

基于改进蚁群算法的共享单车配送调度研究

吴会丛 王 敬

(河北科技大学信息科学与工程学院 河北 石家庄 050018)

摘要 针对使用蚁群算法对共享单车进行配送得到最优解的问题,通过设定蚂蚁觅食路径上初始信息素浓度和信息素更新方法获得更优的配送距离,对蚁群算法进行改进。将对共享单车使用量有影响的特征因素输入到 XGBoost 模型中进行预测,根据转移概率通过轮盘赌的方式选择接下来要访问的租赁点,利用提出的基于初始信息素衰减的方法进行更新,以快速得到更优的解。通过对某学校内的 14 个租赁点进行实验,在初始信息素衰减的基础上,改变初始信息素浓度,可以在较短时间内获得更短的配送距离,比基本蚁群算法的距离缩短了约 1%,实验结果验证了该算法的有效性。

关键词 共享单车 蚁群算法 信息素浓度 更新信息素方法 最短距离

中图分类号 TP312 **文献标志码** A **DOI**:10.3969/j.issn.1000-386x.2020.09.006

SHARED BICYCLE DISTRIBUTION SCHEDULING BASED ON IMPROVED ANT COLONY ALGORITHM

Wu Huicong Wang Jing

(College of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, Hebei, China)

Abstract To solve the problem of using ant colony algorithm to get the optimal solution for the distribution of shared bicycles, this paper improves the ant colony algorithm by setting the initial pheromone concentration and pheromone update method on the ant feeding path to get a better distribution distance. The characteristic factors that affect the usage of shared bicycle were input into the XGBoost model for prediction. According to the transfer probability, the next rental point to be visited was selected through roulette, and the method based on the initial pheromone attenuation was used to update, so as to get a better solution quickly. Through the verification of 14 rental points in a school, based on the attenuation of the initial pheromone, changing the concentration of the initial pheromone can get a shorter distribution distance in a short time, which is about 1% shorter than the distance of the basic ant colony algorithm. The experimental results show the effectiveness of the algorithm.

Keywords Shared bicycle Ant colony algorithm Pheromone concentration Update pheromone method Shortest path

0 引言

当今城市发展迅速,为了解决城市交通问题,共享单车应运而生。随着共享单车的使用率逐步增加,其在各个租赁点之间的配送调度问题也随之产生^[2]。各个租赁点的共享单车不能完全满足用户的需求,或者

很容易造成极大的资源浪费,因此需要管理者对共享单车进行合理地调度并及时进行配送,最大限度满足用户的使用需求。如何均衡共享单车在不同租赁点的分布,满足用户使用共享单车的需求量,寻找调度配送路径的最短距离,对完善公共交通体系,提升交通服务水平,保护城市环境具有非常重要的意义。

目前,对于调度问题已经有了许多研究。文献[3]

提出了一种以基于斥候蚁实现的动态局部搜索为核心策略,城市分类、加大信息素浓度差和继承式的信息素清零等三种策略进行辅助改进的蚁群算法。文献[4]在蚁群算法中引入遗传算法中一些遗传算子的内容,经过对循环次数分组后,对组内、组外进行操作,以加快搜索出最优解的速度。文献[5]通过 BP 神经网络对单车的初始量进行预测,完成之后以期望调度次数最少为目标,建立整数规划模型,利用 lingo 进行求解,并得出期望调度次数。文献[6]使用两个约束函数来评估并提供有关性能和预算成本的反馈,这两个约束函数使改进的蚁群算法根据反馈情况及时调整解决方案的质量,以实现最优解。文献[7]对区域化管理给出了一种基于 P-中值模型的联营区划分方法,实验证明该方法灵活方便。对于高峰时的调度需求,建立以加权移动平均法为基础的模型,最后采用遗传算法进行模型求解。

上述算法存在一些问题,如:只适合空间复杂度较小的情况;只考虑调度次数而没有考虑配送距离;没有采用实际运营数据做支撑而难以判断优化性。因此,本文提出一种既适合较大空间范围又考虑到配送距离,具有运营数据支撑的租赁点之间蚂蚁残留信息素浓度的计算方法。通过该方法能够在一定的时间内得到一段较合理的配送路线,实验结果表明,该方法能够得到行驶距离较短的配送调度路线。

1 需求分析及预测

1.1 调度需求分析

共享单车的使用情况会受到季节、星期、时间等因素的影响^[8]。在天气较恶劣的情况下,用户对共享单车的使用意愿会下降,共享单车的利用率也会大大降低,此时,管理者就可以从拥有共享单车数量充足或者富余的租赁点中调一部分单车到有需要的租赁点中,以免造成资源的浪费。在春季或者温度适中的情况下,共享单车的使用率会大大增加,此时,应该适当地增加对共享单车的派送量,以满足用户的使用需求。

本文实验所需的数据集包含日期、季节、是否节假日、是否工作日、天气、湿度、温度和风速等相关因素。值得注意的是,日期时间特征由年、月、日和具体小时组成,还可以根据日期计算其星期,因此可以将日期时间拆成年、月、日、时和星期 5 个特征。计算得出预测车辆数与各相关因素之间的相关系数,如表 1 所示。

表 1 预测车辆数与各相关因素之间的相关系数

特征因素	相关系数	特征因素	相关系数
小时	0.401	天气	0.129
温度	0.394	风速	0.100
湿度	0.318	是否节假日	0.019
季节	0.163	是否工作日	0.011

1.1.1 时间对共享单车数量的影响

在工作日或者周末时,某租赁点共享单车的使用率有所不同,而且在不同的时间段内也不尽相同。工作日每天都有两个高峰期,分别是临近八点和十七点三十分左右,这个时间段正好是工作日的上、下班的时间点,因此,共享单车的需求量会大大增加,而介于这两个时间段之间的单车使用量则比较少。在周末时,用户的普遍用车量主要集中在上午十点至晚上七点的时间段里,但最高的使用量仍旧达不到上、下班高峰时的使用量。时间对共享单车数量的影响如图 1 所示。

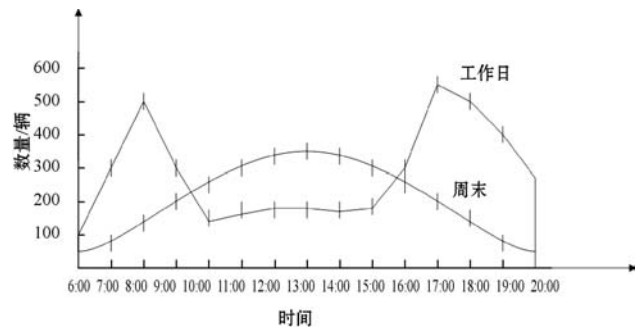


图 1 时间点对共享单车数量的影响

1.1.2 季节对共享单车数量的影响

通过对某地共享单车使用情况的分析,可以知道季节对某一租赁点共享单车的使用量存在一定的影响。在天气较寒冷的冬季,共享单车的使用量相对较少,在天气较好的夏季和秋季,其使用量明显增加。季节对共享单车数量的影响如图 2 所示。

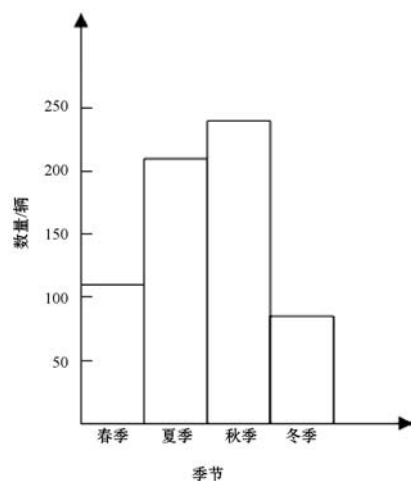


图 2 季节对共享单车数量的影响

此外,温度、湿度、风速等原因也会对共享单车的租用情况有一定的影响。

1.2 基于 XGBoost 的需求预测

通过上述分析,可以知道共享单车的使用需求会随着时间、季节、星期、温度、风速等因素的改变而改变。本文通过将些影响因素作为自变量输入到 XGBoost 预测模型^[9]中,得到各个租赁点的实时需求,然后根据实时需求量与当前租赁点的实际数量的差值,为各个租赁点配送对应数量的共享单车,以满足用户的需求。

本文采用的 XGBoost 模型是基于 Scikit-learn 接口的分类,它采用线性回归模型,基于树的模型进行提升计算。实验所用到的每一组数据样本都包含日期、季节、是否节假日、是否工作日、天气、湿度、温度和风速等相关因素。根据特征因素的特性,将这些因素与已有的标签因素匹配对应,从而得到比较准确的预测数据,即根据以往同样或相似的前提条件,预测此时可能需要的共享单车的数量,以此达到及时配送的可能。

2 共享单车任务调度

共享单车的调度问题,主要发生在拥有自行车数量过多或者过少的租赁点中,此时需要对租赁点进行合理的调度配送,以解决租赁点“无车可借,无处可还”的欠佳状态。

目前,城市公共自行车系统一般都具有成百上千个租赁点,这是非常庞大的。本文采用小范围内的地区进行实例验证,同时采用单一车辆进行调度配送。根据各个租赁点的配送量和各个租赁点之间的距离,通过改进蚁群算法中初始信息素浓度和路径中信息素的更新方法在较短时间内求得更优配送路线^[10]。

2.1 问题描述

在某学校内选择具有代表性的 15 个租赁点(停靠点),规定其中一个租赁点为配送中心,将其设置为配送路线的出发点,亦是终点。根据各个租赁点的配送量和距离信息,对其余 14 个租赁点进行合理的配送。首先从配送中心出发,用一辆负载量足够大的配送车辆为该区域内的 14 个租赁点配送共享单车,各个租赁点的地理位置是固定不变的,但是各个租赁点的需求量是随着时间、天气等外界因素的变化而实时变化的,由此可知时刻调度是非常有必要的^[11]。

配送车辆的负载量是已知且固定的,将所有租赁点的相对位置以坐标的形式显示在画布上,其数据是从百度地图中的坐标拾取系统中得到的。要求在不重

复访问各个租赁点,且满足所有租赁点的需求并符合各个约束条件的情况下,制定出合理的车辆配送路线,以实现配送距离最短的目标^[12]。要求满足以下条件:

- 1) 配送路线上各个租赁点的配送量之和不能超过配送车辆的最大负载量。
- 2) 配送路线上各个租赁点的需求必须都满足。
- 3) 配送路线中,每个租赁点有且只能访问一次。
- 4) 配送车辆都是由调度中心出发,最后返回调度中心。

2.2 参数设置

租赁点集合 $S = \{1, 2, \dots, m\}$, 各租赁点的需求量 $C = [c_0, c_1, \dots, c_m]$, 租赁点的横、纵坐标分别为 $distance_x = [x_0, x_1, \dots, x_m]$, $distance_y = [y_0, y_1, \dots, y_m]$, 配送车最大负载量 $Maxbike$ 和决策变量 x_{ij} 。

$$x_{ij} = \begin{cases} 1 & \text{配送车访问了从租赁点 } i \text{ 到租赁点 } j \text{ 之间的路线} \\ 0 & \text{配送车没有访问租赁点 } i \text{ 到租赁点 } j \text{ 之间的路线} \end{cases}$$

2.3 数学描述

共享单车配送调度的目标函数为配送路线的最短距离:

$$\min f = total_distance \quad (1)$$

式中: f 为目标函数; $total_distance$ 为配送距离总长度。

$total_distance$ 由三大部分组成,分别是配送中心到租赁点 i 的距离 d_{0i} 、各个租赁点之间的距离 d_{ij} 和租赁点 j 返回到配送中心的距离 d_{j0} , 其计算公式如下:

$$total_distance = d_{0i} + \sum_{j=1}^m \sum_{i=1}^m d_{ij} x_{ij} + d_{j0} \quad i, j \in S \quad (2)$$

该实验还有很多约束条件:在同一条配送路线上,各个租赁点需要配送的车辆总和不能超过配送车辆的最大负载重量,即:

$$\sum_{i=1}^m b_i \leq T \quad (3)$$

式中: b_i 为各个租赁点实际需要配送的共享单车数量, T 为配送车辆的最大负载量,本文中 $T = 200$ 。

在配送共享单车的路线中,各个租赁点能且只能访问一次,公式如下:

$$\sum_{i=1}^m x_{ij} = 1 \quad i \neq j, i = 1, 2, \dots, m \quad (4)$$

$$\sum_{j=1}^m x_{ij} = 1 \quad i \neq j, j = 1, 2, \dots, m \quad (5)$$

根据 XGBoost 模型预测出每个租赁点的需求量,即 c_1, c_2, \dots, c_m 。预测模型中的输入为影响租赁数量的各个特征因素,输出为租赁点的实时需求量。将必要的特征因素输入到 XGBoost 模型中,采用上述参数进行预测,得到比较符合实际的需求量 c_i 。

根据当前各个租赁点已有共享单车数量与预测出

的需求量,得到实际需要的配送量 b_i 。 b_i 为已有单车数与单车需求量的偏差值:当 $b_i > 0$ 时,供大于求,此时需要将 b_i 辆共享单车装上货车;当 $b_i < 0$ 时,供不应求,此时需要卸下配送车中 $|b_i|$ 辆共享单车,以供用户使用。

共享单车的调度问题,主要是为了满足人们平时的需求。如果想对现在或者未来某时间段内各个租赁点所需的共享单车进行配送,可以与之前已知数据中同样的时间段、天气、月份、温度、湿度和风速等特征因素进行匹配,由此预测出当前或者未来时段内租赁点可能需要的单车数量。然后根据预测出来的数量以及距离等相关因素,对配送的调度路线进行合理有效的划分,并派出调度员进行调度配送,及时满足人们的出行需求,从而达到对单车的合理利用并促进社会的和谐发展。

3 基于信息素衰减的蚁群算法

目前已经有许多启发式算法应用于路径调度问题中,其中蚁群算法尤为显著。本文采用基于蚁群的启发式算法对共享单车的配送调度进行调度。

3.1 规定参数

基本蚁群算法中涉及许多的参数,主要有信息启发式因子、期望启发式因子、蚁群数量、信息挥发因子等重要参数,同时还包括配送车量的最大负载量、信息素的初始浓度等。

3.2 获取禁忌列表

对于一条需要配送的路径来说,所有的租赁点能且只能访问一次,以免获得多余的行程,因此需要将访问过的租赁点存储到禁忌表中^[13],避免重复访问。在所有可以访问的租赁点中,还要考虑当前配送数量的正负情况:配送数量为正,且当前配送车上已拥有的数量与配送数量的和仍不超过最大负载量 $Maxbike$,或者配送数量为负,且当前配送车上已拥有的数量与该配送数量的和大于等于零时,该租赁点才可以加入配送调度路线中。

3.3 轮盘赌选择租赁点

在蚂蚁的觅食过程中,蚂蚁选择下一个租赁点的方式多种多样,本文实验采用典型的轮盘赌方法^[14]来进行选择。

蚂蚁选择每个租赁点的概率采用两个租赁点之间的距离与当前路径上信息素浓度的比值来计算:

$$P_i = \frac{\tau_{ij}^\alpha}{d_{ij}^\beta} \quad (6)$$

式中: P_i 为选择概率; α 为信息启发式因子; β 为期望启发式因子; τ_{ij} 为信息素浓度; d_{ij} 为租赁点 i 到租赁点 j 之间的距离。

通过式(6)得到各个租赁点被选中的概率,将这些概率求和得到总的概率值,然后得到到达各个租赁点真正的概率。虽然得到了到达每一个租赁点的概率,但由于蚁群算法的多样性,因此也要保证其他租赁点有被选中的机会。否则,如果只选择概率最大的租赁点,就会变成贪心算法。因此用轮盘选择的方法,获得每个租赁点的概率值,即随机产生一个 0 到总概率范围之间的随机浮点数,然后轮次相减到各个租赁点的概率值,直至为负数,此时得到下一个租赁点的序号。重复同样的操作,直到所有的租赁点全部遍历结束。

3.4 信息素衰减

信息素的更新^[15]在蚁群算法中是一个需要解决的难点。蚁群算法将初始信息素浓度设置为一个比较小的值,在所有蚂蚁进行一次完整的行走后,环境中的信息素就需要进行更新。信息素的变化主要受到两个因素的影响:每只蚂蚁在走过的路径中留下的信息素;环境信息素的自然减少,即原有的路径上信息素浓度会随着时间的增加而有适当的挥发。因此,信息素浓度包括整条路径上残留的初始信息素浓度以及在整条路径上新迭代产生的信息素浓度如下:

$$pheromone_graph[i][j] = pheromone_graph[i][j] \times (1 - \rho) + temp_pheromone[i][j] \quad (7)$$

式中: $temp_pheromone[i][j]$ 为新迭代的信息素浓度; $pheromone_graph[i][j]$ 为总的信息素浓度。

而新迭代产生的信息素浓度又包括蚂蚁在其路径上留下的信息素以及当前两个租赁点之间残留的信息素,如式(8)所示。在蚂蚁搜索的过程中,租赁点之间的信息素不断减少,会在一定程度上影响到蚂蚁的选择。

$$temp_pheromone[start][end] = (1 - \rho) \times temp_pheromone[start][end] + Q/ant.total_distance \quad (8)$$

式中: ρ 为信息挥发因子; Q 为信息素的初始浓度; $ant.total_distance$ 为配送的总路径。

通过式(7)和式(8)对路径上的信息素浓度进行更新,达到一定的迭代次数时可以得到较优的路径解。本文主要从路径上信息素的浓度进行改进。首先

设置信息素的初始浓度为一个较大值,然后对新迭代的信息素浓度进行修改,实验发现每只蚂蚁在其路径上留下的信息素与蚂蚁走过此次路径所携带的信息素浓度的相对残留度和信息素的增量有关。通过将新迭代的信息素以及初始信息素浓度都进行适当衰减后,能够在更加少的迭代次数时得到最优的配送路线,如算法1所示。

算法1 初始信息素和路径信息素的更新

输入:路径和初始信息素信息。

输出:更新后的信息素。

```

1. for ant in self. ants:
2.     ant. search_path()
3.     if ant. total_distance < self. best_ant. total_distance:
4.         self. best_ant ← copy. deepcopy( ant)
5. //信息素更新:
6. for ant in self. ants:
7.     for l to m:
8.         start, end ← ant. path[ i - 1 ], ant. path[ i]
9.         temp_pheromone[ start ][ end]
10.        ←( 1 - ρ) × temp_pheromone[ start ][ end] +
11.        Q/ant. total_distance
12.        temp_pheromone[ end ][ start]
13.        ←temp_pheromone[ start ][ end]
14. for( 租赁点):
15.     for( 租赁点):
16.         pheromone_graph[ i ][ j]←pheromone_graph
17.         [ i ][ j] × ( 1 - ρ) + temp_pheromone[ i ][ j]

```

3.5 计算距离

本实验中蚂蚁行驶的路径长度,不考虑道路的实际情况,仅考虑两租赁点之间的最短距离。首先需要将这15个租赁点根据其经纬度在画布中画出相对应的位置,然后采用欧氏距离求解任意两个租赁点之间的距离,并求和得到总的配送距离,如算法2所示。

算法2 计算总行驶距离

输入:各个租赁点之间的距离。

输出:配送路线的总距离。

```

1. for i in range( city_num ):
2.     for j in range( city_num ):
3.         temp_distance←pow( ( distance_x[ i ] -
4.         distance_x[ j] ), 2) + pow( ( distance_y[ i ] -
5.         distance_y[ j] ), 2)
6.         temp_distance←pow( temp_distance, 0. 5)
7.         distance_graph[ i ][ j]←float( int( temp_
8.         distance + 0. 5) )

```

与现有解决路径调度问题的蚁群算法和遗传算法

相比,本文提出的租赁点之间蚂蚁残留信息素浓度的计算方法可以更有效地调度共享单车的配送,同时得到更短的调度路线。

4 实验

本文通过设定初始信息素浓度以及信息素的更新方法对基本的蚁群算法进行改进,使调度路线能够在较短的时间内得到。为了减少得到最短配送路线的迭代次数,本文对信息素浓度进行合理的衰减,以在短时间内得到配送路线与最短距离。

4.1 实验设计

从百度地图中的拾取坐标系统中获得某学校内选定的15个租赁点(包括配送中心)所对应的经纬度,然后再根据经度和纬度计算得到在画布中相对应的横、纵坐标,如表2所示。

表2 租赁点的横、纵坐标

租赁点	(x,y)	租赁点	(x,y)
0	(890,383)	8	(100,280)
1	(32,386)	9	(176,426)
2	(523,25)	10	(163,90)
3	(292,618)	11	(67,303)
4	(540,270)	12	(87,556)
5	(416,533)	13	(63,107)
6	(666,406)	14	(580,60)
7	(171,275)		

本文采用的调度算法是对蚁群算法的改进,因此基本蚁群算法中的各个实验参数在该实验中同样适用。参数设置如表3所示。

表3 实验参数表

参数名称	数值	参数名称	数值
α	2.0	Q	100.0
β	1.0	zu_num	15.0
ρ	0.4	ant_num	22.0
T	200.0		

为了保证后期能够准确地对15个租赁点的需求进行配送调度,本文采用基于XGBoost的预测模型对各个租赁点进行需求预测,其中最大深度为3,学习步长为0.1,迭代次数为100次。然后与当前各个租赁点已经拥有的共享单车数量作比较,得到需要配送的车辆数 b_i 如表4所示。

表 4 各租赁点需要配送共享单车的数量

租赁点	配送量	租赁点	配送量
0	30	8	-2
1	-20	9	-5
2	20	10	-10
3	-30	11	-20
4	30	12	-5
5	25	13	8
6	10	14	20
7	-8		

4.2 实验结果及分析

由于人们对共享单车的需求是随着一些外界因素而改变的,因此需要时时刻刻了解共享单车的使用情况。首先根据以前时间段得到的数据对其影响因素进行大致分析,然后通过 XGBoost 模型对未来时刻的共享单车的需求量进行分析,并与当前已有数量进行对比,得到真正需要配送的单车数量。最后通过改进的蚁群算法对各个租赁点进行车辆配送,以及时满足人们的满意度。利用本文获取到的某学校内的 15 个租赁点进行实验,它们所需共享单车的配送路线如图 3 所示,其配送路线为:0-4-14-2-10-13-7-8-11-1-12-9-5-3-6-0。

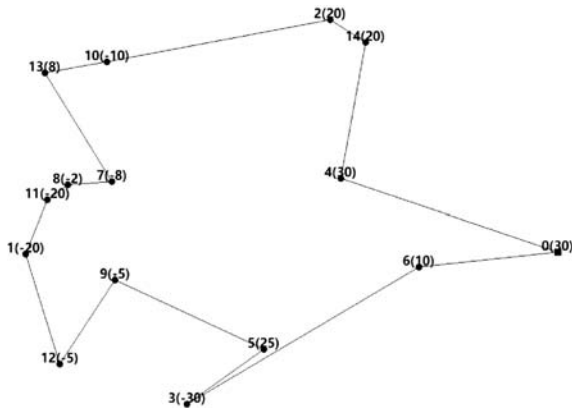


图 3 改进蚁群算法的调度路线

基本蚁群算法参数设置见表 3,初始信息素的浓度为 1。在没有设置较高的初始信息素浓度和改进信息素的更新方法前,在迭代 20 000 次以前,始终无法达到图 3 所示的最优路径,但是在迭代 10 次左右时可以得到次优的配送路线:0-6-4-14-2-10-13-7-8-11-1-12-9-5-3-0,此路线的配送距离为 2 952 m。

考虑到最大最小蚂蚁系统^[16]中,对初始信息素的浓度有设定,因此,将初始信息素浓度设置为一个较大的数值,防止在搜索路途中,初始信息素浓度淡化以影

响蚂蚁的判断。

与基本蚁群算法相比,当改进信息素的更新方法和初始信息素浓度之后,能够得到更短距离的配送路线,其迭代次数也比较小。在改进的算法中,初始信息素浓度为 100 时,仅需要迭代 26 次就可以获得配送距离为 2 922 m 的行驶路线。而采用遗传算法求解该问题时,发现很难达到以上这两种短距离的配送路线,其达到的最短距离为 3 774 m。虽然得到该结果的迭代次数并不是很高,但是经过实验对比发现该结果的可行性比较低,因此舍弃该算法。为了更加直观地显示该内容,用图表示它们之间的关系。对于不同的初始信息素浓度,本文提出的基于信息素衰减的更新方法也有一定的作用。

4.2.1 初始信息素浓度为 1 时的情况

当初始信息素浓度为 1,基本蚁群算法和改进的蚁群算法分别得到不同的最短配送距离,如图 4 所示。

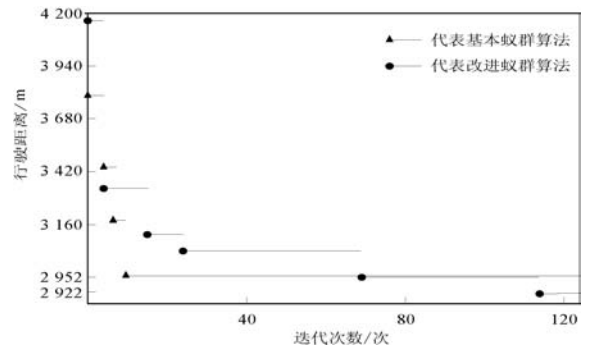


图 4 初始信息素浓度为 1 时的情况

可以看出,当初信息素浓度为 1 时,基本的蚁群算法得到的最短距离为 2 952 m,其迭代次数为 10 次左右;改进蚁群算法迭代 114 次,得到的最短距离为 2 922 m,这个距离在基本蚁群算法中是短时间内无法得到的。

4.2.2 初始信息素浓度为 100 时的情况

当初始信息素浓度为 100 时,改进前和改进后的蚁群算法得到最优配送距离如图 5 所示。

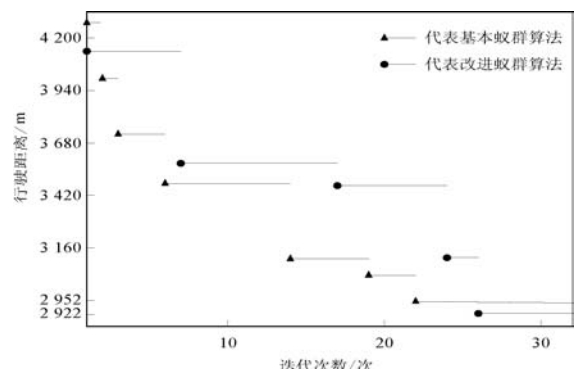


图 5 初始信息素浓度为 100 时的情况

可以看出,初始信息素浓度为 100 时,基本蚁群算法仍旧无法在短时间内获得 2 922 m 这个较短的配送

距离,而改进后的算法仅在迭代 26 次时便可以获得 2 922 m 的配送路线。

4.2.3 算法对比

基本蚁群算法、遗传算法和改进的蚁群算法得到的最短配送距离的关系如图 6 所示。

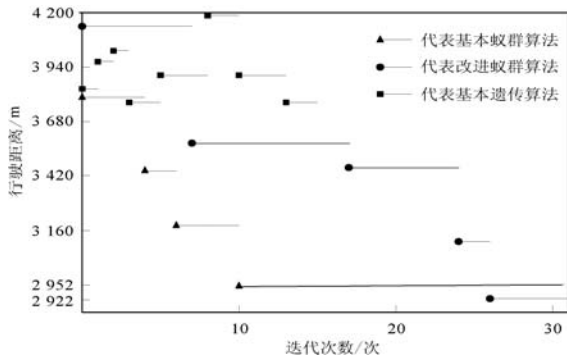


图 6 三种算法行驶距离的比较

可以看出,在本实验中,蚁群算法比遗传算法得到的结果更好,蚁群算法的结果依次递减,而遗传算法并没有得出一个比较好的配送结果,并且结果具有波动性。改进后的蚁群算法又比基本的蚁群算法有所改善,当信息素的浓度有一个较大的数值时,考虑到蚂蚁在觅食过程中信息素浓度的挥发,由此得到最优路径的迭代次数要小很多。这是因为路径上信息素浓度足够大时,随着时间的增加,信息素的浓度还会存在一定的量,不会在多次访问后,完全淡化。此外,改进后的蚁群算法也能得到更短的配送距离。

通过以上实验得到以下几点结论:

- 1) 各个租赁点的实时需求对于动态调度是非常有必要的,具有及时性。
- 2) 影响共享单车需求的因素主要是季节、时间等因素。
- 3) 不改变初始信息素浓度时,仅改进信息素更新方法得到的配送距离比基本蚁群算法和遗传算法要小。
- 4) 本文提出的改进蚁群算法可以获得更短的配送距离,并且迭代次数也比较小。

本文对初始信息素浓度以及信息素的更新方法进行了设计。由此得到的最优配送共享单车的调度路线为:0-4-14-2-10-13-7-8-11-1-12-9-5-3-6-0,最短距离为 2 922 m,迭代次数为 26 次,此最短距离比基本蚁群算法的最短距离缩短了约 1%,比遗传算法缩短了约 22%,而且迭代次数也相对较少,如表 5 所示。改进的蚁群算法可以获得更优的配送距离,同时大大减少了程序运行时间和计算时间,使管理员能够在短时间内得到最优的配送路线,及时为各个租赁点配送适量的共享单车,以满足人们的需

求,因此该算法具有一定的实用性与可行性。

表 5 不同算法的配送距离与迭代次数对比结果

调度算法	信息素浓度	配送距离/m	迭代次数
遗传算法	-	3 774	4
基本蚁群算法	1	2 952	10
	100	2 952	22
改进蚁群算法	1	2 922	114
	100	2 922	26

5 结 语

本文针对共享单车调度路线问题,首先确定需要进行配送的租赁点,然后根据 XGBoost 模型求出各个租赁点的需求量,通过需求量来判断需要如何安排调度路线,并根据需求约束和距离约束等因素,采用改进后的蚁群算法设计调度方案,得到比基本蚁群算法更短的配送距离,迭代次数也比较小,同时也减少了计算时间,具有一定的现实意义。

对于路径调度问题,蚁群算法是非常有效且实用的算法之一。目前已经有很多人对其进行了改进,并得到了相当可观的结果。本文改进方法只是其中的一方面,可能考虑得还不是很全面,使用的数据也是自己收集到的数据,后续还需要继续阅读大量的文献,以对该算法有更加全面、新颖的了解与认识。缩短得出最优路线的时间,对于实时配送共享单车是非常有必要的,这样能够给配送共享单车留出更加充足的时间,防止配送过程中出现意外。

参 考 文 献

- [1] 徐冠宇. 城市公共自行车调运优化问题研究[D]. 成都:西南交通大学,2017.
- [2] 周炳海,谭芬. 基于循环配送策略的汽车装配线物料配送调度方法[J]. 东北大学学报(自然科学版),2018,39(3):389-393.
- [3] 周树伦. 一种基于动态局部搜索的蚁群算法及其对 TSP 问题的求解[D]. 南宁:广西大学,2016.
- [4] 李茹. 基于改进混合蚁群算法的物流配送路径优化研究[D]. 太原:山西大学,2017.
- [5] 周龙雨,常兴甲,田博. 基于 BP 神经网络的共享单车调度优化[J]. 中国战略新兴产业,2017,3(24):60.
- [6] Zuo L, Lei S, Dong S, et al. A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing[J]. IEEE Access,2017,3:2687-2699.

- [4] Yang S S, Hanzo L. Fifty years of MIMO detection: The road to large-scale MIMOs [J]. *IEEE Communications Surveys and Tutorials*, 2015, 17(4): 1941 – 1988.
- [5] Bogale T, Le L. Massive MIMO and mmWave for 5G wireless HetNet: Potential benefits and challenges [J]. *IEEE Vehicular Technology Magazine*, 2016, 11(1): 64 – 75.
- [6] Jose J, Ashikhmin A, Marzetta T L, et al. Pilot contamination and precoding in multi-cell TDD systems [J]. *IEEE Transactions on Wireless Communications*, 2011, 10(8): 2640 – 2651.
- [7] Wang Q, Jing Y. New rank detection methods for reduced-rank MIMO systems [J]. *Eurasip Journal on Wireless Communications and Networking*, 2015(1): 230.
- [8] Appaiah K, Ashikhmin A, Marzetta T L. Pilot contamination reduction in multi-user TDD systems [C]//2010 IEEE International Conference on Communications. IEEE, 2010: 1 – 5.
- [9] Jin S, Wang X Y, Li Z, et al. On massive MIMO zero-forcing transceiver using time-shifted pilots [J]. *IEEE Transactions on Vehicular Technology*, 2016, 65(1): 59 – 74.
- [10] Xu C, Zhang J H, Liu M M, et al. Pilot design for sparse channel estimation in large-scale MIMO-OFDM system [J]. *International Journal of Antennas and Propagation*, 2016, 2016(5): 6142574.
- [11] Zhang C C, Qiu R C. Massive MIMO as a big data system: random matrix models and testbed [J]. *IEEE Access*, 2017, 3: 837 – 851.
- [12] Rusek F, Persson D, Lau B K, et al. Scaling up MIMO: Opportunities and challenges with very large arrays [J]. *IEEE Signal Processing Magazine*, 2013, 30(1): 40 – 60.
- [13] Li M, Jin S, Gao X. Spatial orthogonality-based pilot reuse for multi-cell massive MIMO transmission [C]//2013 International Conference on Wireless Communications and Signal Processing. IEEE, 2013: 1 – 6.
- [14] Vinogradova J, Bjornson E, Larsson E G. On the separability of signal and interference-plus-noise subspaces in blind pilot decontamination [C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016: 3421 – 3425.
- [15] Fan D, Gao F F, Wang G P, et al. A 2D-DFT based channel estimation scheme in indoor 60GHz communication systems with large-scale multiple-antenna [C]//2016 83rd IEEE Vehicular Technology Conference. IEEE, 2016: 1 – 5.
- [16] Muppirisetty L S, Wymeersch H, Karout J, et al. Location-aided pilot contamination elimination for massive MIMO systems [C]//2015 IEEE Global Communications Conference. IEEE, 2015: 1 – 5.
- [17] Wang P B, Li R, Zhao C L, et al. Joint interference mitigation approach using space-time pilot design in large-scale antenna systems [J]. *IET Communications*, 2016, 10(3): 253 – 262.
- [18] Zhu X D, Wang Z C, Dai L L, et al. Smart pilot assignment for massive MIMO [J]. *IEEE Communications Letters*, 2015, 19(9): 1644 – 1647.
- [19] Zhu X D, Wang Z C, Qian C, et al. Soft pilot reuse and multi-cell block diagonalization precoding for massive MIMO systems [J]. *IEEE Transactions on Vehicular Technology*, 2015, 65(5): 3285 – 3298.
- [20] 樊自甫, 胡敏, 李悦宁. 基于图着色的大规模 MIMO 系统中动态导频分配 [J]. *计算机应用*, 2017, 37(12): 3356 – 3360.
-
- (上接第 41 页)
- [7] 叶钦海. 城市公共自行车系统运营调度研究 [D]. 广州: 华南理工大学, 2017.
- [8] 刘路美. 城市公共自行车站点需求预测及调度优化方法研究 [D]. 北京: 北京交通大学, 2017.
- [9] 贾文慧, 孙林子, 景英川. 基于 XGBoost 模型的股骨颈骨折手术预后质量评分预测 [J]. *太原理工大学学报*, 2018, 49(1): 174 – 178.
- [10] Mavrovouniotis M, Muller F M, Yang S. Ant colony optimization with local search for dynamic traveling salesman problems [J]. *IEEE Transactions on Cybernetics*, 2017, 47(7): 1743 – 1756.
- [11] Ming F, Han Q, Yang X. Energy minimization for on-line real-time scheduling with reliability awareness [J]. *Journal of Systems & Software*, 2017, 12(7): 168 – 176.
- [12] Zhang Y, Yu Y, Zhang S, et al. Ant colony optimization for Cuckoo Search algorithm for permutation flow shop scheduling problem [J]. *Systems Science & Control Engineering An Open Access Journal*, 2019, 7(1): 20 – 27.
- [13] Ma Y N, Gong Y J, Xiao C F, et al. Path planning for autonomous underwater vehicles: An ant colony algorithm incorporating alarm pheromone [J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(1): 141 – 154.
- [14] Zang Z, Li Z, Dan Z, et al. Improving selection strategies in zeroth-level classifier systems based on average reward reinforcement learning [J]. *Journal of Ambient Intelligence & Humanized Computing*, 2018(2): 1 – 11.
- [15] Qin Z, Zhang C. An improved ant colony optimization algorithm with strengthened pheromone updating mechanism for constraint satisfaction problem [J]. *Neural Computing & Applications*, 2017(1): 1 – 12.
- [16] 程永强, 汪岚. 基于改进 MMAS 算法的堆垛机货物拣选路径优化 [J]. *延边大学学报(自然科学版)*, 2017(3): 70 – 73.