

文章编号:1671-6833(2019)04-0048-06

离散差分进化算法求解共享单车调度问题

汪慎文^{1,2}, 杨 锋^{1,2}, 徐 亮^{1,2}, 李美羽^{1,3}

(1.河北地质大学 信息工程学院,河北 石家庄 050031; 2.河北地质大学 人工智能与机器学习研究室,河北 石家庄 050031; 3.北京交通大学 交通运输学院,北京 100044)

摘 要:为了解决共享单车调度问题,设计了一种离散差分进化算法进行求解.系统地介绍了离散差分进化算法原理,并针对单车调度问题的求解,重新设计了算法中的个体编码、变异算子以及修补算子,使得算法在执行过程中能够对具体调度路径进行计算.结果表明,相比于贪心算法和蚁群算法,本研究算法解的质量较高且收敛速度较快,在共享单车调度等一系列的调度问题中具有一定的实用价值.

关键词:共享单车; 调度问题; 离散差分进化算法; 车辆路径问题

中图分类号: TU528.1 **文献标志码:** A **doi:**10.13705/j.issn.1671-6833.2019.04.022

0 引言

作为共享经济的一种新形态,共享单车不仅解决了市民“最后一公里”出行的问题^[1],更以无桩、便捷等优势,在一定程度上改变了人们的出行方式,缓解了城市交通拥堵.但是,由于在高峰时段,公司、居民区等客流量高发区域的潮汐现象^[2],容易出现共享单车的积聚或者短缺问题,因此需要快速构建出一种最佳的共享单车的调度方案,对共享单车进行调度.共享单车调度^[2]是根据整个区域的单车分布情况以及各区域的需求信息,依靠调度算法来使得整个区域的单车数目趋向于合理,即将共享单车从积聚区域及时运送到短缺区域.既不会因为单车的过多聚集而影响其他交通方式,也不会因为缺少单车而导致人们的出行困难,是一个将整个区域的共享单车数量和布局进行调整的过程.良好的调度算法不仅能够使共享单车调度体系正常运转,同时也能有效提升整个共享单车系统的服务水平.

1 离散差分进化算法原理及步骤

1.1 离散差分进化算法原理

离散差分进化算法^[3-4](discrete differential

evolution, DDE)框架与传统的进化算法^[5](differential evolution, DE)相同,首先在问题的定义域内随机产生第 0 代种群作为初始种群^[6];然后将每一个个体进行变异,变异的方式为在每一个个体上加上或者减去随机两个个体的差分量的缩放值,并将此个体作为变异个体,如有非法解则需要进行修补;然后对此变异个体与当前个体进行杂交操作,生成实验个体;最后将实验个体与当前个体进行竞争,较优者保留下来^[7].通过迭代,逐渐将种群引导到最优解位置.主要步骤如图 1 所示^[8].

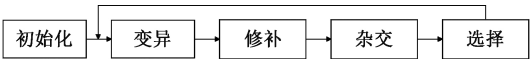


图 1 DDE 算法主要步骤

Fig.1 Main steps of DDE algorithms

不同之处在于,将其个体编码、变异算子和修补算子重新设计,使其适合求解共享单车调度问题.每个区域的编号作为每个个体的元素,将传统变异算子改造成可以进行离散变异的变异算子,同时,一旦产生非法解,则进行修补操作.

1.2 个体编码设计

考虑到所使用的区域编号为整数,笔者拟采用整数进行个体编码^[9].

收稿日期:2018-12-14; 修订日期:2019-03-09
基金项目:国家自然科学基金资助项目(61402481); 河北青年拔尖人才支持计划(冀字[2013]17号); 河北省教育厅自然科学基金重点项目(ZD2018083, ZD2018043, ZD2019134); 河北地质大学博士科研启动基金项目(BQ201322); 河北省科技创新引导计划项目(19970311D)
作者简介:汪慎文(1979—),男,湖北红安人,河北地质大学副教授,博士后,硕士生导师,CCF 会员,主要研究领域为智能计算、机器学习等, E-mail: wangshenwen@hgu.edu.cn.

每条可行线路编码成长度为 l 的个体 $\{x_{1,1}(G), x_{1,2}(G), \dots, x_{1,l}(G)\}$, 其中, $x_{i,k}(G)$ 表示第 G 代种群中的第 i 个个体的第 k 个分量. 其中, 分量表示的是区域的编号. 如个体为 $5 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 3$ 表示从配送中心 1 出发, 最后又返回配送中心 1 的一条调度路线. 个体编码长度 l 由式(1)表示.

$$l = \frac{T}{\text{shortest}(u, v)}, \quad (1)$$

式中: T 表示整个高峰期的时间; $\text{shortest}(u, v)$ 表示任两区域之间通行时间最小值.

1.3 变异操作设计

与传统的差分进化算法的变异操作不同, 本文设计的简单的变异算子如下^[8].

$$v_i(G) = x_{r1} + F(x_{r2}(G) - x_{r3}(G)), \quad (2)$$

式中: F 为缩放因子且 $F \in \{1, 2, \dots, n\}$. 在本实例中, 选取 $F = 1$, 示意图如图 2 所示^[10].

x_{r1}	5	2	3	4	3	2	5
x_{r2}	4	3	2	5	2	4	3
x_{r3}	2	4	5	3	4	3	5
v_i	7	1	0	6	1	3	3

图 2 变异操作示例图

Fig.2 Variation operation diagram

1.4 修补算子设计

相比其他优化问题, 共享单车调度问题对解的限制有所不同^[2]: ①相邻两个变量不能相同; ②个体中的每一维取值都必须在区域列表中. 因此, 修补操作设计步骤如下.

输入: $item$ 为个体, n 为染色体个数, R_{lower} 为定义域下界, R_{upper} 为定义域上界

```

1: function REPAIR( $item, n, R_{\text{lower}}, R_{\text{upper}}$ )
2:   for  $i = 1 \rightarrow n$  do
3:      $item[i] \leftarrow (item[i] - R_{\text{lower}}) \bmod (R_{\text{upper}} - R_{\text{lower}} + 1) + R_{\text{lower}}$ 
4:   end for
5:   for  $i = 2 \rightarrow n - 1$  do
6:     while  $item[i] = item[i - 1]$  or  $item[i] = item[i + 1]$  do
7:        $item[i] = \text{Random}(R_{\text{lower}}, R_{\text{upper}})$ 
8:     end while
9:   end for
10:  return  $item$ 
11: end function

```

1.5 DDE 算法流程

通过以上对 DDE 算法各个操作的描述, DDE 算法的流程如下.

步骤 1 初始化: 确定编码方式后, 选择合适的种群大小, 种群规模、 CR 、 F 值, 种群代数 $G = 0$, 初始化第 0 代随机种群, 并计算每个个体的适应度值.

步骤 2 变异操作: 种群代数 $G = G + 1$, 根据新设计的变异操作对 $X(G)$ 中的每个个体进行变异操作, 操作方式如上述变异操作所示.

步骤 3 修补操作: 根据新设计的修补算子对变异后的每个个体进行修补操作, 使得每个个体能够满足之前问题的两个限制条件.

步骤 4 杂交操作: 使用“bin”杂交方式对原本个体与变异个体进行操作, 生成实验个体.

步骤 5 选择操作: 通过新设计的适应度函数, 选择适应度值更大的个体, 从而得到下一代种群.

步骤 6 中止判断: 如果条件满足, 则中止当前算法执行, 并输出最优路径, 否则, 转步骤 2.

2 共享单车调度模型

2.1 问题描述

通常, 在早高峰期, 居住区域对共享单车需求量较大, 工作区域则单车到达量较多; 在晚高峰期则相反. 这种出行量和到达量的差异决定了各区域存在调度需求, 具体来看, 这种需求分为两种^[2]: 调入需求和调出需求. 因此共享单车的调度问题可描述为: 在早晚高峰期, 存在一定数量具有调度需求的区域, 从调度中心出发的一辆调度车在已知各区域位置、区域单车分布、调度车容量、各区域租还速率等信息, 要求在一定的约束条件下, 完成调度区域间的单车调度, 计算出调度路线和各区域的调度车辆数量.

2.2 调度过程

调度区域分为 3 种类型: 调度中心、出行区域以及到达区域. 调度中心无法存放单车, 也无单车调度需求, $t = 0$ 时刻调度车从调度中心空车出发, 前往各个区域进行调度; 出行区域的单车数量会随时间均匀减少, 当调度车抵达出行区域时, 按照一定的调度规则卸下一定数量的单车; 到达区域的单车数量会随时间均匀增多, 当调度车抵达到达区域时, 按照一定的规则装走一定数量的单车.

如果抵达某一区域时尚未超出调度时间段的范围, 则按照一定的调度规则完成对该区域的单车调度, 然后调度车前往下一区域; 如果超出调度时间段的范围, 则终止整个调度过程并返回调度中心.

鉴于主目标是满足用户需求,所以优化目标确定为实现最大限度的调度量,因为在系统规模一定的情况下,经过最大量的调度可实现较高的利用率,因此调度目标是使在调度时间段内调度车的总装卸量 Q 尽可能大。

为了表达更简洁,本文引入一些符号,其含义如表 1 所示。

表 1 符号说明
Tab.1 Symbol declaration

变量	含义
i	区域编号, $i \in \{1, 2, \dots, n\}$, $i = 1$ 表示调度中心
j	调度次数编号, $j \in \{1, 2, \dots, m\}$, m 表示最大调度次数
C	调度车的容量,即调度车能够容纳的最大单车数量
t_d	调度车装上或卸下一辆单车所需的时间
T	调度终止时刻
t_j	第 j 次调度的开始时刻
q_i	$t = 0$ 时刻 i 区域的单车数量
Q_i	$[0, T]$ 时间段内 i 区域单车数量的净变化量
F_i	到达区域 i 在调度终止时刻希望保持的单车数目
C_j	第 j 次调度开始时调度车上的单车数量
$n_{i,j}$	第 j 次调度开始时 i 区域单车的数量
$B_{i,j}$	第 j 次调度开始时 i 区域的调度需求
b_j	第 j 次调度实际装卸的单车数目
Q	调度车在整个调度过程中的总装卸量, $Q = \sum_{j=1}^m b_j $

2.3 调度规则

当第 j 次调度调度车抵达 i 区域时,首先计算 i 区域的当前单车数目 $n_{i,j}$,计算方法为:

$$n_{i,j} = \begin{cases} q_i, & j = 1; \\ \max\left\{0, n_{i,j-1} + (t_j - t_{j-1}) \cdot \frac{Q_i}{T}\right\}, & j > 1. \end{cases} \quad (4)$$

然后计算 i 区域对单车的需求,计算方法为:

$$B_{i,j} = \begin{cases} n_{i,j} + \frac{Q_i}{T} \cdot (T - t), & \text{出行区域;} \\ n_{i,j} + \frac{Q_i}{T} \cdot (T - t) - F_i, & \text{到达区域,} \end{cases} \quad (5)$$

当 $B_{i,j} < 0$ 时,表示该区域车辆短缺,需要补充单车.若调度车上的单车数量不足以满足该区域的需求,则将调度车上所有车卸下;若调度车上的车辆足以满足该区域的需求,则从调度车上卸下 $-B_{i,j}$ 辆单车,如式(6)所示。

$$b_j = \max\{-C_j, B_{i,j}\}. \quad (6)$$

当 $B_{i,j} > 0$ 时,表示该区域车辆过剩,需要装走多余单车.若调度车上的剩余空间不足,则将调

度车装满;若调度车的剩余空间足够,则将 $B_{i,j}$ 辆单车装上调度车,如式(7)所示。

$$b_j = \min\{C - C_j, B_{i,j}\}. \quad (7)$$

3 实验及结果

3.1 参数设置

总调度时间 $T = 60$ min,调度车容量 $C = 20$,装卸一辆单车的时间 $t_d = 0.2$ min.一共有 5 个区域,区域 1 为调度中心,区域 2、3 为出行区域,区域 4、5 为到达区域.各个区域间的最短通行时间如表 2 所示.各区域的初始车辆变化如表 3 所示。

表 2 各区域最短通行时间

Tab.2 Shortest passage time between different zones					
起点 编号	终点 编号	通行时 间/min	起点 编号	终点 编号	通行时 间/min
1	2	6	2	4	7
1	3	12	2	5	6
1	4	10	3	4	4
1	5	5	3	5	7
2	3	8	4	5	5

* 注:本文的双向距离相同

表 3 各区域初始车辆数和车辆变化

Tab.3 The initial number of vehicles and vehicle changes in each zone			
区域 编号	初始 单车 数量 q_i	调度时间段内单车 数量的净变化量 Q_i	调度结束时 期望维持的 车辆数目 F_i
1	0	0 (无)	—
2	10	-40 (净出行 40 辆)	—
3	15	-25 (净出行 25 辆)	—
4	10	40 (净到达 40 辆)	20
5	10	30 (净到达 30 辆)	15

设置种群规模 NP 为 30,每个个体表示一条调度路径,维度 D 设置为 15,缩放因子 F 为 1,杂交概率 CR 为 0.6,迭代次数为 100 次.执行过程中会判断每个个体的有效长度,保证调度路径可以在规定时间内完成。

3.2 实验结果及分析

DDE 算法收敛曲线如图 3 所示.图 3 表明,种群在第 2 代左右得到最优解,在 50 代左右所有个体收敛到最优值,最终平均调度量为 85.9,收敛速度较快。

最优的调度路线、调度车辆数量和调度时间节点如表 4 所示,其中,调度顺序为调度次数,

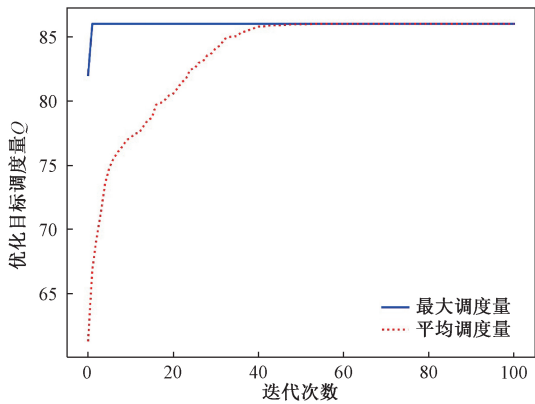


图 3 DDE 算法收敛曲线图

Fig.3 The convergence curve of DDE algorithm

0 表示初始状态,区域编号为当前调度次数下调度车辆所在的区域,调度量为此次调度下调度车辆的装卸量,正数表示装车,负数表示卸车,总调度量表示截至当前次数下已经调度的车辆总和,包括装车和卸车,最终得到的路径为 1→5→2→4→3→2→4→5→1,总调度量 $Q=86$.

DDE 算法得到的解所模拟的调度过程如图 4 所示.其中,红色方框表示每一时刻小车所在的区域,方框内的数表示当前时刻调度车辆的装载量.红色线段表示调度车辆的路径变化,即行驶路径.其中左上角图例中的 $n_{i,j}$ 和 $B_{i,j}$ 的含义与表 1 中的变量说明一致.每个子图下方标示了该子图所处的时间点 t 和总调度量 Q .该图的每一张子图都为一个小图,如第一张子图表示初始状态,当前时刻 $t=0$,总调度量 $Q=0$,调度车辆当前所处的区域为调度中心 1,每个区域的当前单车数目 $n_{i,j}$ 和需求量 $B_{i,j}$ 在其下方显示.

表 4 DDE 算法调度过程

Tab.4 The scheduling process of DDE algorithm

调度顺序	区域编号	调度车抵达该区域的时间/min	调度量 b_j	总调度量 Q
0	1	0	0	0
1	5	5	+13	13
2	2	13.6	-13	26
3	4	23.2	+20	46
4	3	31.2	-10	56
5	2	41.2	-10	66
6	4	50.2	+10	76
7	5	57.2	+10	86

3.3 与其他算法对比

(1) 应用贪心算法,贪心策略为每次选择使单位时间调度数量最大的区域作为前往的下一个区域,所得结果如表 5 所示,最终的调度量 $Q=84$.

表 5 贪心算法调度过程

Tab.5 The scheduling process of greedy algorithm

调度顺序	区域编号	调度车抵达该区域的时间/min	调度量 b_j	总调度量 Q
0	1	0	0	0
1	5	5	+13	13
2	2	13.6	-13	26
3	4	23.2	+20	46
4	3	31.2	-10	56
5	4	37.2	+10	66
6	2	46.2	-9	75
7	5	54	+9	84

(2) 应用蚁群算法^[11],重复运行 50 次,所得解的平均质量同 DDE 算法的对比如表 6 所示;其中两种算法一次运行的情况对比如图 5 所示.图 5 的横纵坐标以及图的各项说明与图 4 相同.

表 6 蚁群算法与 DDE 算法得到的平均调度量对比

Tab.6 Comparison of average scheduling volume of ant colony optimization and DDE algorithm

算法	最优值	最差值	平均值	标准差	成功次数/试验次数
蚁群算法	81.45	77.83	79.52	1.223	50/50
DDE	85.97	85.84	85.90	0.043	50/50

图 6 表明,在相同的较小规模问题下,蚁群算法得到的最优解与 DDE 算法相同,但相比蚁群算法,DDE 算法得到的解的平均质量更高,且收敛速度更快.表 6 表明,蚁群算法和 DDE 算法均能稳定求解共享单车调度问题,图 6 的情况具有一般性和代表性.

综合比较之下,DDE 算法在这个问题中具有收敛快、解的质量高的优点.

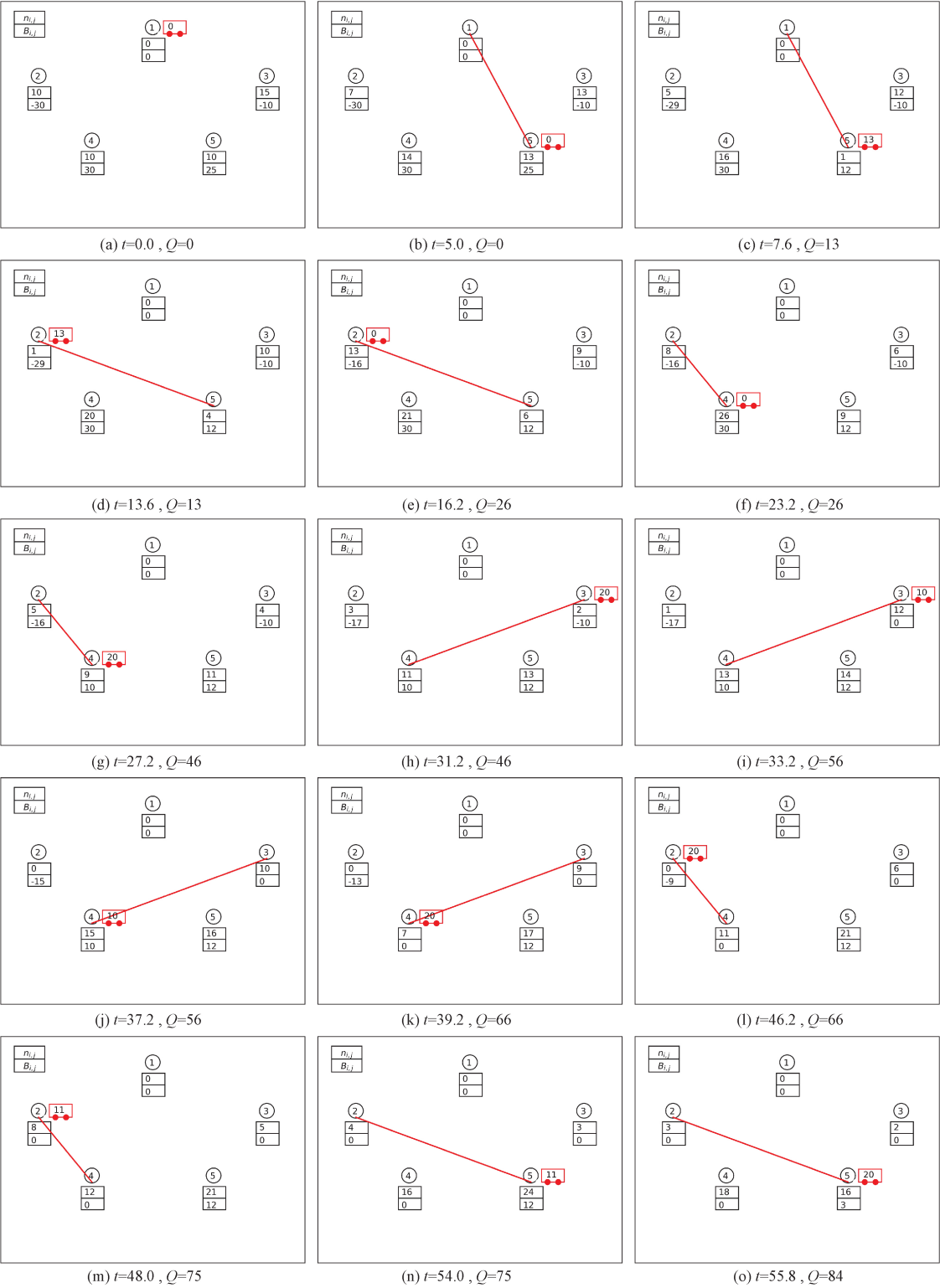


图 4 DDE 算法模拟调度过程

Fig.4 The specific route finding process of DDE algorithm

4 结论

笔者针对共享单车调度问题,设计了相应的 DDE 算法进行求解.实验结果表明,DDE 算法可以稳定求解共享单车调度问题,相比贪心算法和

蚁群算法,DDE 算法在较小规模下求解共享单车调度问题时解的质量更优、收敛速度更快.下一步的工作将优化本身的调度模型,并逐步将问题的规模扩大,使得模型更符合现实生活.

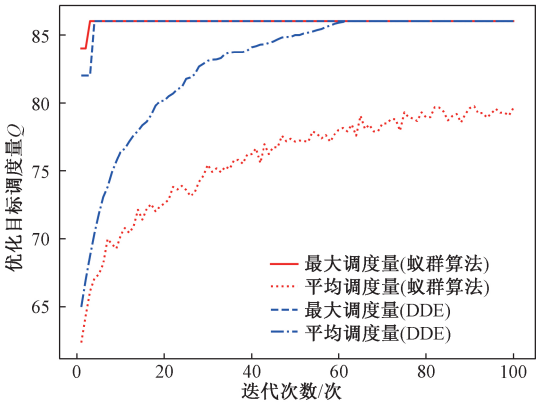


图 5 蚁群算法与 DDE 算法收敛曲线对比图

Fig.5 Comparison of convergence curves of ant colony optimization and DDE algorithm

参考文献:

[1] 高楹,宋辞,舒华,等.北京市摩拜共享单车源汇时空特征分析及空间调度[J].地球信息科学学报,2018,20(8):1123-1138.

[2] 周传钰. 共享单车投放量测算和调度方法研究[D]. 北京:北京交通大学交通运输学院,2018.

[3] 薛羽,庄毅,顾晶晶,等. 自适应离散差分进化算法策略的选择[J]. 软件学报,2014,25(5):984-996.

[4] AKG ÜNGÖR A P, KORKMAZ E. Estimating traffic accidents in turkey ssing differential evolution algorithm[J]. Selected scientific papers-journal of civil engineering,2017,12(1):75-84.

[5] 梁静,周钦亚,瞿博阳,等. 基于混合策略的差分进化算法[J].郑州大学学报(工学版),2013,34(5):59-62.

[6] 程适,王锐,伍国华,等. 群体智能优化算法[J]. 郑州大学学报(工学版),2018,39(6):1-2.

[7] 曾冰,王梦雨,高亮,等.改进鲸鱼群算法及其在炼钢连铸调度中的应用[J].郑州大学学报(工学版),2018,39(6):14-22.

[8] 汪慎文,丁立新,张文生,等.差分进化算法研究进展[J].武汉大学学报(理学版),2014,60(4):283-292.

[9] 侯玲娟,周泓,梁春华. 不确定需求和旅行时间下的车辆路径问题[J]. 计算机集成制造系统,2011,17(1):101-108.

[10] SAKR W S, EL-SEHIEMY R A, AZMY A M. Adaptive differential evolution algorithm for efficient reactive power management [J]. Applied soft computing,2017,53:336-351.

[11] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation,1997,1(1):53-66.

Discrete Differential Evolution Algorithm for Solving Free-floating Bike-Sharing System Scheduling Problem

WANG Shenwen^{1, 2}, YANG Feng^{1, 2}, XU Liang^{1, 2}, LI Meiyu^{1, 3}

(1. School of Information Engineering, Hebei GEO University, Shijiazhuang 050031, China; 2. Laboratory of Artificial Intelligence and Machine Learning, Hebei GEO University, Shijiazhuang 050031, China; 3.School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China)

Abstract: In order to solve the free-floating bike-sharing system scheduling problem, a discrete differential evolution algorithm was designed to solve the problem. After introducing the principle of discrete differential evolution algorithm systematically, the individual coding, mutation operator and repair operator were redesigned to solve the free-floating bike-sharing system scheduling problem, so that the specific scheduling path could be calculated during the execution of the algorithm. The results showed that compared with greedy algorithm and ant colony optimization algorithm, the proposed algorithm had higher quality and faster convergence speed, and had certain practical value in a series of scheduling problems such as shared bicycle scheduling.

Key words: free-floating bike-sharing system; scheduling problem; discrete differential evolution algorithm; vehicle routing problem