

基于改进狮群进化算法的面向空间众包平台的多工作者多任务路径规划方法

赵 杨 倪志伟 朱旭辉 刘 浩 冉家敏

合肥工业大学管理学院 合肥 230009

过程优化与智能决策教育部重点实验室 合肥 230009

(729805176@qq.com)

摘 要 针对面向空间众包平台的多工作者多任务路径规划问题,以求解时间成本和路程成本最小的全局最优路径规划方案为目标,提出了基于改进狮群进化算法的路径规划方法。首先,结合现实问题场景,提出带有任务开始点和结束点的路径规划模型;其次,借鉴狮群进化算法的思想,改进狮群智能行为,引入驱逐行为,针对求解问题设计染色体编码方式、交叉、变异操作等,提出了面向空间众包平台的多工作者多任务路径规划的改进狮群进化算法;最后,运用改进狮群进化算法求解面向空间众包平台的多工作者多任务路径规划模型,并根据真实数据集制作问题算例进行测试。实验结果表明了算法的可用性和有效性。

关键词: 空间众包;路径规划;狮群进化算法

中图法分类号 TP182

Multi-worker and Multi-task Path Planning Based on Improved Lion Evolutionary Algorithm for Spatial Crowdsourcing Platform

ZHAO Yang, NI Zhi-wei, ZHU Xu-hui, LIU Hao and RAN Jia-min

School of Management, Hefei University of Technology, Hefei 230009, China

Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei 230009, China

Abstract In order to solve the problem of multi-worker and multi-task path planning for spatial crowdsourcing platform and aiming at solving the global optimal path planning scheme with the minimum time cost and distance cost, a path planning method based on the improved lion evolutionary algorithm is proposed. Firstly, a path planning model with task start and end points is proposed based on realistic problem scenarios. Secondly, by referring to the algorithm idea of lion evolutionary algorithm, the intelligent behavior of lions is improved, the expulsion behavior is introduced, and the chromosomal coding mode, crossover, mutation operation, etc. are designed for solving the problem. An improved lion evolutionary algorithm for multi-worker and multi-task path planning based on the spatial crowdsourcing platform is proposed. Finally, the improved lion evolutionary algorithm is used to solve the multi-worker and multi-task path planning model of the spatial crowdsourcing platform, and the problem is tested by making an example based on the real data set. The experimental results show the availability and effectiveness of the algorithm.

Keywords Spatial crowdsourcing, Path planning, Lion evolutionary algorithm

1 引言

随着移动互联网和线上到线下营销模式的快速发展,各种空间众包平台(如 Gmission、美团外卖、滴滴出行等)越来越受欢迎^[1],空间众包平台与人们日常生活的联系也越来越紧密^[2]。空间众包路径规划问题是空间众包的一项重要研究内容,在诸如拼车和送餐等实际应用程序中,空间众包平台在进行任务分配的本质就是在给工作者进行路径规划。现存研究通常采用任务规划模型对应用场景进行建模,旨在一段时间内为每位众包参与者分配多项众包任务并规划出执行这些任

务的详细顺序与路径,其典型应用场景为物流派送类服务,如百度外卖等^[3]。根据不同的分类标准,空间众包路径规划问题可以划分为以下不同的类别。

(1)根据面向对象的不同,可以分为面向空间众包平台、面向工作者、面向空间众包平台和工作者 3 种。其中 To 等^[4]面向空间众包平台,考虑一类众包问题,即工作者将位置信息发送到服务器,服务器分配给靠近工作者位置的任务,以最大化分配任务的总数量,Deng 等^[5]针对定义的最大任务调度问题,提出了精确算法和基于贪婪算法两种方法来解决该问题,从平台角度对空间众包任务进行规划。Zhao 等^[6]面向

基金项目:国家自然科学基金(91546108,71521001,71901001);安徽省科技重大专项项目(201903a05020020);安徽省自然科学基金项目(1908085QG298)

This work was supported by the National Natural Science Foundation of China (91546108,71521001,71901001), Science and Technology Major Special Projects of Anhui Province, China(201903a05020020) and Natural Science Foundation of Anhui Province, China(1908085QG298).

通信作者:倪志伟(zhwnelson@163.com)

工作者提出了一种空间众包动态任务规划算法,用时空邻近算法生成初始解,再利用禁忌搜索进一步提高解的质量。Nie等^[7]从全局优化的角度,提出了一种面向三方综合效益全局优化的在线任务分配算法。

(2)根据工作者数量的不同,可以分为单工作者多任务问题与多工作者多任务问题两种。在单工作者多任务路径规划中,大多数研究的目标都是在旅行预算约束下,为一名工人寻找一条可执行任务数量最大的路径^[8]。其中,Costa等^[9]假设工作者偏好走自己喜欢的道路,并愿意权衡旅行成本与收益,在静态场景下提出了一种绕路导向启发式方法来给单工作者推荐路径。Cui等^[10]提出了基于粒子群的在线粒子群路径规划算法,用于解决单工作者多任务情形下的动态路径规划问题。She等^[11]在多工作者多任务的静态场景下,提出了一个基于效用感知的社会事件参与计划问题,并针对该问题设计了一种基于贪心的启发式算法。Ma等^[12]在多工作者多任务的动态场景下,首次针对道路网络上共乘服务的动态路径进行研究。

(3)根据应用场景的不同,可以分为静态路径规划问题与动态路径规划问题两种。Meng等^[13]将众包的实际地理位置映射成地图结构,考虑众包任务起始位置、任务目标位置和快递员位置,提出的基于图形的空间任务分配方法可以获得分配路径。Tao等^[14]提出了在线场景下的多工人感知任务规划问题,为工人分配任务并制定计划,以使总效用最大化。

总体来说,现在的研究大多以单目标优化为主,从空间众包平台或者工作者的角度出发,目标在于提高任务分配的总效用或减少差旅成本等,多目标优化的研究相对较少。单工作者多任务问题和多工作者多任务问题的研究也会考虑是否在线规划,其中,多工作者多任务问题的研究更具挑战性并且更贴近实际应用场景。在静态路径规划问题方面,很多研究提出了基于贪婪策略的求解方法以提高求解效率,但是几乎所有基于贪婪策略的解决方案都没有理论保证其有效性^[8]。在动态路径规划问题方面,现有研究提出的解决方案存在效率较低,考虑多目标优化时目标冲突等问题。在使用启发式算法解决空间众包路径规划问题时,需要改善算法陷入局部最优的情况,提高算法的运算效率,从而与实际问题相匹配。因此,本文针对面向空间众包平台的多工作者多任务路径规划问题(Multi-worker and Multi-task Path Planning for Spatial Crowdsourcing Platform, MMPP-SCP),首先,考虑现实中工作者和任务的匹配标准,根据工作者与任务的时空信息,构建 MMPP-SCP 模型;在此基础上提出改进狮群进化算法(Improved Lion Evolutionary Algorithm),并用其来求解出工作者的无效行驶路程(工作者前往任务开始点和任务结束点的距离)和时间惩罚成本最小的全局路径规划方案。

随着近年来移动网络和移动设备的快速发展并被广泛应用到人们的日常生活中,空间众包已经成为一个很有前景的研究领域。本文使用纽约市 2016 年 1 月的出租车出行数据,对 MMPP-SCP 改进狮群进化算法(MMPP-SCP-LEA)进行测试,实验结果表明 MMPP-SCP-LEA 在本文设计的问题场景下具有良好的效果,尤其在复杂问题场景下具有很好的效果。

2 问题分析与建模

2.1 问题定义

假设工作者以单位速度行驶,工作者的行驶速度恒定,工作者在闲置状态下会进入空间众包平台等待派单,并发布自己的空间位置与偏好的工作的开始时间和结束时间。发布在空间众包平台上的任务需要工作者前往特定地点完成^[15],任务包含任务的开始点与结束点,工作者需要先到达任务开始点再到达任务结束点才完成该任务。另外,在低峰期场景下,任务数小于工作者数时,由空间众包平台优先给先进入平台的工作者进行路径规划,选取的工作者数等于任务数。而在高峰期场景下,任务数大于工作者数时,工作者可以接取多个任务。

定义 1(空间众包平台) 空间众包平台在一定的时间阈值下便会获取任务集合和工作者集合,并且根据任务和工作者到达空间众包平台上的时间来排序。例如在一次路径规划中, A 工作者第一个进入空间众包平台等待派单,那么 A 工作者便会被记为 1 号工作者。假设平台在计算工作者任务收益时,是根据工作者的有效服务路程计算的,并且任务收益与服务路程成正比。平台一次路径规划结束后,在一定的时间阈值下会再次获取任务集合和工作者集合进行下一次路径规划。

定义 2(空间众包任务) 空间众包任务指的是具有时空属性的一类任务^[16],用四元组 $\langle Ts, Td, Te, Tl \rangle$ 表示该空间众包任务。其中, Ts 表示任务开始点; Td 表示任务结束点; Te 表示任务开始时间; Tl 表示任务结束时间。任务集合记为 $N = \{1, 2, 3, \dots, i, \dots, n\}$ 。

定义 3(空间众包工作者) 空间众包工作者指的是具备移动设备并且愿意执行空间众包任务的工人^[17],用四元组 $\langle Ws, Wd, We, Wl \rangle$ 表示。其中, Ws 表示工作开始点; Wd 表示工作结束点; We 表示工作开始时间; Wl 表示工作结束时间。工作者集合记为 $M = \{1, 2, 3, \dots, j, \dots, m\}$ 。

2.2 问题复杂性

本节分析所研究的 MMPP-SCP 离线版问题的复杂性。在离线版本中,空间众包任务数、空间众包工作者数、空间众包任务位置、空间众包工作者位置已知,其中,任务与工作者的位置信息与时间信息一一对应。由于三维匹配问题可以规约到该问题,可证明该问题是 NP-hard 问题^[18]。

定理 1 MMPP-SCP 离线版问题是 NP-hard 问题。

证明:考虑 MMPP-SCP 离线版问题的一种特殊情况:空间众包任务数、空间众包工作者数、空间众包任务开始点数相等,工作者和任务开始点容量为 1。在这种情形下,该问题等价于三维匹配问题的优化问题,因为此问题已经被证明是 NP 完全问题,进而 MMPP-SCP 离线版问题是 NP-hard 问题。

因此,MMPP-SCP 离线版问题是 NP-hard 问题,进而 MMPP-SCP 属于 NP-hard 问题。

2.3 面向空间众包平台的多工作者多任务路径规划模型

2.3.1 符号说明

(1) C :工作者行驶单位距离的成本。

(2) q_i : i 号任务的需求量, $i \in N$ 。

(3) q_k : k 号任务的需求量, $k \in N$ 。

(4) Q : 工作者的额定任务容量。

(5) R : 工作者接取的任务集合。

(6) $speed$: 工作者在工作过程中的行驶速度。

(7) S : 工作者非空闲状态下的决策变量。

$$S = \begin{cases} 1, & \text{接取 } i \text{ 号任务后去往 } k \text{ 号任务开始点,} \\ & i, k \in N, k \neq i \\ 0, & \text{完成 } i \text{ 号任务后去往 } k \text{ 号任务结束点,} \\ & i, k \in N, k \neq i \end{cases}$$

(8) ds_{ji} : j 号工作者从其工作开始点 W_{s_j} 到 i 号任务开始点 T_{s_i} 的距离。

(9) dm_{jik} : j 号工作者从当前任务结束点 Td_i 或任务开始点 T_{s_i} 到下一个任务开始点 T_{s_k} ($k \in N, k \neq i$) 的距离。

(10) de_{ji} : j 号工作者从其最后一个任务结束点 Td_i 到其工作结束点 Wd_j 的距离。

(11) de_{jk} : j 号工作者从其最后一个任务结束点 Td_k 到其工作结束点 Wd_k 的距离。

(12) dt_{ii} : 工作者从 i 号任务开始点 T_{s_i} 到任务结束点 Td_i 的距离。

(13) ts_i : 工作者到达 i 号任务开始点的时间, $i \in N$ 。

(14) ts_k : 工作者到达 k 号任务开始点的时间, $k \in N$ 。

(15) te_j : j 号工作者到达工作结束点的时间, $j \in M$ 。

(16) Te_i : i 号任务的任务开始时间, $i \in N$ 。

(17) W_{ij} : j 号工作者的工作结束时间, $j \in M$ 。

(18) $pt_i(ts_i)$: 工作者到达 i 号任务开始点时产生的时间惩罚成本。

(19) $pw_j(te_j)$: j 号工作者到达工作结束点时产生的时间惩罚成本。

(20) T_c : 客户的忍耐时间。

(21) pt : 在客户忍耐时间内, 施加的时间惩罚量。

(22) pc : 超过了客户忍耐时间, 增加的额外时间惩罚。

(23) x_{ji} : j 号工作者是否最先接取了 i 号任务的决策变量。

$$x_{ji} = \begin{cases} 1, & \text{第 } j \text{ 个工作者最先接取了 } i \text{ 号任务, } i \in N \\ 0, & \text{否则} \end{cases}$$

(24) y_{jik} : j 号工作者是否接取或完成 i 号任务后再次接取新的任务的决策变量。

$$y_{jik} = \begin{cases} 1, & \text{第 } j \text{ 个工作者在开始或完成 } i \text{ 号任务} \\ & \text{后前往 } k \text{ 号任务开始点, } i, k \in N, k \neq i \\ 0, & \text{否则} \end{cases}$$

(25) z_{ji} : j 号工作者是否最后接取了 i 号任务的决策变量。

$$z_{ji} = \begin{cases} 1, & \text{第 } j \text{ 个工作者最后接取了 } i \text{ 号任务, } i \in N \\ 0, & \text{否则} \end{cases}$$

2.3.2 模型构建

基于以上的假设条件和参数定义, 建立 MMPP-SCP 模型。由于任务收益与工作者的有效服务路程成正比, 产生正向收益的路程为工作者从任务开始点到任务结束点的路程, 因此优化目标为工作者的无效行驶路程(工作者前往任务开始点和任务结束点的距离)和时间惩罚成本最小。

$$\min C(\sum_{j \in M} \sum_{i \in N} x_{ji} ds_{ji} + \sum_{j \in M} \sum_{i \in N} \sum_{k \in N, k \neq i} y_{jik} dm_{jik} +$$

$$\sum_{j \in M} \sum_{i \in N} z_{ji} de_{ji}) + \sum_{j \in M} \sum_{i \in N} pt_i(ts_i) + \sum_{j \in M} pw_j(te_j) \quad (1)$$

$$pt_i(ts_i) = \begin{cases} \frac{(Te_i - ts_i)}{T_c} pt, & ts_i \leq Te_i \\ \frac{(ts_i - Te_i)}{T_c} pt, & Te_i < ts_i < Te_i + T_c \\ pt + pc, & ts_i > Te_i + T_c \end{cases} \quad (2)$$

$$pw_j(te_j) = \begin{cases} \frac{(Wl_j - te_j)}{T_c} pt, & te_j \leq Wl_j \\ \frac{(te_j - Wl_j)}{T_c} pt, & Wl_j < te_j < Wl_j + T_c \\ pt + pc, & te_j > Wl_j + T_c \end{cases} \quad (3)$$

式(1)表示目标函数为最小化工作者在工作期间的无效行驶路程和时间惩罚成本之和; 式(2)表示工作者到达任务开始点时所产生的时间惩罚成本; 式(3)表示工作者到达工作结束点时所产生的时间惩罚成本。

$$x_{ji}, y_{jik}, z_{ji} \in \{0, 1\}, i, k \in N, j \in M, k \neq i \quad (4)$$

$$\sum_{i \in N} x_{ji} = \sum_{j \in M} x_{ji} = \sum_{i \in N, k \neq i} y_{jik} = \sum_{k \in N, k \neq i} y_{jik} = \sum_{i \in N} z_{ji} = \sum_{j \in M} z_{ji} = 1 \quad (5)$$

$$\forall i, k \in N, \forall j \in M, k \neq i \quad (6)$$

$$(\sum_{i \in N} x_{ji} q_i + \sum_{i \in N} \sum_{i \in N, k \neq i} y_{jik} q_k S) \leq Q \quad (7)$$

若 j 号工作者依次接取了 i, k 号任务, 且 ts_k 未知, 则:

$$ts_k = \frac{ds_{ji}}{speed} + \frac{dt_{ii}}{speed} f + \frac{dm_{jik}}{speed} \quad (8)$$

若 j 号工作者最后接取了 k 号任务, 且 te_j 未知, 则:

$$te_j = \frac{ds_{ji}}{speed} + \sum_{i, k \in N, k \neq i} \frac{dm_{jik}}{speed} + \sum_{i \in R} \frac{dt_{ii}}{speed} + \frac{de_{jk}}{speed} \quad (9)$$

式(4)表示 x_{ji}, y_{jik}, z_{ji} 为 0-1 变量; 式(5)表示一项任务只由一位工作者完成(但工作者可完成多项任务); 式(6)表示在工作者处于空闲状态前, 工作者接取的任务量不能超过额定的任务容量; 式(7)为工作者到达 k 号任务开始点的时间的计算公式; 式(8)为工作者到达工作结束点的时间的计算公式。

3 MMPP-SCP-LEA

通过本文 2.2 节可知, MMPP-SCP 属于 NP-hard 问题。遗传算法因具有较强的全局搜索能力和良好的扩展性, 被广泛应用于复杂问题的优化求解, 包括车辆路径规划领域, 但是遗传算法对个体局部搜索能力较弱^[19], 而狮群进化算法相比经典遗传算法具备更多的进化机制, 所以本文借鉴狮群进化算法^[20]的思想, 并改进狮群进化算法的智能行为, 进一步提高算法的局部搜索能力, 从而设计了一种改进狮群进化算法对上述 MMPP-SCP 模型进行求解。

3.1 MMPP-SCP-LEA 染色体编码设计

鉴于 MMPP-SCP 是基于次序的组合优化问题^[21], 本文采用自然数编码方式对个体染色体进行编码。对于 MMPP-SCP, 一位工作者的一次工作周期是从工作开始点出发, 并依次访问任务开始点与任务结束点后, 最后到达工作结束点, 本次工作周期结束, 因此将工作者的工作开始点和任务结束点映射为 0, 将空间众包任务分解为前往任务开始点 T_{s_i} 和前往任务结束点 Td_i 两个动作, 进而将两个动作定义为两个任务(映射到基因编码可表示为 1, 2 或 3, 4 等此类连贯的两个正整数, 并且前奇后偶)。假定在空间众包平台的一次路径规划中, 获取的任务集合和工作者集合分别为 $N = \{1, 2, 3, \dots, i, \dots, n\}$, $M = \{1, 2, 3, \dots, j, \dots, m\}$ (工作者的编号顺序不能改

变,染色体中第一对0即表示1号工作者的工作开始点与工作结束点,依次类推),可知染色体共有 $2m+2n$ 个基因位,且包括 $2m$ 个0与 $2n$ 个互不重叠的正整数。若一条染色体为 $[0,1,2,5,6,0,0,3,4,0,0,7,8,0]$,则表示推荐1号工作者依次完成1号和3号任务,2号工作者完成2号任务,3号工作者完成4号任务,编码案例如图1所示。



图1 染色体编码方式示意图

Fig. 1 Schematic diagram of chromosome coding

3.2 MMPP-SCP-LEA 智能行为

本文针对狮群进化算法中的智能行为进行改进,建立多个狮群,加入驱逐行为,流浪狮群的个体从其他狮群中抽取,更符合自然界中流浪狮群的来源方式,有利于避免算法陷入局部最优。同时,为了避免群体的同质化现象严重,导致算法早熟和陷入局部最优,且为了保持种群的多样性,在选取雌狮时,优先选取与雄狮差异度高的个体,并设计了与之对应的差异度计算公式。

3.2.1 领地争夺与领地接管行为

与其他猫科动物不同,狮子的社会行为导致了领地狮会在幼崽成长的时间里保卫自己的领地,而流浪狮由于没有领地会通过领地争夺与领地狮进行竞争,从而获取领地和交配权。如果将狮子个体看作一个解决方案,那么领地争夺行为是对现有解决方案(领地狮)和新生成的解决方案(流浪狮)进行项评估的过程,如果新解决方案优于现有的解决方案,则用新解决方案替代现有解决方案^[20]。改进狮群进化算法的领地争夺与领地接管行为的伪代码如算法1所示。

算法1 改进狮群进化算法的领地争夺与领地接管行为

输入:(Pr,malelions,nomads,cubs)

输出:(malelions)

1. $P_s \leftarrow$ 生成一个0到1之间的随机数;
2. if $P_s \leq Pr$ then
3. challenge = [];
4. 使流浪狮群的个体随机排序;
5. challenge $\leftarrow$ nomads[0];
6. $k \leftarrow$ 随机生成一个索引值(不大于雄狮群的个数);
7. if $f(\text{challenge}) > f(\text{malelions}[k])$ then
8. malelions[k] $\leftarrow$ challenge;
9. 将该流浪狮个体从流浪狮群中剔除;
10. end if
11. else
12. 使幼狮群的个体按适应度值从大到小排序;
13. inheritlion = [];
14. inheritlion $\leftarrow$ cubs[0];
15. $k \leftarrow$ 随机生成一个索引值(不大于雄狮群的个数);
16. if $f(\text{challenge}) > f(\text{malelions}[k])$ then
17. malelions[k] $\leftarrow$ inheritlion;
18. end if
19. end if

其中 malelions,nomads,cubs 分别为雄狮集合、流浪狮集

合和幼狮集合;Pr为触发领地争夺行为的概率;Ps为一个范围在0至1间的随机数;另外用 challenge 表示参与领地争夺行为的流浪雄狮;inheritlion 表示参与领地接管行为的领地雄狮; $f(x)$ 为适应度函数, $f(x)$ 值越大代表个体越适应环境。本文以上述伪代码详细地描述了改进狮群进化算法的领地争夺与领地接管行为,首先,当 $Pr > P_s$ 时,触发领地争夺行为,从流浪狮群中选取一头雄狮与领地狮群中的一头狮王进行比较,如果流浪雄狮的适应度值大于领地狮王,则由该流浪雄狮替换领地狮王,反之则不替换。其次,当 $Pr \leq P_s$ 时,触发领地接管行为,从幼狮群中选取一头雄狮与领地狮群中的一头狮王进行比较,如果幼雄狮的适应度值大于领地狮王,则由该幼雄狮替换领地狮王,反之则不替换。

3.2.2 雄狮驱逐行为

参考自然界中幼狮群中的雄狮会在发育成年时被驱逐出狮群,将领地狮群中的部分幼狮个体驱逐出狮群,由流浪狮群接纳这些被驱逐出的个体,避免领地狮群同质化严重,防止算法早熟收敛。另外,在流浪狮群内部引入竞争机制,随机挑选流浪狮群中的两个个体,比较二者的适应度值大小,小的一方将植入大的一方的部分基因^[22]。这样一方面可以增加流浪狮群的多样性;另一方面适应度值高的流浪狮通过领地争夺行为会及时替换掉适应度值低的领地狮,可以使算法尽快跳出局部最优。

3.2.3 差异度计算公式

在改进狮群进化算法中,由于子代会更多地遗传狮王的基因,因此引入了差异度计算公式,对雌狮进行筛选,以保持种群的多样性。设染色体集合为 $Ga = \{G1, G2, G3, \dots, Ga, \dots, Gu\}$,Ga 和 Gb 的差异度为 $\delta(Ga, Gb)$, Ga_i 表示 Ga 染色体第 i 个基因位上的值。差异度计算公式如下:

$$\delta(Ga, Gb) = \frac{1}{2m} \sum_{a=1}^{2m} \|Ga_i - Gb_i\| \quad (9)$$

$$\|Ga_i - Gb_i\| = \begin{cases} 1, & Ga_i \neq Gb_i \\ 0, & Ga_i = Gb_i \end{cases}$$

3.3 针对染色体编码方式设计保序双点交叉操作

基于传统的双点交叉方式,考虑到 MMPP-SCP 中工作者与工作任务的配对有序特征,本文设计了保序双点交叉操作。该交叉方式根据两条父代染色体,先在一条染色体中随机选取一段基因片段,以0开始并以0结束(代表第 n 位工作者的一次工作周期)。同理,在另一条父代染色体上随机选取一段染色体片段,两条子代染色体直接遗传所对应的父代染色体中被选取的部分,再原序遗传异父代染色体中未被选取的基因,从而形成两条子代染色体。保序双点交叉操作示意图如图2所示。

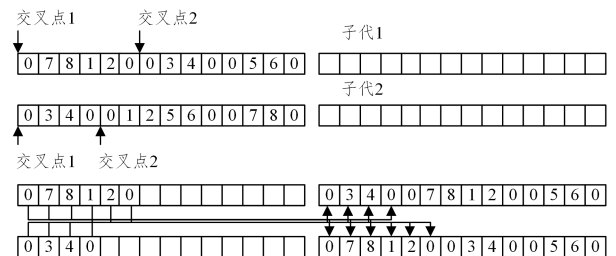


图2 保序双点交叉操作示意图

Fig. 2 Schematic diagram of sequence preserving double point crossover operation

3.4 针对染色体编码方式设计保序双点变异操作

基于传统的双点变异方式,考虑到 MMPP-SCP 中工作者与工作任务的配对有序特征,本文设计了保序双点变异操作。该变异方式根据原染色体,先从中随机选取两位连续非零基因(代表工作者的一项任务),再从中选取与之前不同的两位连续非零基因,将选取的两段染色体片段交换位置,并将原染色体中未选取的部分直接遗传到新染色体中,从而形成一条新染色体。保序双点变异操作示意图如图 3 所示。

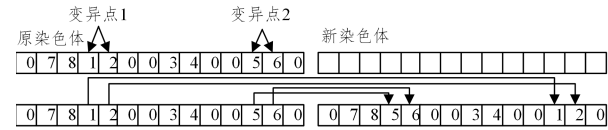


图 3 保序双点变异操作示意图

Fig. 3 Schematic diagram of sequence preserving double point mutation operation

3.5 算法步骤

本节给出 MMPP-SCP-LEA 算法的具体步骤。

Step 1 参数初始化及染色体编码:对领地狮群和流浪狮群进行初始化,初始化领地狮群规模为 U 、种群迭代次数为 V 、驱逐率为 β ,并且将领地狮群分为 F 个群落和对应的群落领地,在每个群落中,将比例 μ 的个体作为雄狮,其余为雌狮。随机生成 λ 个个体作为流浪狮群。各个群落具备自己的狮王,繁衍出来的雄狮会通过领地接管行为接管领地或者被驱逐加入流浪狮群,各个群落都面临着同一流浪狮群中流浪狮的入侵威胁。根据 3.1 节设计的染色体编码方式,生成 $U + \lambda$ 条染色体,对应领地狮群和流浪狮群的个体,每个狮群个体对应一条染色体,设置当前迭代次数为 $L = 0$ 。

Step 2 驱逐行为:计算幼狮群中所有个体的适应度值,并按照适应度值从大到小的顺序对染色体进行排序,将前 β 的个体作为成熟的雄狮驱逐出群落,加入流浪狮群。

Step 3 领地争夺和领地接管:随机选取 $\frac{\mu U}{F}$ 个狮王参与领地争夺和领地接管行为,设置当流浪狮入侵领地狮群时,触发领地争夺行为的概率为 Pr ,并产生一个范围在 0 至 1 间的随机数 Ps 。

Step 3.1 若 $Pr > Ps$,触发领地争夺行为,从流浪狮群中随机选取一个个体,比较其与被选取狮王的适应度值大小,如果该流浪狮适应度值大于被选取的狮王,则领地争夺成功,由这只流浪狮替换该狮王,否则不做改变。

Step 3.2 若 $Pr \leq Ps$,触发领地接管行为,将幼狮群个体按适应度值大小从大到小排序,从幼狮群中选取适应度值最大的个体,比较其与被选取狮王的适应度值大小,如果该幼狮适应度值大于或等于被选取的狮王,则领地接管成功,由这只幼狮替换该狮王,否则不做改变。

Step 4 狮群繁衍:为了避免群体的同质化现象严重,导致算法早熟和陷入局部最优,在选取雌狮时,优先选取与狮王差异度高的个体,将待选雌狮按照差异度从大到小的顺序排序,选取前 μU 个个体后,再随机从待选雌狮群中选取雌狮,直到满足雌狮个数。在狮群繁衍行为中,所有狮王都有繁衍机会,与其交配的雌狮从雌狮群中随机选取,二者进行交叉操作产生幼狮群,接着幼狮群再进行变异操作。

Step 5 狮王竞争:计算领地狮群中所有个体的适应度

值,并按照适应度值从大到小的顺序对染色体进行排序,选取前 μ 的个体作为狮王,狮王个数为 μU 。

Step 6 更新狮群:随机挑选流浪狮群中的两个个体,比较二者的适应度值大小,小的一方将植入大的一方的部分基因,将流浪狮群中的狮群个体按照适应度从大到小的顺序排序,保留前 λ 个狮群个体构成流浪狮群。将幼狮群加入领地狮群中,将领地狮群中的狮群个体按照适应度从大到小的顺序排序,保留前 U 个狮群个体构成领地狮群,并且领地狮群由 F 个群落组成,每个群落有自己的领地,每个群落中的狮王个数为 $\frac{\mu U}{F}$,其余为雌狮,令 $L = L + 1$ 。

Step 7 判断是否满足 $L \geq V$,若不满足,循环 Step 2 - Step 6;否则,算法终止,计算并选取种群中适应度值最大的狮群个体作为 MMPP-SCP 的最佳路径规划方案。

改进狮群进化算法流程图如图 4 所示。

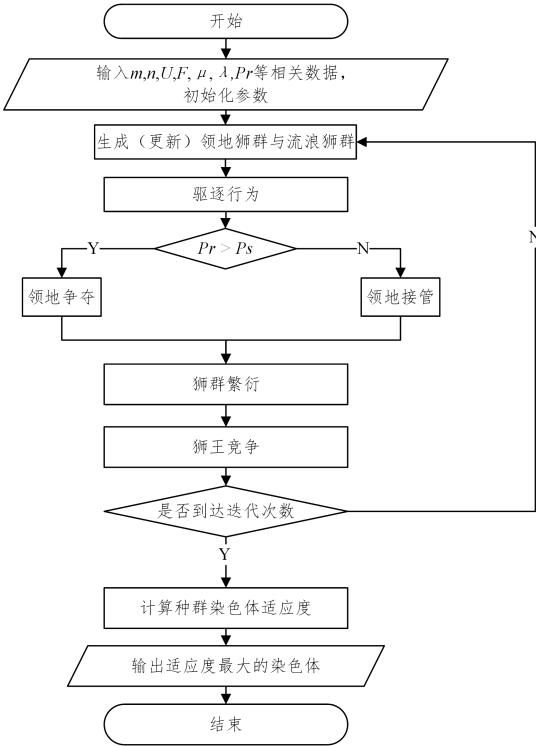


图 4 改进狮群进化算法流程图

Fig. 4 Flow chart of improved lion evolutionary algorithm

4 实验及结果分析

4.1 实验数据与参数设置

数据集:本文采用纽约市 2016 年 1 月的出租车出行数据,该数据集中每一个元组包含任务的起止位置、发布时间以及终止时间。由于该数据集中没有工作者的起止位置和工作起止时间数据,所以在此基础上增添了工作者信息,再从数据集中随机抽取 48 条数据,以形成 MMPP-SCP 算例,算例包含 24 条任务的时空信息与 24 条工作者时空信息。其中,任务的时空信息包括任务起止位置、任务开始时间与结束时间,工作者时空信息是从数据集中未使用到的任务信息中随机抽取的,包括工作起止位置、工作开始时间与结束时间。另外, $speed$ 记为 40 km/h; Tc 记为 300 s; pt 记为 2; pc 记为 3; C 记为 1; i 号任务的需求量默认为 1;工作者的额定任务容量 Q 不限。

参与实验对比的算法如下。

MMPP-SCP-LEH:最小过期时间启发式算法(Least Expiration Time Heuristic),该算法的基本思路是贪婪地选择截止时间最小的任务。

MMPP-SCP-EDH:最早期限启发式算法(Earliest Deadline Heuristic),该算法的基本思路是贪婪地选择最先到达空间众包平台的任务,即先到先服务。

MMPP-SCP-NNH:最近邻启发式算法(Nearest Neighbor Heuristic),该算法的基本思路是贪婪地选择离工作者最近的空间众包任务。

MMPP-SCP-GA:选用遗传算法作为一个基准比较算法,用来判断本文改进的算法性能。

MMPP-SCP-MGA^[21]:由于微生物遗传算法中竞争机制与狮群算法中的领地争夺和领地接管等智能行为相似,将其引入判断算法针对智能行为的改进设计是否有效。

MMPP-SCP-LEA:本文针对面向空间众包平台的多工作者多任务路径规划问题提出的改进狮群进化算法。

实验所涉及的代码由 JetBrains PyCharm Community 软件编写,编译运行的计算机参数为:64 位 Windows10 操作系统,Intel(R) Core(TM) i7-4510 CPU @ 2.00GHz 2.60GHz, 8.00GB。

改进狮群算法的参数设置如表 1 所列。

表 1 改进狮群进化算法的参数设置

Table 1 Parameter setting of improved lion evolutionary algorithm

参数	数值
种群规模 U	40
迭代次数 V	200
交叉概率 $c p$	1.0
变异概率 $v p$	0.2
触发领地争夺的概率 $P r$	0.8
驱逐率 β	0.5
初始雄狮比例 μ	0.1
初始流浪狮个体 λ	4
群落个数 F	4

4.2 MMPP-SCP-LEA 性能测试实验及结果分析

首先,在低峰期场景下,将工作者数设置为 24 人不变,在任务数量变动时,6 种算法的表现如图 5 所示。其次,在高峰期场景下,将任务数设置为 24 个不变,在工作者数量变动时,6 种算法的表现如图 6 所示。由图 5 和图 6 可知,相比其他算法,MMPP-SCP-LEA 花费的路程成本和时间成本最小,并且明显优于 3 种启发式算法,证明 MMPP-SCP-LEA 从全局最优的角度寻找成本最小的路径规划方案是有效的,在高低峰期场景下都有不错的效果。最后,MMPP-SCP-GA, MMPP-SCP-MGA 和 MMPP-SCP-LEA 分别针对不同的问题场景进行 10 次实验,运行结果对比分析如表 2 所列。

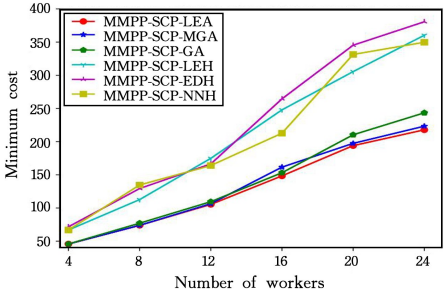


图 5 不同任务数量下的算法性能

Fig. 5 Algorithm performance under different number of tasks

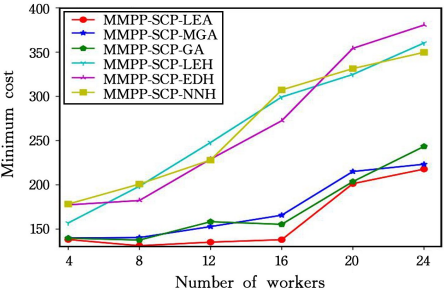


图 6 不同工作者数量下的算法性能

Fig. 6 Algorithm performance under different number of workers

表 2 3 种算法在不同问题场景下的对比结果

Table 2 Comparison results of three algorithms in different problem scenarios

问题场景	方法	最优值	最劣值	平均值	平均运行时间
8 tasks 8 workers	MMPP-SCP-GA	74.904718	94.780364	84.842541	3.538879
	MMPP-SCP-MGA	75.052999	94.692157	84.872578	5.972481
	MMPP-SCP-LEA	74.796433	91.258438	83.027435	6.442520
16 tasks 4 workers	MMPP-SCP-GA	79.544980	99.895523	89.720252	4.053751
	MMPP-SCP-MGA	78.915487	96.387039	87.651263	7.347557
	MMPP-SCP-LEA	78.524215	93.957636	86.240926	7.935125
16 tasks 8 workers	MMPP-SCP-GA	94.274393	138.802222	116.538308	4.779567
	MMPP-SCP-MGA	93.719513	136.330583	115.025048	8.758495
	MMPP-SCP-LEA	92.175541	131.688048	111.931794	9.447495
16 tasks 16 workers	MMPP-SCP-GA	157.352577	211.757932	184.555255	6.317085
	MMPP-SCP-MGA	154.081088	208.262770	181.171929	11.523011
	MMPP-SCP-LEA	145.687821	203.847508	174.767664	13.123799
24 tasks 8 workers	MMPP-SCP-GA	136.922188	178.788916	157.855552	6.272606
	MMPP-SCP-MGA	135.807456	182.205381	159.006419	11.448150
	MMPP-SCP-LEA	133.424770	176.914832	155.169801	15.221800
24 tasks 24 workers	MMPP-SCP-GA	233.599769	329.137687	281.368728	9.342755
	MMPP-SCP-MGA	228.904199	332.666879	280.785539	16.991482
	MMPP-SCP-LEA	217.123510	317.200275	267.161893	18.371602

图 7 为 3 种方法在 6 种不同问题场景下的求解曲线。图 8 为 8 任务 8 工作者场景下工作者接送任务图。

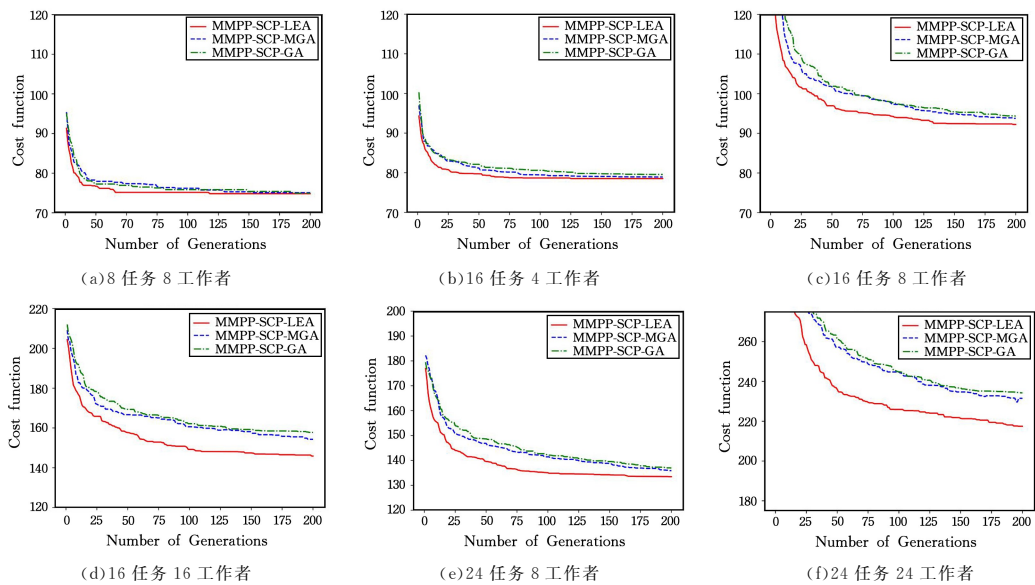
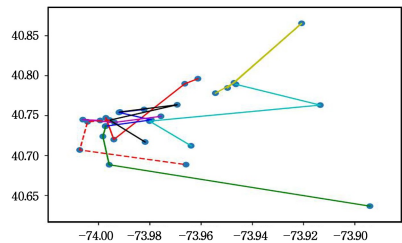


图 7 3 种方法在 6 种不同问题场景下的求解曲线

Fig. 7 Solution curves of three methods in six different problem scenarios



注：横纵坐标分别为经纬度

图 8 8 任务 8 工作者场景下工作者路径规划图

Fig. 8 Path planning diagram of 8 tasks 8 workers in worker scenario

对比分析表 2 和图 7, 可以得出的结论如下: (1) 从算法的收敛速度来看, MMPP-SCP-LEA 明显快于另外两种算法; 在求解效果方面, 从表 2 可以看出, 在 5 种不同的问题场景下, MMPP-SCP-LEA 的最优值、最劣值、平均值均优于另外两种算法。 (2) 从不同问题场景下算法的表现来看, MMPP-SCP-LEA 较另外两种算法更适合用于解决复杂的问题, 例如在 8 任务 8 工作者的问题场景下, 3 种算法之间的差距较小, 而在 24 任务 8 工作者的问题场景下, MMPP-SCP-LEA 的求解效果相较于另外两种算法提升较大。 (3) 从算法的平均运行时间来看, MMPP-SCP-LEA 的运行速度是慢于另外两种算法的, 这是由于算法内置的智能行为中存在一些判断操作,

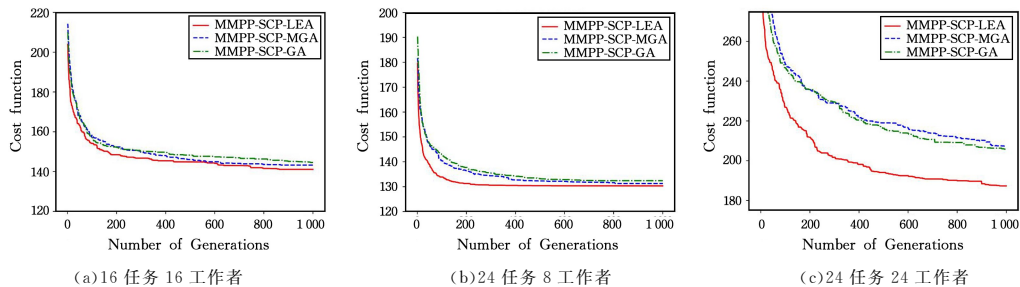


图 9 3 种方法在 3 种不同问题场景下的求解曲线

Fig. 9 Solution curves of three methods in three different problem scenarios

4.4 MMCP-SCP-LEA 参数分析

经过 4.2 节对 MMCP-SCP-LEA 的性能测试分析和 4.3

但是在实际应用场景下, 算法运行时间是在可接受范围内, 在设置的最复杂的问题场景下也在 18 s 左右。

4.3 MMCP-SCP-LEA 收敛测试实验及结果分析

为了进一步分析 MMCP-SCP-LEA 的效率和收敛性情况, 将迭代次数调整为 1 000 次, 结合 4.2 节可知 MMPP-SCP-LEA 在 8 任务 8 工作者、16 任务 4 工作者以及 16 任务 8 工作者的问题场景下 200 代内便基本达到收敛; MMPP-SCP-LEA 在另外 3 种复杂问题场景下的性能比较显著, 从求解曲线来看 200 代内未达到收敛状态, 算法仍有进一步寻优的能力。因此, 分别针对 16 任务 16 工作者、24 任务 8 工作者、24 任务 24 工作者共 3 种问题场景进行 10 次实验, 图 9 为 3 种方法在 3 种不同问题场景下的迭代曲线。由图 9(a) 和图 9(b) 可知, 在 16 任务 16 工作者和 24 任务 8 工作者的问题场景下, MMPP-SCP-LEA 的收敛速度明显高于另外两种算法, 并且在 200 代左右基本收敛, 随后达到平稳, 这说明 MMPP-SCP-LEA 在前期具有较好的搜索寻优能力, 而且在后期能保留住优质的种群基因。由图 9(c) 可知, 在 24 任务 24 工作者的问题场景下, 由于问题场景设计比较复杂, 可行解具有更多的排列组合情况, 而 MMPP-SCP-LEA 更适合用于解决复杂问题, 所以在迭代 1 000 次的情况下, MMPP-SCP-LEA 显著优于另外两种算法, 在 1 000 代左右基本收敛。

节对 MMCP-SCP-LEA 的收敛测试分析, 可知通过借鉴狮群进化算法的思想, 设计改进的智能行为能帮助算法较快跳出

局部最优点,并且算法具有比较好的持续寻优能力。在此基础上,本节对 MMCP-SCP-LEA 中影响 MMCP-SCP-LEA 的其他参数进行分析。问题场景设置为 24 任务 24 工作者,在保持原参数设置的基础上,主要对种群规模 U 、触发领地争夺或领地接管的概率 Pr 、变异概率 vp 、群落个数 F 进行变动及分析。各算法参数对算法性能的影响如图 10 所示。

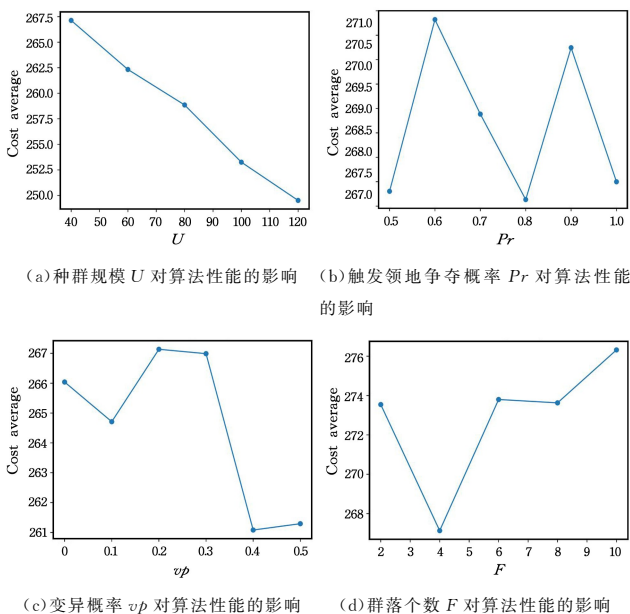


图 10 各算法参数对算法性能的影响

Fig. 10 Influence of each algorithm parameters on algorithm performance

种群规模 U : 种群规模决定了解空间中可行解的数量,通常分布在解空间中的可行解数量越多,越有可能得到最优解。但是解空间中的可行解数量越多,算法的迭代时间也就越长,算法的迭代效率就越低。

迭代次数 V : 迭代次数的选择要看针对的问题场景,若问题场景对算法时效性要求较高并且场景较为简单,建议采用 200 次;若问题场景对算法时效性要求不高或者希望求解的精度较高,建议采用 1000 次。

触发领地争夺的概率 Pr : 领地争夺行为本质上是对现有可行解和新生成的可行解进行评估的过程,如果新的可行解优于现有的可行解,则由新的可行解代替现有的可行解。由图 10(b)可知,在保持原参数设置的基础上,触发领地争夺的概率 Pr 设为 0.8 较优,触发领地争夺的概率 Pr 对平均成本影响不大,但 Pr 不宜过小,过小会导致算法跳出局部最优的能力较差。

变异概率 vp : 种群中的狮王作为最优可行解的代表,决定了未来种群的进化方向,并且狮王的基因会更多地被后代所遗传,如果变异概率设置得较低,会导致种群中的狮王在多次迭代过程中未被替换,同质化的个体增多,算法易于陷入局部最优。由图 10(c)可知,MMCP-SCP-LEA 的变异概率 vp 设置得较高比较好,但是变异概率过高,运行较多的变异操作会导致算法的迭代时间变长,也有可能使本来优质的基因变异成为较差的基因,所以在实际应用中需要综合考虑再行选取。

群落个数 F : 群落个数代表建立多少个狮群,各个群落会通过驱逐行为将非狮王的雄狮驱逐出狮群,这些雄狮会加入

同一个流浪狮群,在触发领地争夺行为时,流浪狮群中的雄狮可能会替代群落中的雄狮,从而实现原本应该独立进化的狮群,会通过这些智能行为建立联系,由别的狮群中的优质雄狮来帮助自己的群落进化,从而更快地跳出局部最优,朝最优解的方向进化。由图 10(d)可知,在保持原参数设置的基础上,种群个数 F 设为 4 较优,种群个数过小或过大都会影响算法的性能,所以在选取时需要合理考虑群落中的雌雄狮比例,选取群落个数时要结合种群的规模、初始雄狮比例等来考虑。

综上所述,改进狮群进化算法针对不同的问题场景可以选取不同的参数组合,从时效均衡的角度考虑,参数组合: $U=40, V=200, Pr=0.8, vp=0.2, F=4$ 能使 MMCP-SCP-LEA 取得较好的效果。

结束语 首先,本文针对 MMPP-SCP,结合现实生活空间众包问题场景下工作者和任务的匹配标准,根据工作者与任务的时空信息,构建 MMPP-SCP 模型;其次,在此基础上提出针对该问题的改进狮群进化算法,并提出与 MMPP-SCP 相匹配的染色体编码方式,设计了对应的保序双点交叉操作和保序双点变异操作,借鉴狮群进化算法的算法思想,设计了 MMPP-SCP-LEA;最后,根据真实数据集制作问题算例,并通过不同算法在不同问题场景下的对比分析,表明了 MMPP-SCP-LEA 的有效性与可用性,通过 3 种算法在 3 种不同问题场景下的收敛情况分析,表明了 MMPP-SCP-LEA 具有良好的收敛性。

考虑到问题的复杂性,本文只研究了在离线场景下的小数据量情形下的面向空间众包平台的多工作者多任务路径规划问题。空间众包平台在一定的时间阈值下便会获取任务集合和工作者集合进行一次路径规划,在平台一次路径规划结束后,在一定的时间阈值下会再次获取任务集合和工作者集合进行下一次路径规划。动态场景下的面向空间众包平台的多工作者多任务路径规划问题可划分为若干个静态路径规划问题,在即将到来平台的任务和工作者的时空信息未知的前提下,动态场景下应用本文提出的 MMPP-SCP-LEA 可获得良好的路径规划方案。由于 MMPP-SCP-LEA 在大数据量情形下进行路径规划预期运行时间较长,为进一步提高模型和算法的有效性与可用性,接下来将尝试通过网格划分的方式,合理划分任务分布区域,引入时间阈值,建立动态优化框架,在动态场景下的数据量较大且分布稀疏度不均匀的情形下进行路径规划。

参 考 文 献

- [1] GAO D, TONG Y, SHE J, et al. Top-k Team Recommendation and Its Variants in Spatial Crowdsourcing[J]. Data Science and Engineering, 2017, 2(2): 136-150.
- [2] ZHANG C, GUO Y C, LIN P G, et al. Location prediction-based task assignment in spatial crowdsourcing[J]. Journal of Nanjing University(Natural Science), 2018, 54(2): 471-480.
- [3] TONG Y X, YUAN Y, CHENG Y R, et al. Survey on Spatio-temporal Crowdsourced Data Management Techniques[J]. Journal of Software, 2017, 28(1): 35-58.
- [4] TO H, SHAHABI C, KAZEMI L. A Server-Assigned Spatial Crowdsourcing Framework[J]. ACM Transactions on Spatial Algorithms & Systems, 2015, 1(1): 1-28.

[5] DENG D, SHAHABI C, DEMIRYUREK U. Maximizing the Number of Worker's Self-Selected Tasks in Spatial Crowdsourcing[C]// ACM Sigspatial International Conference on Advances in Geographic Information Systems. New York: Association for Computing Machinery, 2013: 324-333.

[6] ZHAO T H, TU W, LUAN Z L, et al. Dynamic Task Schedule Algorithm of Spatial Crowdsourcing for the Worker[J]. 2019, 44(2): 41-44.

[7] NIE X C, ZHANG Y, YU D H, et al. Spatial Crowdsourcing task allocation algorithm for global optimization[J]. Journal of Computer Applications, 2020, 40(7): 1950-1958.

[8] TONG Y, ZHOU Z, ZENG Y, et al. Spatial crowdsourcing: a survey[J]. The VLDB Journal, 2020, 29(1): 217-250.

[9] COSTA C F, NASCIMENTO M A. In-Route Task Selection in Crowdsourcing[C]// Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York, NY, USA: Association for Computing Machinery, 2018: 524-527.

[10] CUI J Y, CHEN D, YUAN Y, et al. Online route planning algorithm in spatial crowdsourcing[J]. Journal of Tsinghua University (Science and Technology), 2020, 60(8): 672-682.

[11] SHE J, TONG Y, CHEN L. Utility-aware social event-participant planning[C]// Proceedings of the 2015 ACM International Conference on Management of Data. 2015: 1629-1643.

[12] MA S, ZHENG Y, WOLFSON O. T-share: A large-scale dynamic taxi ridesharing service[C]// 2013 IEEE 29th International Conference on Data Engineering (ICDE). IEEE, 2013: 410-421.

[13] MENG F, ZHANG S, ZHENG X, et al. Task Assignment Method in Spatial Crowdsourcing Based on Graph Search[C]// 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2019: 2623-2629.

[14] TAO Q, ZENG Y, ZHOU Z, et al. Multi-Worker-Aware Task Planning in Real-Time Spatial Crowdsourcing[M]// Database Systems for Advanced Applications. Cham: Springer, 2018: 301-317.

[15] GAO D, TONG Y, JI Y, et al. Team-Oriented Task Planning in Spatial Crowdsourcing[C]// Asiapacific Web. Cham: Springer, 2017: 41-56.

[16] AN Y, QIN K, LUO G C. Survey on location privacy preservation technology in spatial crowdsourcing[J]. Application Research of Computers, 2018, 35(8): 2241-2244, 2264.

[17] LI Y, JIA M D, YANG W Y, et al. Optimal Task Assignment Algorithm Based on Tree-Decouple in Spatial Crowdsourcing[J]. Journal of Software, 2018, 29(3): 824-838.

[18] SONG T S, TONG Y X, WANG L B, et al. Online Task Assignment for Three Types of Objects under Spatial Crowdsourcing Environment[J]. Journal of Software, 2017, 28(3): 611-630.

[19] ZHANG Y Z, XU T Z, ZHENG J S, et al. Solving Emergency Logistics Problem with Uncertain Urgency Based on a Hybrid Genetic Algorithm[J]. Journal of Systems Science and Mathematical Sciences, 2020, 40(4): 714-728.

[20] RAJAKUMAR B R. The Lion's Algorithm: A New Nature-Inspired Search Algorithm[J]. Procedia Technology, 2012, 6: 126-135.

[21] TONG R. Study on takeaway distribution routing optimization based on crowdsourcing mode[D]. Xi'an: Xi'an University of Technology, 2019.

[22] HARVEY I. The Microbial Genetic Algorithm [C]// European Conference on Artificial Life. Berlin: Springer, 2009: 126-133.



ZHAO Yang, born in 1996, postgraduate. His main research interests include spatial crowdsourcing and intelligent computing.



NI Zhi-wei, born in 1963, Ph.D, professor, Ph.D supervisor. His main research interests include artificial intelligence, machine learning and cloud computing.