任务需要移动的空间距离。在智慧城市市场下, 考虑到工作者需要穿越城市的街区完成任务^[9], 本文使用曼哈顿距离计算两点之间的距离。如果任务和工作者的地理位置用经纬度表示, 那么, 两个位置点 $l_1(lat_1, lon_1), l_2(lat_2, lon_2)$ 之间的曼哈顿距离由式(3-1)计算, 其中 α, β 分别是单位纬度和单位经度与距离的换算因子。

$$\text{dist}(l_1, l_2) = |lat_1 - lat_2| * \alpha + |lon_1 - lon_2| * \beta \quad (3-1)$$

本文使用 $tu_1, tu_2, \dots, tu_m$ 表示每个工作者分配到的任务集合, $\text{dist}(ul_i, tu_i)$ 表示第 i 个工作者完成其分配到的任务集合 tu_i 时的移动距离。基于以上定义, WSTS 可以形式化为式(3-2)。

$$\min \sum_{j=1}^m \text{dist}(ul_i, tu_i) \quad (3-2)$$

满足条件:

$$|tu_j| \leq q \quad (1 \leq j \leq m) \quad (3-3)$$

$$|\{u_j \mid u_j \in W, t_i \in tu_j\}| = p_i \quad (3-4)$$

WSTS 是一个组合优化问题且是一个 NP^[9-10] 难问题,它可以被规约为经典的旅行商问题(Travelling Salesman Problem, TSP): 假设每个任务需求的工作者数量是 1,并且每个工作者能完成的任务数量超过任务的总数量,这样 WSTS 就转化成为一个工作者寻找一条最短路径遍历所有任务的问题。WSTS 问题的解空间非常大^[11],假设任务和工作者的数据分别是 10 和 20,那么 WSTS 的潜在分配方案规模将会是 $20 \times 10!$,这是一个不可能计算出的结果,因此需要寻找高效的解决方法对 WSTS 问题进行求解。

3.1.3 贪心解法 NearsFirst

贪心算法是解决许多 NP 难问题普遍采用的策略,本章将针对 WSTS 提出的贪心算法称为 NearsFirst,因为算法每次选择与给定任务距离最近的一组工作者执行任务。NearsFirst 算法的伪代码如下。

算法 1 NearsFirst 算法

输入: 任务集合,工作者集合,每个工作者可以获得的任务数的限制。

输出: 分配结果<任务,工作者>元组。

1. 计算任意 $t \in T, w \in W$ 和 w 之间的距离,构建距离矩阵 $D_{|T| \times |W|}$ 。
 2. 对于 $D_{|T| \times |W|}$, do。
 3. 在矩阵 D 中选择距离最近的元组 $\langle t, w \rangle$,将 t 分配给 w 。
 4. 去除已经拥有足够工作者的任务和已有 q 任务的工作者。
 5. 更新距离矩阵 $D_{|T| \times |W|}$ 。
 6. 循环执行 2~5 步。
 7. 直至 $T \neq \emptyset, W \neq \emptyset$ 与 t 还没有被分配给 w 这 3 个条件有一个条件不满足。
 8. 结束。
-

在每一次迭代过程中,对于每一对“任务-工作者”组合 $\langle \text{task}, \text{worker} \rangle$,算法计算它们之间的曼哈顿距离;然后遍历所有组合,选出距离最近的一对组合作为问题求解的方案。接下来,算法对刚刚被分配的任务,将其需求人数减 1。如果此时该任务的需求人数变为 0,则该任务分配完毕,对于刚刚分配到任务的工作者,将其已分配任务数加 1;如果此时该工作者已获得任务数量达到阈值 q ,则该工作者不再被分配任务。如果没有任务需要被分配或者没有工作者可以被选择,则算法终止。

NearsFirst 无法得到全局最优的解,因为它每次仅选择当前最优的组合进行分配,因此需要设计全局搜索能力更强的算法。正如在问题定义中所述的,WSTS 问题的解空间非常大,传统的组合优化算法难以解决,因此本文采用智能进化算法^[2](遗传算法)进行问题的求解。

3.1.4 遗传算法

遗传算法是一种模拟自然界优胜劣汰的启发式算法,能够以较低的运算时间复杂度解决大规模的组合优化问题。在问题定义部分,本文已经提到 WSTS 可以规约为普通的旅行商问题,已有的研究文献中也有将遗传算法应用到节点数非常大的旅行商问题中^[12]的情况。

在遗传算法中,一个群体经数次更新迭代后,逐渐收敛到具有某个适应度值的个体上。

在每一代,具有最优适应度的个体将有更高的概率存活下来并且保留到下一代中。遗传算法中的每一个个体称为一段基因,一种基因就是问题的一个可行解(不一定是最优的)。每一次的更新迭代就是要淘汰较差的个体,保留较好的个体。一次更新迭代包括下面 3 种操作。

(1) 选择操作(selection operator): 选择操作就是选择较好适应度的个体,根据一定的规则淘汰适应度差的个体。

(2) 交叉操作(crossover operator): 利用选择操作留下来的个体,通过交叉操作产生新的个体,以补充在选择操作中被淘汰的个体。

(3) 变异操作(mutation operator): 为了避免个体陷入局部解空间,通过变异操作改变现有种群中的部分个体。

为了保证上述操作顺利进行,遗传算法需要定义个体基因的编码表示方式以及个体适应度的评价方法。基因的表达是指采用何种数学形式编码和表示问题的解^[13],常用的个体编码方式有二进制编码、实数编码、向量编码、矩阵编码等。个体的适应度是衡量个体优劣的函数,一般是个体编码到实数值的映射,也需要根据问题具体定义。

总体来说,遗传算法将初始生成的个体尽量均匀地散布在解空间^[14],然后根据个体适应度计算的结果,通过选择、交叉、变异操作,使整个种群中的个体不断向当前种群中适应度优的个体靠近,最终达到整个种群都逼近最优解的目的。但是,由于 WSTS 问题的解空间^[15]非常大,一个种群内的个体基本很难有机会靠近最优解或者次优解,因此,在种群的初始化上,传统的随机初始化^[16]便不再适用。已有的研究表明,在基本的遗传算法基础上,结合一些其他的启发式方法,能够减小种群进化过程中的随机性,加快收敛的速度^[17]。考虑到上面所述贪心解法 NearsFirst 的解可以使用到遗传算法的种群初始化中,因此本章提出结合贪心算法和遗传算法的混合式优化求解方法解决 WSTS 问题。

3.1.5 融合贪心策略的遗传算法(GGA-I)

将贪心算法的输出解运用到遗传算法的输入中,本文提出融合贪心策略的遗传算法(greedy-enhanced genetic algorithm for intentional movement,GGA-I)用于解决面向即时任务的多任务分配问题。GGA-I 以 NearsFirst 算法的输出结果为基础构建初始解,因此其最终解的质量将不低于 NearsFirst 算法。图 3-2 是 GGA-I 算法的流程图。GGA-I 算法的

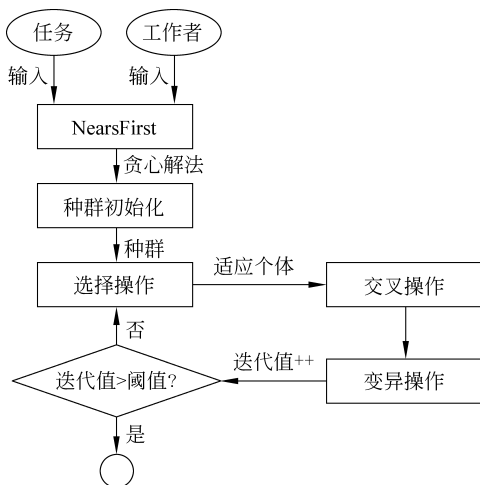


图 3-2 GGA-I 算法的流程图

伪代码见算法2,算法中的具体细节将在下文介绍。

1. 个体编码

WSTS 问题是在给定移动群智感知任务与可用工作者之间搜寻一个最优匹配解,为了表示任务与工作者之间的分配关联关系,本文使用矩阵编码表示每个候选个体:个体中行代表每个工作者,列表示给定的任务集合;矩阵中的元素值 (i, j) 为“1”表示将 j 任务分配给 i 工作者,为“0”则表示不分配。图3-3所示矩阵就是 GGA-I 个体编码的一个示例,表示当任务数为6,工作者数量为8时的个体编码结构。

算法2 GGA-I 算法

输入: NearsFirst 算法的输出,任务集合 T ,工作者集合 W ,迭代次数阈值 g_{thld} 。

输出: 分配结果<任务,工作者>元组。

1. 根据 NearsFirst 算法的结果构建任务分配矩阵 $A_{m \times n}$ ($m = |T|, n = W$),矩阵形式与个体编码方式一致。
2. 初始化种群,建立种群的表示矩阵 $p_{c \times m \times n}$, c 是种群中个体的数量。
3. 设置迭代次数为0,执行。
4. 计算 $p_{c \times m \times n}$ 所有个体的适应度 $f(i_k)$ ($0 \leq k < c$)。
5. 更新当前最好适应度的个体 I 。
6. 根据选择操作选择种群中的优胜个体。
7. 根据交叉操作复制个体填充 $p_{c \times m \times n}$ 中被淘汰的个体。
8. 对 $p_{c \times m \times n}$ 执行交叉操作。
9. 迭代次数增加1。
10. 循环执行4~9步。
11. 直至迭代次数达到阈值 g_{thld} 。
12. 根据 I 输出分配结果<任务,工作者>元组。
13. 结束。

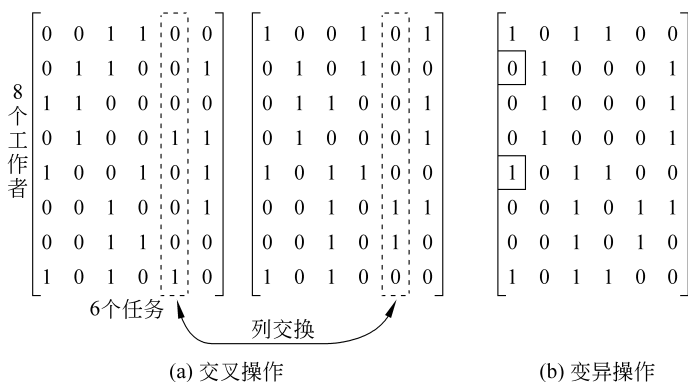


图3-3 GGA-I算法的交叉操作和变异操作示意图

根据 WSTS 问题的定义,特别是式(3-3)和式(3-4),这里对个体的编码定义两个限制条件:任务可行性和工作者可行性。只有当一个个体的编码同时满足任务可行性和工作者可行性时,该个体才是一个可行解;否则,便不是可行解。本文使用的遗传算法中的选择、交叉

及变异操作均需保证在种群进化过程中个体的可行性。

- 任务可行性：每一列元素的和等于该列对应任务的工作者需求数量,任务可行性保证每个任务都分配到数量足够的工作者。
- 工作者可行性：每一行元素的和不超过工作者可以获得的最大任务数,工作者可行性保证每个工作者获得的任务数量不超过上限限制。

2. 种群初始化

初始种群对遗传算法的结果具有显著的影响,因为它是遗传算法个体更新和演化的基础。一般情况下,初始种群中的个体应该均匀分布在解空间,但由于 WSTS 问题的解空间巨大,均匀分布的个体可能离最优解差距很大^[18],所以 GGA-I 使用 NearsFirst 的结果初始化种群。

3. 适应度函数

在 WSTS 中,需要优化的目标是所有任务被完成时工作者移动的总距离,这个总距离是所有获得任务的工作者移动距离之和。GGA-I 中的每一个个体对应一种任务分配方案,该方案下工作者移动的总距离是确定的。因此,GGA-I 中个体的适应度与个体对应的总移动距离相关。设种群中的所有个体是 $I = \{i_1, i_2, \dots, i_n\}$,第 k 个个体从 $i_k (1 \leq k \leq n)$ 对应的总移动距离为 $Td(i_k)$,那么个体 i_k 的适应度 $f(i_k)$ 可以用式(3-5)计算:

$$f(i_k) = \frac{Td(i_k)}{\sum_{j=1}^n Td(i_j)} \quad (3-5)$$

由于优化的目标是求最小值,所以个体适应度值越小,个体对应的解越优。

4. 选择操作

选择操作的作用是将具有较好适应度的个体传承到下一代。但由于适应度较劣的个体中仍然可能包含较好的基因片段,所以 GGA-I 使用轮盘赌(roulette wheel)方法进行个体的选择^[19],其选择个体的原则是具有较优适应度的个体有较大的概率保留到下一代,具有较差适应度的个体有较低的概率保留到下一代。具体来说,在每一代的选择操作时,首先累加所有个体的适应度得到适应度之和,然后计算每个个体的适应度与适应度之和的比值,该比值就是该个体被选中的概率。

5. 交叉操作

交叉操作用于繁衍新的个体,它模拟自然界中生物的基因重组过程^[20]。交叉操作的使用也需要根据个体的编码方式以及问题的定义进行。对于我们所定义的 WSTS 问题,GGA-I 使用个体编码矩阵对应列交换的方式进行交叉操作,即交换两个父母个体对应列的元素得到两个子个体,图 3-3(a)是交叉操作的一个示例。之所以选择这样的方式,是因为这种交叉操作产生的两个子个体一定是满足任务可行性的。所以,在这种方式下只需要考虑所交换的列,使得交换之后产生的新个体也可同时满足工作者可行性。

6. 变异操作

如果遗传算法中缺少变异操作,那么整个种群可能会陷入局部最优,无法向更好的

结果逼近,变异操作则带来了这种转变的机会。在 GGA-I 中,本文随机选择一个值为“1”的元素,将其值变为“0”,为保证这样改变之后,个体仍然满足任务可行性和工作者可行性,本文挑选与改变元素同一列的另外一个值为“0”的元素,将其值变为“1”,如图 3-3(b) 所示。

3.1.6 实验验证

我们提出的多任务工作者选择方法针对的是位置相关的感知任务,因此,为了模拟工作者在城市中的位置以及移动规律,采用了一个公开的电信通话记录数据集 D4D。该数据集包含法国 Orange 电信公司在科特迪瓦黄金海岸地区 50 000 名客户两个星期的通话数据记录。每一条记录由匿名化的用户 ID、通话日期和时间,以及通话发生时的基站 ID 组成。所有基站的数目是 1 231 个。考虑到科特迪瓦首都阿比让地区的基站密度最大(347 个),人的地理位置的精确度最好,因此仅筛选了阿比让地区的通话数据记录进行实验,如图 3-4 所示。

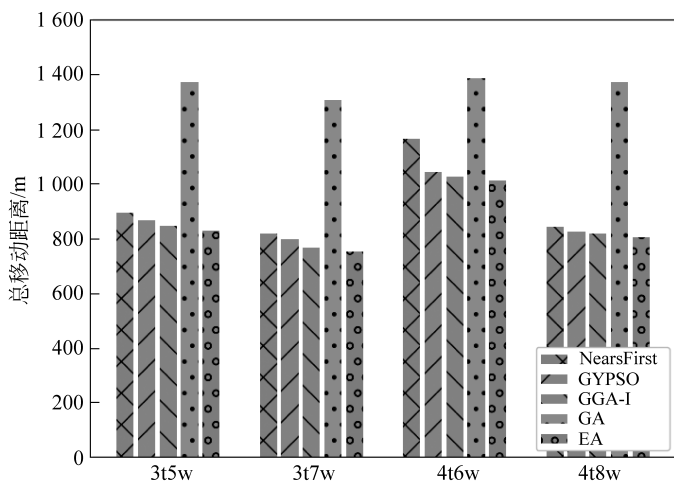


图 3-4 在不同数量任务-工作者组合下的总移动距离

为了评价本章提出的方法的优劣,也对其他 3 种算法进行了对比。

GA: 表示传统的遗传算法^[21],以未融合贪心算法的结果作为种群的初始化。

EA: 表示枚举算法^[22]。为了评估前面提出算法的真实性能,实现了枚举算法来搜索问题的整个解空间,以获取问题的最优解。鉴于多任务的选人问题 WSDT 和 WSTS 解空间都非常大,本文只在任务数和工作者数量比较小的时候运行出实验结果。

GYPSO: 粒子群优化(Particle Swarm Optimization, PSO)算法是另一种常见的组合优化算法^[23]。它同样是一种进化算法,是通过模拟鸟类觅食行为而建立起来的一种基于群体协作的随机搜索算法。与遗传算法一样,粒子群优化算法进化的基础也是种群,它们之间的区别是,粒子群优化算法没有选择、交叉和变异操作。粒子群优化算法中的个体通过对比自身当前状态与全局最优和当前种群最优状态,向这两种最优状态靠近,类似于鸟群中的个体通过观察其他鸟的行为调整自己当前的搜索策略。为与融合了贪心算法和遗传算法的

GGA-I 对比,本文也使用贪心算法的结果初始化粒子群算法的种群。

WSTS 问题的优化目标是所有任务完成时工作者的总移动距离,图 3-4 显示了在 4 种不同的任务-工作者组合下 5 种算法的实验结果(为方便描述,使用 3t5w 这样的记法表示在 5 个工作者中分配 3 个任务,下文同理)。针对每种任务-工作者数量,本文重复 20 次实验得到平均值。图 3-4 显示出 GGA-I 较其他 4 种算法选择的工作者更优。特别地,GGA-I 在结果上优于 NearsFirst,表明在贪心算法的结果上进行遗传算法中的进化操作确实能提高 NearsFirst 的效果。另一方面,GGA-I 大大优于单一的 GA 算法,说明其采用贪心算法的结果初始化种群对获取到更优解有很大益处。此外,GGA-I 可以取得比 GYPSO 更好的效果,因为粒子群算法更容易陷入一个局部最优解中。在当前任务数和工作者数量较少的情形下,EA 的结果可以计算出来。可以看到,GGA-I 的结果很接近 EA 的结果,表明使用 GGA-I 算法的结果比较接近最优解。

5 种算法的时间复杂度随工作者数量的增加对比结果如图 3-5 所示。图中,纵轴显示的是时间复杂度的对数。每种工作者数量下任务数量保持为 4。可以看出,尽管 EA 的工作者选择能够得到最优解,但是其时间复杂度比其他 4 种算法大很多,且时间复杂度随着工作者的数量呈指数级增长。GGA-I 比 NearsFirst、GYPSO 和 GA 的时间复杂度稍大,但是 GGA-I 能取得更准确的结果。因此,综合考虑时间复杂度和计算结果有效性,GGA-I 表现更优。

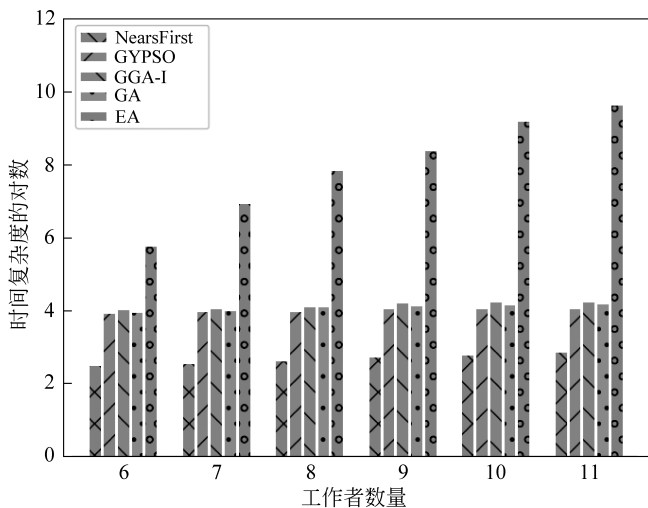


图 3-5 各种算法的时间复杂度的对数

图 3-6 中,WSTS 问题的解空间随着问题规模的增大急剧扩张,但是 GGA-I 的运算时间并没有增加很多,表 3-1 中的实验结果更加清晰地表明了这一点。因此,当较大数目的感知任务需要被分配时,GGA-I 会取得更好的效果。

任务的完成时间对于即时任务很重要,完成时间越短越好。本文比较了 4 种算法工作者选择结果的任务完成时间。从图 3-7 中可以看出,GGA-I 的任务完成时间最短,这主要是因为 GGA-I 中工作者的总移动距离最短。

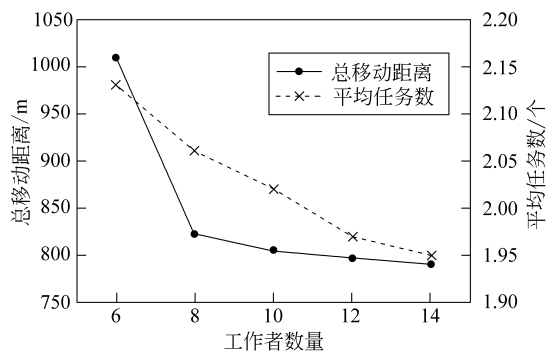


图 3-6 GGA-I 算法中的总移动距离和工作者平均任务数随工作者数量的变化关系

表 3-1 GGA-U 一次迭代的运行时间

任务-工作者	10t20w	20t40w	30t60w	40t80w	50t100w
一次迭代的运行时间/s	0.034	0.043	0.056	0.072	0.089

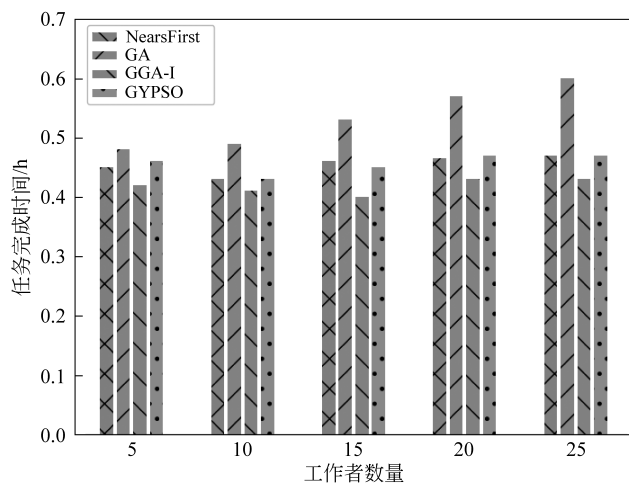


图 3-7 不同算法任务完成时间

3.2 面向容延任务的多任务工作者选择

3.2.1 问题分析

在另外一种感知应用场景下,移动群智感知通用任务平台中存在时间容延这一类型的感知任务。该类任务的特点是:区别于即时任务对任务完成时间的紧迫性(如1~2h内),容延任务允许任务执行者在未来相对较长的一段时间内完成任务。一般而言,这类任务较多针对状态稳定/变化缓慢的感知对象。例如,对某个区域内的植物进行拍照研究植物生长与气候变化^[24]之间的关系,或是收集一个商业街最近的打折销售信息^[25]等。对于这一类

任务,平台可以选择在日常行程中可能经过任务位置点的工作者完成任务,工作者无须特意前往位置点,如图 3-8 所示。在这种情况下,工作者完成一个任务的代价相对较小,因此,为了提高每个工作者的收益,同时提高工作者资源的利用效率,对于平台而言,最佳的选择是利用最少的工作者完成多个不同的任务。

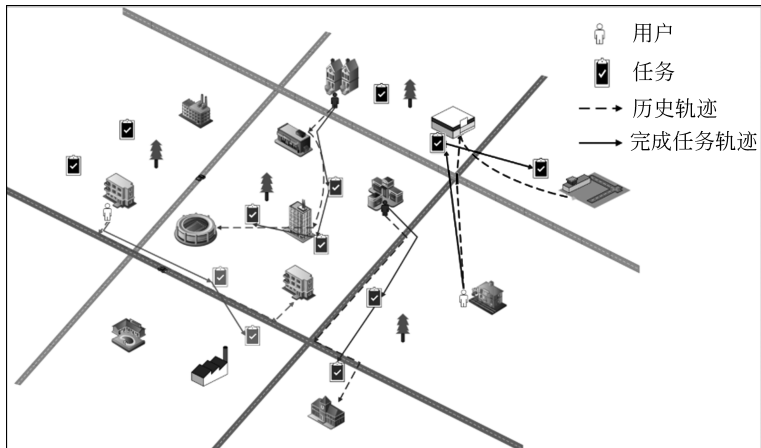


图 3-8 面向容延任务的多任务工作者选择图示

为了解决上述面向容延任务的多任务工作者选择问题(Worker Selection for Delay-tolerant Tasks, WSDT),下面将首先对该问题进行形式化描述,并根据问题的性质分析提出求解方法。

3.2.2 问题定义

WSDT 为了减轻工作者的负担,倾向为工作者分配其日常行为轨迹上的感知任务。为此,需要根据工作者的历史轨迹预测其未来一段时间内的地理位置,所以 WSDT 问题的解决需要通过下面两个阶段完成。

- 工作者移动预测: 建立工作者的移动模式,预测工作者是否会经过任务地点。
- 优化的工作者选择策略: 基于工作者移动预测,选择最少的工作者完成感知任务。

1. 工作者移动预测

设工作者 w 的历史地理位置记录为 $lr = \{r_1, r_2, \dots, r_s\}$, 其中每一个地理位置记录 r_i 由时间 rt_i 和地理位置 rl_i 组成,任务 t 的位置用 tl 表示。为了简化起见,假设所有任务都需要在 24 小时内完成且所有任务的起止时间区间一致。针对移动位置预测^[26],已有的研究提出了多种相应的方法,这里采用一种较为简单的基于统计的预测方法——以统计的该工作者的历史地理位置记录中经过任务地点的频次作为工作者未来经过该任务地点的概率。具体而言,工作者 w 经过 tl 的概率可以用式(3-6)计算。在该公式中, $|\{rt \mid rt \in \{rt_1, rt_2, \dots, rt_s\}\}|$ 是工作者地理位置记录中的所有时段片数目, $|\{rt_i \mid tl = rl_i\}|$ 是时段片内工作者位置记录出现任务地点的所有时段片数目。

$$p(w, t) = \frac{|\{rt_i \mid tl = rl_i\}|}{|\{rt \mid rt \in \{rt_1, rt_2, \dots, rt_s\}\}|} \quad (3-6)$$

为了保证工作者选择方案的基本质量,本文定义阈值 R_{thld} ,只有当工作者经过任务地

点的概率大于该阈值时,任务才能被分配给工作者;反之,则不会。若一个工作者 w 能被分配任务 t ,本文称 w 可以完成 t ,或者说 t 可以被 w 覆盖。

2. 问题形式化

给定一个待分配的任务集合 $T = \{t_1, t_2, \dots, t_n\}$, 对于第 i ($1 \leq i \leq n$) 个任务, 其任务地点为 tl_i , 需求的工作者数量为 p_i 。备选的工作者全集为 $W = \{w_1, w_2, \dots, w_m\}$, 最终选择的工作者集合为 $WS = \{w_1, w_2, \dots, w_s\}$, 工作者 w_i 得到的任务集合为 tu_i 。依据以上定义, WSDT 问题可以定义为

$$\min |WS| \quad (3-7)$$

满足条件:

$$\forall t_i, |\{w_j \mid w_j \in WS, t_i \in tu_j\}| = p_i \quad (3-8)$$

$$\forall t_i, w_j, t_i \in tu_j, p(t_i, w_j) \geq R_{\text{thld}} \quad (3-9)$$

式(3-8)保证每个任务有足够数量的工作者完成, 式(3-9)保证工作者经过每个分配给她/他的任务地点的概率都大于阈值 R_{thld} 。很显然, WSDT 是一个 NP 难问题, 事实上, 它可以被规约到经典的集合覆盖问题(set cover problem): 假设每个任务需要的工作者数量为 1, 每个工作者可以完成若干个任务, 即任务全集的子集, 这样 WSDT 问题就转化成寻求一个工作者集合, 使得整个工作者集合可以覆盖到整个任务全集, 这就是集合覆盖问题。

与 WSTS 不同, WSDT 问题的数学形式有一些不同的性质。给定 W 的任意子集 WS , 可以定义一个 WS 的效用函数 $f(WS)$ 衡量 WS 中的工作者可以覆盖任务全集的程度, 也就是 WS 中的工作者可以完成多少任务。可以证明 $f(WS)$ 函数具有非负性、单调递增性和子模性^[27]。

3.2.3 贪心算法 MostFirst

算法 3 MostFirst 算法

输入: 任务集合 T , 工作者集合 W , 工作者途径任务地点的概率阈值 R_{thld} 。

输出: 分配的<任务, 工作者>元组。

1. 对于所有 $t \in T, w \in W$, 计算 w 经过 t 的概率 $p(w, t)$ 。
2. 对于每个工作者 $w_i \in W$, 计算集合 $\{j \mid p(w_i, t_j) \geq R_{\text{thld}}, 0 \leq j < |T|\}$ 的模 c_i 。
3. 选择 c_i 值最大的工作者 w , 则 w 可以完成的任务分配给 w , 将 w 从 W 中移除。
4. 如果任务 t 分配到足够的工作者, 则将 t 从 T 中移除。
5. 循环执行 3~4 步。
6. 直至 $T = \emptyset$ 或者 $W = \emptyset$ 。
7. 结束。

对于 WSDT, 首先提出贪心算法 MostFirst。MostFirst 首先计算所有“任务-工作者”组合中, 工作者经过任务地点的概率。WSDT 的优化目标是最小化工作者选择的数量, 因此算法每次选择时挑选能完成任务数量最多的工作者。算法的伪代码见算法 3。

与 NearsFirst 类似, MostFirst 的结果也是次优的。因此, 本文将继续提出结合贪心算法和遗传算法的方法进一步解决 WSDT, 从而获得更好的结果。

3.2.4 融合贪心算法的遗传算法(GGA-U)

与 WSTS 问题的求解类似, 这里同样设计一个融合了贪心算法和遗传算法的方法解决

面向容延任务的多任务工作者选择问题, 简称为 GGA-U (Greedy-enhanced Genetic Algorithm for Unintentional movement)。GGA-U 的遗传算法部分的输入也是 MostFirst 的结果。下面是 GGA-U 算法的基本流程。

- 个体编码: 在 WSDT 中, 一旦一个工作者被选择, 他将获得所有自己能完成的任务。对于一个备选工作者而言, 本文使用“1”表示其被选中, 用“0”表示其未被选中。因此, 一个 0-1 表示的向量即可用来表示 WSDT 的解, 向量的维度就是备选工作者全集的模值。
- 种群初始化: 与 GGA-I 类似, GGA-U 使用 MostFirst 算法的结果初始化种群; 首先获取 MostFirst 算法的结果, 然后根据结果中任务的分配情况得到对应的个体编码结果。
- 适应度: WSDT 问题的优化目标是最小化所有选择的工作者的数量, 因此个体中选择的工作者数量越少, 适应度越好。个体的编码向量中使用“1”表示工作者被选择, 所以一个个体的适应度与其向量编码方式中“1”的个数有关。设种群中的所有个体是 $I = \{i_1, i_2, \dots, i_n\}$, 个体 $i_k \{1 \leq k \leq n\}$ 中“1”的个数是 c_k , 则个体 i_k 的适应度 $f(i_k)$ 可以用式(3-10)计算:

$$f(i_k) = \frac{c_k}{\sum_{j=1}^n c_j} \quad (3-10)$$

$f(i_k)$ 值越小, 个体的适应度越好。

- 选择操作(selection operator): 选择操作淘汰适应度差的个体, 与 GGA-I 类似, GGA-U 也采用轮盘赌的方法淘汰个体。
- 交叉操作(cross over operator): GGA-U 采用交换个体编码向量部分片段的方式进行交叉操作, 如图 3-9 所示。

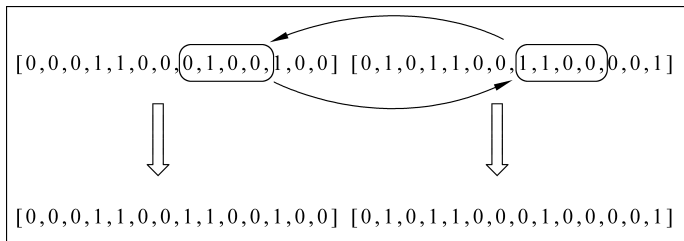


图 3-9 GGA-U 算法的交叉操作示意图

- 变异操作(mutation operator): GGA-U 使用随机选取个体向量编码部分的元素, 将元素的值取反, 即将“1”转换为“0”, 将“0”转换为“1”。

3.2.5 实验验证

在 WSDT 问题中, 根据工作者的移动模式进行最优工作者的选择。当任务地点确定后, 使用 D4D 数据集建模工作者的移动模式, 预测工作者在未来 24h 内是否会经过任务地点。如果在 D4D 数据集中, 工作者确实在 24h 内在任务地点有通话记录数据, 则该工作者

具备完成该任务的条件。每个任务需要的工作者数量同样随机地分布在 2~4。

工作者移动预测准确率是指根据工作者的历史地理位置记录,选择的工作者会经过任务地点的平均概率(预测概率),移动预测的准确率也就是任务的预测完成率;任务的实际有效完成率是指工作者最终确实完成了分配给他的任务的比率,例如,有 100 个单位任务分配 100 人次完成,若最终这 100 个单位任务有 92 个被完成,则任务实际有效完成率为 92%。很显然,工作者移动模式预测的准确率影响任务的实际完成率,因为不准确的预测会导致一些不会经过任务地点的人被选中执行该任务。本文测试了多种情形下的移动预测准确率和任务实际有效完成率。图 3-10 显示了随着阈值 R_{thld} 的增加,任务的预测完成率和实际有效完成率的变化情况。可以看到,任务的实际有效完成率与任务的预测完成率很接近,表明本文使用的移动预测方法是有效的(根据移动预测结果选择的工作者基本能够完成任务)。任务的预测完成率和实际有效完成率始终大于阈值 R_{thld} ,这是因为 R_{thld} 只是工作者能够被选择的基础值,只有工作者经过任务地点的概率被预测大于 R_{thld} 时,任务才会被分配给工作者。

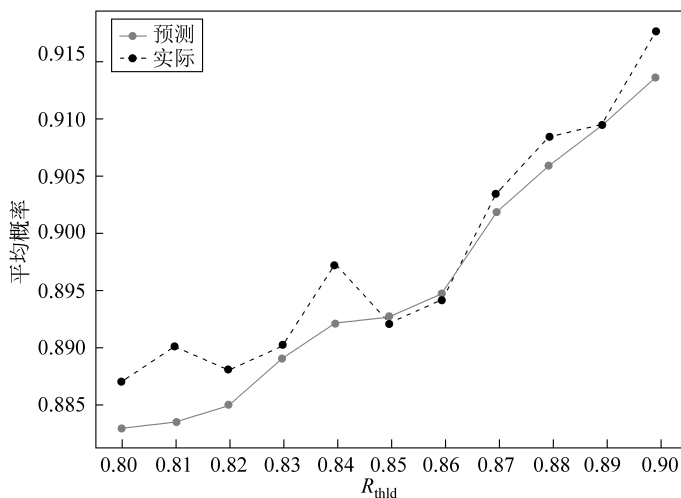


图 3-10 任务的预测完成率和实际完成率随阈值 R_{thld} 的变化情况

本文研究了在 3 种不同任务地理分布下工作者选择数量的变化情况(见表 3-2),分别是集中分布、分散分布以及混合分布。实验中设定需要被分配的任务数目是 20,采取了 R_{thld} 的两种值: 80%, 90%。对于每一种任务分布,实验都生成了 3 个不同的任务集合,在表中标注为 1、2 和 3。实验结果见表 3-2,可以看到,在所有组合情形下,GGA-U 算法选择的工作者人数少于 MostFirst 算法选择为的工作者人数。当阈值 R_{thld} 较小时,两种算法选择的工作者数量更少,这是因为 R_{thld} 越小相当于工作者得到一个任务的条件更宽松,那么工作者可以完成的任务就可能越多,因此总的人数便会减小。同时表明,当 R_{thld} 达到一定值后, R_{thld} 值的增大对任务的预测完成率和实际有效完成率并不会带来较大的影响。因此,使用一个较低的阈值 R_{thld} ,可以使得 GGA-U 选择的工作者数量变少,同时对于任务的实际有效完成率不会有太大影响。

表 3-2 算法选择的工作者的数量

(a) 集中分布			(b) 分散分布			(c) 混合分布		
任务集	GGA-U	Mostfirst	任务集	GGA-U	Mostfirst	任务集	GGA-U	Mostfirst
$R_{\text{thld}} = 90\%$			$R_{\text{thld}} = 90\%$			$R_{\text{thld}} = 90\%$		
1	36	40	1	38	41	1	38	41
2	36	42	2	37	44	2	36	42
3	39	45	3	40	48	3	40	47
平均值	37	42.3	平均值	38.3	44.3	平均值	38	43.3
$R_{\text{thld}} = 80\%$			$R_{\text{thld}} = 80\%$			$R_{\text{thld}} = 80\%$		
1	32	35	1	34	36	1	34	36
2	31	34	2	34	37	2	33	36
3	31	36	3	32	37	3	33	35
平均值	31.3	35	平均值	33.3	36.7	平均值	33.3	35.7

一般来说,在任务的3种分布形式中,当任务集中分布时,GGA-U选择的工作者数量最少,这主要因为人们在日常生活中更倾向在一个局部的范围内活动,而不是在大范围内转移。当任务分散分布时,任务同时出现在工作者路径上的概率变小,因此更多的工作者会被选中。图3-11展示了随着任务数量的增加,GGA-U和MostFirst两种算法选择的工作者数量的变化趋势。MostFirst选择的工作者数量始终比GGA-U多,而且随着任务数量的增加,MostFirst选择的工作者数量增长得更快。

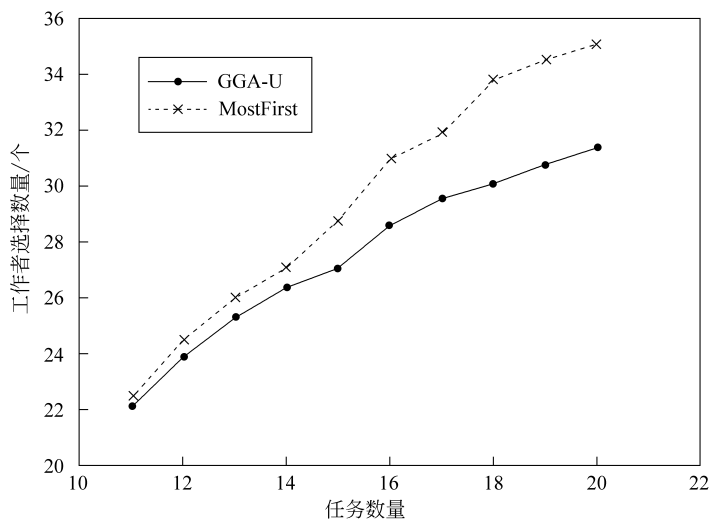


图 3-11 工作者选择数量随任务数量增加的变化趋势

3.3 基于移动社交网络的群智感知社群化任务分发

现有的移动群智感知运行框架绝大多数基于“平台-用户”的运行模式,即移动群智感知平台直接面向参与用户分发移动感知任务,参与用户执行感知任务并将感知数据(如状态数

据、文本数据、图像数据以及音视频数据等)上传至群智感知平台。通过对上述“平台-用户”MCS任务分发模式的分析,发现该模式以感知任务的预期任务完成评估为导向,分发方案制订之后不再更改,缺乏对用户实际执行感知任务情况的灵活、有效的过程监督与适时干预。一旦构建的用户未来行为预测模型存在较大偏差,或是实际的任务执行场景产生显著变化,则该模式难以有效保障感知任务执行的鲁棒性,因此会造成任务完成率低下的问题。而在现实应用场景中,参与用户行为的随机性与动态性将有可能进一步凸显并加剧这一问题。

考虑到离线任务分配场景下用户预测模型精度的有限性所导致的指定用户拒绝执行任务,以及任务执行过程中由于各种突发情况所导致的任务执行失败两种不同的场景,提出以任务完成率为优化目标的基于移动社交网络的群智感知任务分发方法。具体流程如下:基于不同移动社群所表征的空间位置偏好,移动群智感知平台将不同的空间感知任务初次分发给相应的社群(社群组织者)^[28]。社群组织者参照所在社群内部从属者对于相应空间位置的偏好程度、活跃程度以及与社会组织者之间的社交亲密度,以一种随机生成概率方式将任务二次分发给特定的用户进行感知任务的执行。在任务执行过程中,若选定的用户由于现实因素无法继续执行所分派的任务,则其可通知所在社群的组织者重新选择任务执行者,或是利用移动社交网络将该任务传递给高亲密度社交用户(好友)委派其执行感知任务^[29]。在感知结果的回收过程中,利用移动社交网络的机会式通信方式,感知结果将由源节点(任务执行者)通过单跳或是多跳方式传递给社群组织者^[30],再由社群组织者将结果进行初步统计发送给群智感知平台。整个方案的结构流程如图3-12所示。

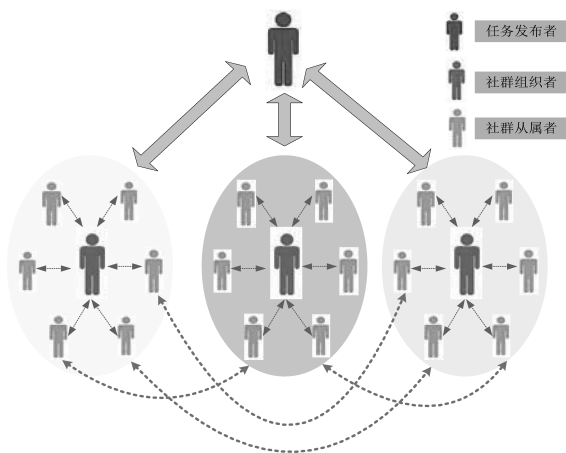


图 3-12 移动群智感知社群化任务分发组织流程图

3.3.1 问题定义

给定 n 个群智感知个体用户集合 $W = \{w_1, w_2, \dots, w_n\}$, 任一用户 w_i 的历史移动轨迹数据表示为 $Tr_{w_i} = \langle loc_j, t_j, day_j \rangle$, 其中 loc_j 为时间戳 t_j 时刻的空间位置, day_j 为相应的日期信息。移动群智感知平台接收任务发布者所发布的 MCS 任务集合 $T = \{t_1, t_2, \dots, t_m\}$, 其中任务 $t_i = \langle loc_i, ti_i, cont_i \rangle$ 表示任务 t_i 需要在指定的空间位置 loc_i 和时间间隔 ti_i 内完成任务描述为 $cont_i$ 的感知任务。对于任意给定的 MCS 任务集合 $T = \{t_1, t_2, \dots,$

t_m }, 任务分发问题的目标在于从潜在可用的参与用户集合 $W = \{w_1, w_2, \dots, w_n\}$ 中寻找一个合适的子集 W^* 以实现 T 中所包含任务的执行以及执行结果的回收过程。按照问题定义以及基本流程描述, 形式化表示移动群智感知任务分配问题, 具体如式(3-11)。

$$\begin{cases} \text{Arg: } \max_{W^* \subset W} \frac{\sum_{i=1}^n \partial_i}{n} \\ \text{s.t.: } \sum_{i=1}^k \sum_{j=1}^n x_{i,j} * mg_i = n \end{cases} \quad (3-11)$$

其中, ∂_i 为标量, 表示任务 t_i 是否完成, 如果 t_i 完成, 则 $\partial_i = 1$; 反之, $\partial_i = 0$ 。 mg_i 表示所划分的任意一个移动用户社群, $x_{i,j} = 1$ 表示将任务 t_j 分配给社群 mg_i ; 反之, 有 $x_{i,j} = 0$ 。

3.3.2 基于移动行为相似度的社群动态划分

1. 移动行为特征

移动行为特征反映用户在给定时间间隔内, 在不同空间位置上的移动时空行为分布特征。本文选择用户的空间出现频次与停留时间长度作为移动行为特征的刻画, 具体可表示为式(3-12):

$$f(w_j, loc_i) = \frac{fre(w_j, loc_i)}{fre(w_j, loc)} \times \frac{dur(w_j, loc_i)}{dur(w_j, loc)} \quad (3-12)$$

其中 $f(w_j, loc_i)$ 表示用户 w_j 在空间位置 loc_i 上的移动行为特征, $fre(w_j, loc_i)$ 与 $fre(w_j, loc)$ 分别表示用户 w_j 在 loc_i 上的访问频次以及在所有空间位置 loc 上的访问频次, $dur(w_j, loc_i)$ 与 $dur(w_j, loc)$ 分别表示 w_j 在 loc_i 上的停留时长以及在所有空间位置 loc 上的停留时长。用户 w_j 在所有空间位置上的移动行为特征分布构成 q 维矢量 $\text{Vec}(w_j, loc) = \{f(w_j, loc_1), f(w_j, loc_2), \dots, f(w_j, loc_q)\}$ 。特征分布刻画的是用户对不同空间位置的“偏好程度”。实际中, 不同用户在空间位置上的时空分布绝对值是不同的, 即 $fre(w_j, loc_i)$ 与 $dur(w_j, loc_i)$ 具有显著的差异性, 以此为基础, 提出社群内用户移动活跃度的概念。

2. 移动活跃度

移动活跃度是相对于移动行为特征分布而言的, 本质上移动行为特征属于相对值, 而移动活跃度属于绝对值。定义移动活跃度为用户 w_j 在不同空间位置上的累次访问频次与停留时长的乘积, 表示为式(3-13):

$$act(w_j, loc) = \sum_{i=1}^q fre(w_j, loc_i) \times \sum_{i=1}^q dur(w_j, loc_i) \quad (3-13)$$

3. 移动时空社群

给定离散时间间隔 ti_i 以及用户集合 W 的历史移动轨迹数据集 Tr_w , 基于不同用户在 ti_i 内的在不同空间位置上的移动行为特征计算, 将用户集合划分为有限个社群 $MG = \{mg_1, mg_2, \dots, mg_k\}$, $mg_k = \{w_{i1}, w_{i2}, \dots, w_{ix}\}$, 其中 w_{ij} , $1 \leq j \leq x$ 对应第 i 个划分的社群 mg_i , 且同一社群内的参与用户具有相似的移动行为特征分布, 不同社群内的用户具有不同的移动行为特征分布。

下面将对移动时空社群的划分过程进行详细阐述。利用移动行为特征的计算公式,可得出每一个参与用户对于不同空间位置的偏好特征值 $f(w_j, loc_i)$, 逐一对用户 w_j 在不同空间位置 $loc_i (1 \leq i \leq q)$ 计算特征值, 并将得到的偏好值以矩阵的形式表示为式(3-14):

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1q} \\ m_{21} & m_{22} & \cdots & m_{2q} \\ \vdots & \vdots & & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nq} \end{bmatrix} \quad (3-14)$$

其中, $m_{ji} = f(w_j, loc_i)$ 。对矩阵 $\mathbf{M}$ 的任意两行矢量进行基于余弦相似度的计算, 具体如式(3-15):

$$\cos\theta_{j1,j2} = \frac{\sum_{i=1}^n (f(w_{j1}, loc_i) \times f(w_{j2}, loc_i))}{\sqrt{\sum_{i=1}^n (f(w_{j1}, loc_i)^2)} \times \sqrt{\sum_{i=1}^n (f(w_{j2}, loc_i)^2)}} \quad (3-15)$$

其中, $\cos\theta_{j1,j2}$ 表示用户 w_{j1} 与 w_{j2} 在所有空间位置上基于移动行为特征分布的相似度。以该相似度值为基础, 利用 k 均值聚类算法对上述 n 个参与用户进行社群划分, 最终得到 k 个不同的移动社群 $MG = \{mg_1, mg_2, \dots, mg_k\}$ 。如上所述, 在每一个划分的移动社群 mg_k 中利用移动活跃度计算, 确定唯一的社群组织者作为与群智感知平台的任务接收与数据上传的终端超级用户, 同时作为所在社群内不同用户从属者之间的任务协调者。

对于所划分的社群在空间位置上的偏好映射, 本文以社群内所有用户空间偏好矢量值的平均值表征社群对空间位置的偏好, 以间接达到利用社群对空间位置进行分组的目的, 即若社群结构为 $mg_k = \{w_{i1}, w_{i2}, \dots, w_{ip}\}$, 则社群对空间位置的偏好 $pf(mg_k, loc)$ 表示为式(3-16):

$$pf(mg_k, loc) = \frac{1}{p} \sum_{x=i1}^{ip} \mathbf{Vec}(w_x, loc) = \frac{1}{p} \sum_{x=i1}^{ip} \{f(w_x, loc_1), \dots, f(w_x, loc_n)\} \quad (3-16)$$

3.3.3 移动群智感知社群化任务分发

1. 概率性随机分发策略

通过对传统群智感知任务的分发流程分析可知, 这一模式大多建立在对用户未来行为的统计性认识之上, 进而以统计性概率值进行排序操作, 按照从高到低的顺序依次选择相应的参与用户执行给定任务^[31]。尽管基于历史数据所发掘的行为统计规律具有一定程度上的稳定性, 但是不可忽视的是身处复杂社会环境下的移动用户其行为往往具有很大的随机性、突发性与不确定性, 这些特征对于确定性的传统任务分发的稳健运行构成了潜在的威胁。此外, 移动群智感知中的用户组织结构松散与参与自发等特点^[32]又进一步强化了用户随机性行为对群智感知平台运行的不利影响, 使得移动群智感知平台的脆弱性进一步加深。移动群智感知用户行为特征模式具有天然的不确定特征, 而传统的任务分发模式其本质却为使用稳定的、确定性的计算方法预测、匹配不确定的行为模式^[33], 这本身就是矛盾的。为此, 我们考虑使用不确定的分发机制以随机生成概率进行任务的分发。

2. 感知任务的初次分发

感知任务的初次分发指 MCS 平台根据给定任务的空间属性,以所划分社群对空间位置的偏好 $pf(mg_k, loc)$ 为依据将任务分派给特定社群(社群组织者)。按照上述概率性随机分发策略,对于给定的 MCS 任务 $t_i = \langle loc_i, ti_i, cont_i \rangle$,以时间间隔 ti_i 对所有可用参与用户进行动态社群划分过程,得到 k 个划分结果 $MG = \{mg_1, mg_2, \dots, mg_k\}$ 。对于每个社群 mg_x ,计算其相对于空间位置 loc_i 的偏好相对程度 $pf'(mg_x, loc_i)$,如式(3-17)所示。

$$pf'(mg_x, loc_i) = \frac{pf(mg_x, loc_i)}{\sum_{x=1}^k pf(mg_x, loc_i)} \quad (3-17)$$

按照 k 个社群对于位置 loc_i 的偏好 $pf'(mg_x, loc_i)$ 以随机概率产生的方式确定任务执行的社群。具体做法为:先将所有偏好值进行归一, $\sum_{x=1}^k pf'(mg_x, loc_i) = 1$,依次将每个社群的偏好值进行 $[0, 1]$ 区间排列,进而在 $[0, 1]$ 区间生成随机数,若该随机数落在第 x 个社群的偏好值范围内,则选择 mg_x 作为该任务的执行社群,同时将该任务分发给选定社群的组织者。为了保障感知数据的真实性,本文采用通用的冗余采样方式,将同一任务独立分发给不同的 g 个社群执行(独立重复上述过程 g 次,若产生相同的社群,则继续该过程,直至产生不同的 g 个社群为止),最后将 g 个社群的感知结果进行组合,验证最终产生的结果。

3. 感知任务的二次分发

当平台将任务 t 分发给社群 mg_x ,在社群内部,社群组织者按照社群从属者用户对于给定任务空间的活跃度以及与其社交的亲密度确定具体的任务执行者,其中活跃度保证所选择的具体任务执行者对于给定任务区域具有较高的访问概率,后者保证具体任务执行者在任务结果数据回收阶段有较大的概率与组织者相遇^[34]。下面定义社交亲密度概念。

社交亲密度是量化移动社交网络环境中,不同参与用户基于时空维度上的共处现象而形成的一种亲近程度刻画,见式(3-18):

$$clos(w_x, w_y) = \frac{\sum_{i=1}^q fre(w_x, loc_i) \cap fre(w_y, loc_i)}{\sum_{i=1}^q fre(w_x, loc_i) + \sum_{i=1}^q fre(w_y, loc_i)} \times \frac{\sum_{i=1}^q dur(w_x, loc_i) \cap dur(w_y, loc_i)}{\sum_{i=1}^q dur(w_x, loc_i) + \sum_{i=1}^q dur(w_y, loc_i)} \quad (3-18)$$

其中, $\sum_{i=1}^q fre(w_x, loc_i) \cap fre(w_y, loc_i)$ 表示用户 w_x 与用户 w_y 共处的频次, $\sum_{i=1}^q dur(w_x, loc_i) \cap dur(w_y, loc_i)$ 表示用户 w_x 与用户 w_y 共处的时长。相应地,具体的执行者适应度的计算可表示为式(3-19):

$$fre(w_x, t_i) = \alpha \times act(w_x, loc_i) + (1 - \alpha) \times clos(w_x, w_y) \quad (3-19)$$

其中参数 α 表示对活跃度以及亲密度指标的折中加权。本文中, α 取值 0.5, w_y 表示社

群的组织者。

在群智感知任务执行阶段,若选定的执行者任务失败,则执行者可选择其将信息告知社群组织者,由组织者重新在当前社群中(剔除该执行者)按照上述流程选择新的任务执行者;或是在移动社交网络中选择亲密度最高的用户进行任务信息传递,由该亲密度最高的用户替补完成感知任务,并将任务信息移交给该亲密度最高者所在的社群。在数据回收阶段,任务执行者按照数据规模的大小选择 3G/4G 网络进行直接通信或机会式短距离通信。

3.3.4 实验结果及分析

本文以 WTD 公开数据集为实验数据,该数据集记录了加州大学圣地亚哥分校校园内 275 个携带 PDA(Personal Digital Assistant)设备的用户与部署的 AP 间约 11 周的通信数据。利用该数据集对所提出的面向移动社交网络的群智感知社群化任务分发方式进行性能验证。由于 WTD 数据集中不同用户数据的稀疏性,本次实验选择其中 140 个 AP 节点作为空间位置,选取其中 68 个用户作为参与用户。首先,对移动社群行为特征分布以及基于亲密度的用户连接网络进行实验测试,其中参数 k 取值为 9,具体结果分别如图 3-13 和图 3-14 所示。图 3-13 中的横轴与纵轴分别表示 AP 空间位置以及相应的参与用户。

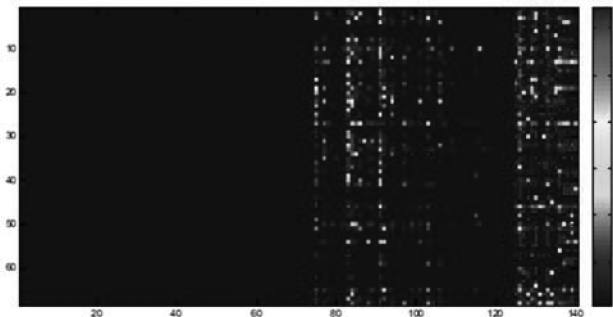


图 3-13 社群移动行为特征分布



图 3-14 基于亲密度的社交连接网络

其次,分别对任务执行效果、任务分发次数以及任务回收能耗进行测试,随机产生 100~400 个移动群智感知任务,感知任务的空间位置从 140 个 AP 点随机选取,时间间隔从 0:00—24:00 随机选取。相关参数的设置具体为:用户以概率 0.3 执行任务失败,社群组织者以概率 0.5 重新选择新的执行者,以概率值 0.5 通过社交网络传递任务;在传递任务的情况下,被传递者以其亲密程度为概率值执行任务或拒绝执行任务,在任务结果回收过程中,通过直接网络将数据传输能耗系数设置为 1,通过机会式网络将数据传输能耗设置为 0.4。由于现有任务分发工作缺乏对任务执行过程的干预,因此以传统任务分发模式为基础,构建基准算法(baseline algorithm)。流程阐述如下:基于用户的历史移动时空行为特征构建预测模型,以预测其在未来某个时段访问给定空间位置的概率,按照概率值的大小排序,选择 g 个用户作为任务的执行者($g=3$),在任务的执行过程中若发生执行失败情况,则平台按照上述概率值排序,依次选择新的用户作为感知任务的执行者,对于同一任务,若连续选择 3 次执行者均失败,则将该任务作为失效任务不再进行分发。图 3-15 显示了随机分发策略与传统的排序分发策略的比较实验,以及任务执行失败后不同的干预策略实验。其中,随机分发

策略与传统的排序分发策略的比较实验中未对失败任务进行用户的替补重选,如果选定的用户在第2天相应的时间间隔未出现在任务中指定的空间区域,则意味着该分发任务执行失败,由图3-15可以看出,本文提出的随机分发策略显著优于传统的随机分发策略,且分别在100~400个MCS任务情况下均性能稳定。任务执行失败后不同的干预策略实验中所列举的3种策略分别为本文提出的社群式分发-失败用户社群内替补、社群式分发-基于社交关系的亲密度用户传递、传统“平台-用户”的全部用户重新选取。由图3-15可知,社群式分发-基于社交关系的亲密度用户传递策略的任务完成率最高,其次为社群式分发-失败用户社群内替补策略以及传统“平台-用户”的全部用户重新选取策略。需要注意的是,由于WTD数据分布非常稀疏,极少数AP位置(如图书馆、宿舍等)集中了大量的位置数据,同时极少数超级活跃用户贡献了大量的数据,因此实验验证的任务完成率总体数值较低,但该值对不同算法的性能评测无影响。

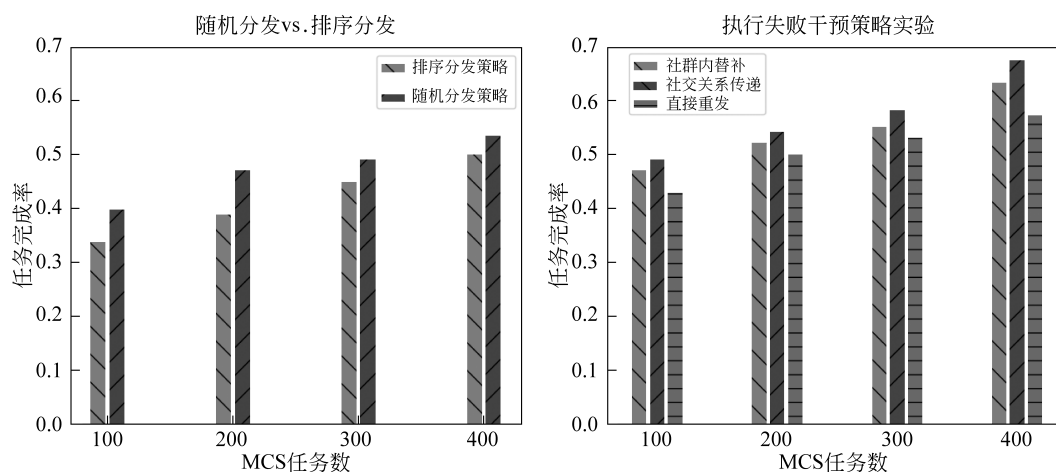


图 3-15 MCS 任务分发实验 1

图3-16所示为任务执行失败干预策略实验下任务的分发次数统计结果,感知数据回收能耗仿真实验结果。失败干预策略实验下任务的分发次数统计结果显示社群式分发-基于社交关系的亲密度用户传递分发次数最少,依次是社群式分发-失败用户社群内替补、传统“平台-用户”的全部用户重新选取。其原因在于,基于社交关系亲密度的用户传递分发具有较高的任务接收与执行率,在上述实验之外,作者同时进行了模拟真实校园环境的问卷调查统计,通过对97名在校生的统计显示,约89.73%的受访者在接到关系亲密的好友任务请求时会接受并完成该任务,这一数据与WTD实验的测试结果基本相符。而失败用户社群内替补优于传统“平台-用户”的全部用户重新选取的原因在于,基于时空移动行为的动态社群划分将具有相近移动行为的用户进行聚类划分,因此可以在相对较小的可选用户范围内搜寻合适的用户进行替补执行。感知数据回收能耗仿真实验结果显示失败用户社群内替补策略最优,依次为基于社交关系的亲密度用户传递分发、传统“平台-用户”的全部用户重新选取。其原因在于,社群内替补策略最优,在最大程度上利用机会式移动社交网进行短距离通信,因此能耗最少,而基于社交关系的亲密度用户传递分发次之,传统“平台-用户”的全部用户重新选取全部采用3G/4G通信方式进行数据上传/下载,其能耗最高。