

# 百度地图 API 及蚁群算法在车辆 路径问题中的应用

林文如, 林国福

(闽江学院计算机科学系, 福建 福州 350121)

**摘要:** 在车辆路径问题(VRP)的研究中,通常采用节点间直线距离之和作为最短路径最优求解的数据基础,而节点间直线距离和与道路(弯曲、单向行驶等情况)实际距离通常相去甚远,使得其最优路径安排难于真正应用在实际需求中。为此,首先通过百度地图 API 获取节点间实际道路的距离数据,然后通过增加约束条件、修改节点间距离的计算、更换选择策略、调整信息挥发因子 $\rho$ 取值等方法改进基本蚁群算法,最后将百度地图 API 获取的道路实际距离数据提交给改进蚁群算法求解。同时,基于 PHP、Ajax 编程求解程序并运行,实验表明,该方法具有可行性与实用性,完善了 VRP 问题从理论研究过渡到实际应用的最后一个环节。

**关键词:** 百度地图; 车辆路径问题; 蚁群算法

**中图分类号:** TP391.9

**文献标识码:** A

**文章编号:** 1009-7821(2015)02-0082-07

## The application of Baidu maps API and ACO in VRP

LIN Wen-ru, LIN Guo-fu

(Department of Computer Science, Minjiang University, Fuzhou, Fujian 350121, China)

**Abstract:** In the vehicle routing problem(VRP) research, the linear distance of nodes was usually regarded as datasets for solving the optimal path, but it's difficult to apply in the actual demand because the gap is even wider between the linear distance and the actual road distance. So firstly, the datasets of actual road distance was acquired by using Baidu maps API; and then the basic ant colony algorithm was improved by adding constraints, modifying nodes' distance datasets, replacing selection strategy, adjusting  $\rho$  (the information volatilization factor) value; lastly, the actual road distance datasets were submitted to the improved ant colony algorithm for solving. Moreover, based on PHP and Ajax, the solver program was developed, the experimental results show that this method is feasible and practical, achieve the last step from theoretical research to practical application for VRP.

**Key words:** Baidu maps; vehicle routing problem; ant colony optimization

收稿日期: 2014-10-15

基金项目: 福建省中青年教育科研项目(JB13165)

作者简介: 林文如(1980-),男,福建尤溪人,闽江学院计算机科学系讲师。

林国福(1980-),男,福建莆田人,闽江学院计算机科学系高级实验师。

物流配送是物流管理中的重要一环,它是指按照客户的订货要求,在配送中心进行分货、配货,并调度车辆及时将货物送达客户<sup>[1]</sup>。物流配送车辆的调度效果对配送成本、速度、效益影响很大,已经成为制约物流业发展的要素之一<sup>[2]</sup>。而车辆调度研究中最核心的是车辆路径问题(vehicle routing problem, VRP),该问题由 Dantzig 和 Ramser 于 1959 年首次提出,一直是组合优化领域的热点和前沿问题<sup>[3]</sup>。其在邮政投递和物流配送等许多领域都有重要的应用。特别是伴随近年来物流业的不断发展,配送范围不断扩大,任务量不断增加,该问题越来越受到学者的关注<sup>[4]</sup>。

近年来,国内外许多学者对物流配送问题进行了大量的研究,这些研究主要集中在理论及算法领域方面。研究者使用启发式算法和智能算法<sup>[5-6]</sup>(如:遗传算法、蚁群算法和模拟退火算法等),或者是在智能算法优化过程中加入优化策略<sup>[7]</sup>,以构造混合智能算法来求解物流配送问题。在 VRP 的软件系统应用研究方面,随着 GIS 的快速发展,为利用实际地图路径数据实现车辆调度提供了便利,具体而言就是将地图路径数据和物流订单信息数据结合,实现配送决策产生,这方面的研究是开发车辆调度软件的重要工作之一<sup>[8-9]</sup>。而专业 GIS 系统,如国外的 ArcGIS、MapInfo、GeoMedi 和国内的 MapGIS、GeoStar、TopMap 等,其功能强大,但对开发能力、开发成本要求高,造成其应用门槛高<sup>[10]</sup>。而像百度地图 API 这样第三方提供的 GIS 应用服务,因其免费开放使用、学习门槛不高、调研简单、功能强大等特点,成为非专业 GIS 人员应用的首选。

在众多仿真实例中,关于任意两个客户节点之间的距离的计算采用的是直线距离,未采用实际交通条件下(弯曲道路、有些路段单向行驶等)的道路距离。为了适应实际应用的需要,本研究将通过百度地图 API 提供的 GIS 应用服务接口获得客户节点间的实际路径距离信息,并将该信息提交给改进后的蚁群算法进行 VRP 问题的求解,完善了 VRP 问题从理论研究到实际应用的最后一个环节。

## 1 方法流程

本文方法流程如图 1 所示。首先,根据订单信息,将客户地址及请求参数发送到百度地图服务器;百度地图服务器返回请求状态及路径数据(提取订单中所有任意两个节点,包括配送中心地点和客户地点之间的有向最短道路距离);将返回的路径数据提交给改进蚁群算法进行求解;最后在客户端输出显示求解结果。



图 1 系统流程图

Fig. 1 System flow

## 2 蚁群算法基本模型

Dorigo M 等人首先提出了蚁群算法<sup>[11]</sup>。其主要特点就是:通过正反馈、分布式协作来寻找最优路径。同时,该算法还被用于求解 Job-Shop 调度问题、二次指派问题以及多维背包问题等,显示了其适用于组合优化类问题求解的优越特征<sup>[12]</sup>。

在求解模型问题前,引入如下符号定义:蚂蚁数量  $K$ ; 客户个数  $L$ ; 客户集合  $R(R = \{i\} | i = 1, 2, \dots, L)$ ; 配送中心  $R_0$ ;  $t$  时刻客户  $i$  到客户  $j$  线路上的信息素量  $\tau_{ij}(t)$ ; 启发信息量  $\eta_{ij} = 1/d_{ij}$ ,  $d_{ij}$

表示客户  $i$  和客户  $j$  之间的距离; 每只蚂蚁在一次巡回期间仅访问每个客户节点一次, 用禁忌表  $\text{tabu}(k)$  控制,  $\text{tabu}(k)$  表示禁忌表中第  $k$  只蚂蚁已访问过的所有客户;  $\text{allowed}(k)$  表示第  $k$  只蚂蚁下一步允许访问的客户, 即该蚂蚁还未访问过的所有客户,  $\text{allowed}(k) = R - \text{tabu}(k)$ ; 蚂蚁在其行进路线上释放信息素. 因此,  $t$  时刻位于客户  $i$  的蚂蚁  $k$  将客户  $j$  作为下一访问节点的转移概率按式 (1) 计算<sup>[13]</sup>.

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ik}(t)]^\beta}{\sum_{s \in \text{allowed}(k)} [\tau_{is}(t)]^\alpha \times [\eta_{is}(t)]^\beta}, & j \in \text{allowed}(k) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

式 (1) 中,  $\alpha$  为信息启发式因子, 其值越大说明蚂蚁对信息素越敏感;  $\beta$  为期望启发式因子, 其值越大说明蚂蚁对线路的长短越敏感.

信息素的更新是影响蚁群算法收敛性的重要因素之一<sup>[14]</sup>, 当所有蚂蚁都已建立起访问的路径后, 路径上各条边的信息素将按如下规则进行更新:

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij}(t), \quad (2)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \tau_{ij}^k(t), \quad (3)$$

式 (2) 和式 (3) 中  $\rho$  为信息素挥发因子  $1-\rho$ , 则表示信息素残留因子, 为了防止信息的无限积累,  $\rho$  的取值范围:  $0 \leq \rho < 1$ ;  $\tau_{ij}(t)$  和  $\tau_{ij}(t+n)$  分别表示  $t$  时刻和  $t+n$  时刻在路径上残留的信息素数量; 表示在本次循环中路径  $(r_i, r_j)$  上信息素的增量  $\Delta\tau_{ij}^k(t)$  表示第  $k$  只蚂蚁在本次循环中留在路径  $(i, j)$  上的信息素量.

根据信息素的更新策略不同, Dorigo M 提出了 3 种不同的基本蚁群算法模型, 分别称之为 Ant-Cycle 模型、Ant-Quantity 模型和 Ant-Density 模型, 其主要差别在于  $\Delta\tau_{ij}^k(t)$  的求法不同. 由于 Ant-Cycle 模型利用的是全局信息, 求解 VRP 问题时性能较好<sup>[15]</sup>. 所以, 通常在蚁群算法中采用该模型作为计算方法, 表达式如下:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{第 } k \text{ 只蚂蚁经过路径 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (4)$$

式 (4) 中,  $Q$  为常数, 表示信息素强度, 它在一定程度上影响算法的收敛速度;  $L_k$  表示第  $k$  只蚂蚁在本次循环中的访问路径总长度.

### 3 蚁群算法的改进

#### 3.1 增加约束条件

针对物流车辆配送路径问题, 增加以下两个约束条件:

(1) 每条配送路线上的客户货物量之和不超过汽车总载重量:

$$\sum_{i=1}^{n_R} q_{r_i} \leq Q_k, n_k \neq 0, \quad (5)$$

式 (5) 中,  $q_i$  是客户点  $i$  的货物需求量, 其中  $i = 1, 2, \dots, L$ ,  $L$  为客户点总数;  $Q_k$  是车辆  $k$  的最大装载量, 其中  $k = 1, 2, \dots, K$ ,  $K$  为车辆总数;  $n_k$  是车辆  $k$  配送的客户总数, 当  $n_k = 0$  时, 表示车辆  $k$  没有参与配送.

(2) 每个客户的货物必须且只能由一辆汽车来完成:

$$R_{k_1} \cap R_{k_2} = \Phi, k_1 \neq k_2, \quad (6)$$

式(6)中,  $R_k$  是车辆  $k$  配送的客户点的集合.

### 3.2 节点间距离计算方法改进

去掉基本蚁群算法中用勾股定理计算任意两个节点间的距离的方法,改用百度地图 API 返回的 json 封装得到任意两个节点间的实际距离矩阵. 通过该改进,不仅可以提高运行速度,而且提高了系统的计算精度(百度地图 API 返回的数据精度达到 m 的级别).

主要涉及到百度地图 Route Matrix API 的使用,该接口适用于仅获取线路距离和时间,无需获取详细线路信息或者需要同时获取多起点、多终点线路距离等的情况<sup>[16]</sup>,该接口的服务地址 <http://api.map.baidu.com/direction/v1/routematrix>,接口请求参数包括 origins(起点)、destinations(目的地)、mode(导航模式,用 driving 驾车模式)、output(返回数据格式,用 json)、coord\_type(坐标类型,用百度经纬度坐标)、tactics(导航策略,有:不走高速、最少时间、最短路径 3 个可选项,本文设置为最短路径策略)、ak(用户的访问权限)、sn(用户的权限签名)、timestamp(时间戳,与 sn 配合使用). 数据返回参数包括 status(状态码,0:成功 2:参数错误 3:权限或配额校验失败)、message(状态码对应的信息)、info(版权信息)、result(返回的结果数组). 这里得到的任意两节点间的距离 dij 是实际道路有向距离,而非传统理论研究中使用勾股定理计算两点间的直线距离.

### 3.3 选择策略改进

在基本蚁群算法中,蚂蚁通常会将通过式(1)计算的最大概率节点作为下一个要访问的节点,这样会导致每只蚂蚁选择的下一个点都是同一个点,进而导致所有蚂蚁找到的路径都是相同的,使蚂蚁失去探索新路径的机会,算法陷入停滞. 因此有必要采用轮盘选择法<sup>[17]</sup>对一般蚁群算法的选择策略进行改进. 具体做法是:在  $[0, 1]$  之间取一个随机数  $R$ , 然后用  $R$  减  $P_{ij}^k(t)$ , 如果减后的结果小于等于 0 就选当前节点作为下一个访问节点,如果减去后还大于 0,就继续再减去,直到减去后的结果小于等于 0,然后用最后减去时的那个概率值对应的点作为下一个访问节点.

### 3.4 信息素挥发因子 $\rho$ 的取值改进

按照基本蚁群算法, $\rho$  是一个常量,其值的设定对求解结果影响较大. 如果  $\rho$  值太大,则会使未搜索过的路径被选择的概率相对减小,影响全局搜索能力;而如果  $\rho$  值太小,又会影响算法的收敛速度. 因此在改进算法中将对  $\rho$  作适当调整. 在算法初期,我们希望算法能尽快找到较优解,因此  $\rho$  要比较大,增大信息浓度的影响,加快算法收敛速度;而当算法停滞不前时,要减小  $\rho$ ,从而减小信息素对蚁群的影响,增大蚁群对解空间的搜索,以脱离局部最优解的束缚<sup>[18]</sup>.

$$\rho(t+1) = \begin{cases} \max\{\lambda \times \rho(t), \rho_{\min}\}, & r = r_{\max} \\ e(t), & \text{otherwise}, \end{cases} \quad (7)$$

式(7)中, $r$  表示连续没有进化的循环的次数,  $r_{\max}$  是一个常量,  $\lambda \in (0, 1)$  是一个常量,控制  $\rho$  衰减速度,  $\rho_{\min}$  是  $\rho$  的最小值,防止  $\rho$  过小影响收敛速度. 当  $r$  达到预先设置的一个数值  $r_{\max}$  时,就减小  $\rho$ ,  $r$  重新计数,如此反复,直至  $\rho$  达到预设最小值  $\rho_{\min}$  为止.

## 4 实例分析

为此,程序基于 B/S 模式开发,服务端采用 PHP 语言开发,主要完成算法求解功能;客户端

采用 Ajax 技术与服务端实现异步数据交互,开发语言包括 HTML、JavaScript、CSS、json 等。以福建省福州市某物流公司的物流配送订单为例,配送中心位置  $R_0$  (地址:仓山区金山公园),需给周边 10 个客户配送,客户地址及货物量分别为:仓山万达广场(1.4 t)、仓山区浦上大道红江小区(0.3 t)、江南水都名筑(1.5 t)、融信大卫城(1.2 t)、仓山区金桔路水印长天(0.8 t)、仓山区万兴隆连锁超市(1.5 t)、仓山区洪塘路利家超市(1.4 t)、仓山区金山大道江南名城(1.2 t)、仓山区金丰综合服务大楼(1.6 t)、建新中路良友家居商场(1.9 t)。假定配送中心的车辆充足,能满足任务要求,每台车辆载重量为 5 t,现要求规划合适的车辆安排及行车路线,使配送运输总路程最短。

通过百度地图 API 计算得到各节点(1 个配送点及 10 个客户)的实际道路距离如表 1 所示。根据配送车辆的总载重量 5 t 和总运量重 12.8 t,初步确定可以选择配送车辆数,即蚂蚁数量  $K=3$  进行配送。初始参数设置为:信息启发因子  $\alpha=1$ ,期望启发因子  $\beta=2$ ,信息素挥发因子  $\rho=0.95$ ,最大迭代次数  $N_{\max}=200$ ,蚂蚁数量  $K=3$ 。运行程序无数次,并去除类似结果,表 2 仅列出 13 次具有代表性的求解结果,均满足两个约束条件(式(5)、(6)),其中最优解为第 7 次计算结果,其最短实际道路距离为 36.408 km,车辆 1 的运行路线为:0-10-1-3-0,车辆 2 的运行路线为:0-5-7-8-0,车辆 3 的运行路线为:0-6-9-2-4-0。

表 1 节点间实际道路距离表

Tab. 1 The actual road distance between nodes

单位: km

$i$	节点 $j$										
	0	1	2	3	4	5	6	7	8	9	10
0	0	3.991	3.593	4.44	1.781	2.069	1.777	3.336	0.749	3.032	1.765
1	3.377	0	2.347	2.395	2.915	4.708	4.992	6.467	3.889	4.345	2.939
2	3.626	3.494	0	3.64	2.452	4.245	5.230	6.719	4.163	5.181	3.188
3	3.97	2.114	4.253	0	4.484	5.218	5.001	6.019	3.898	4.259	3.532
4	1.432	4.067	2.438	4.267	0	2.495	3.261	4.820	2.233	4.293	2.300
5	2.417	5.477	5.396	5.765	3.593	0	2.325	3.843	2.117	4.343	3.629
6	2.088	5.229	5.178	5.517	3.264	2.206	0	1.777	1.173	2.804	3.35
7	3.662	6.722	6.702	6.505	4.838	3.989	1.783	0	2.725	2.827	4.874
8	2.379	4.008	4.006	4.344	3.130	3.317	2.493	4.023	0	2.510	2.178
9	3.350	4.599	5.253	4.716	4.377	4.229	2.811	2.867	2.326	0	3.425
10	1.837	3.544	1.829	3.641	0.953	3.543	3.441	4.930	2.374	3.392	0

## 5 讨论

从表 1 中可以看出,  $d_{ij}$  不一定和  $d_{ji}$  相等,这是因为这些数据是由百度地图 API 计算获得,因此得到的任意两节点间的距离是实际道路距离,而非传统理论研究中使用勾股定理计算两点间的直线距离,另外,实际道路条件还包括是否走高速、城市某些路段是否单向行驶、某些交叉路口是否禁止左拐、禁止右拐等,而通过百度地图 API 获取的节点间路径距离恰恰考虑了这些信息,因此这样返回的路径数据提交给蚁群算法求解更具实际应用意义。

根据本文实例,发现若按表 2 中项目 4(节点间直线行驶路程)归类的话,有 4 种最短路程结果,分别是:34.778(表 2 中 1~5 行)、35.555(表 2 中 6~9 行)、35.221(表 2 中 10~11 行)、35.666 km(表 2 中 12~13 行),若按节点间直线行驶路程作为最优评判标准的话,35.221 km(表 2

中 10 ~ 11 行) 应是最优求解结果 ,但其对应的实际道路行驶路程未必是最优求解结果( 分别达到 43. 062、42. 866 km) 这说明在 VRP 问题的实际应用中 ,使用节点间直线行驶最短路程作为最优解是不科学的 ,应使用本文所提的实际道路行驶最短路程作为最优解.

表 2    求解结果对比表

Tab. 2    The solution results contrast

计算 次序	车辆 1 配送路线	车辆 2 配送路线	车辆 3 配送路线	节点间直线 行驶路程 /km	实际道路 行驶路程 /km	3 辆车装载 货物量 /t
1	0-5-4-10-0	0-8-7-3-0	0-2-9-6-1-0	34. 778	45. 237	3. 9/4. 1/4. 8
2	0-10-4-5-0	0-8-7-3-0	0-2-6-9-1-0	34. 778	42. 198	3. 9/4. 1/4. 8
3	0-10-4-5-0	0-8-7-3-0	0-1-6-9-2-0	34. 778	46. 261	3. 9/4. 1/4. 8
4	0-5-4-10-0	0-8-7-3-0	0-2-6-9-1-0	34. 778	44. 649	3. 9/4. 1/4. 8
5	0-10-4-5-0	0-8-7-3-0	0-2-9-6-1-0	34. 778	42. 786	3. 9/4. 1/4. 8
6	0-10-1-3-0	0-5-7-8-0	0-4-2-9-6-0	35. 555	36. 989	4. 8/3. 4/4. 6
7	0-10-1-3-0	0-5-7-8-0	0-6-9-2-4-0	35. 555	36. 408	4. 8/3. 4/4. 6
8	0-3-1-10-0	0-5-7-8-0	0-4-2-9-6-0	35. 555	36. 645	4. 8/3. 4/4. 6
9	0-10-1-3-0	0-8-7-5-0	0-4-2-9-6-0	35. 555	37. 151	4. 8/3. 4/4. 6
10	0-1-10-0	0-2-9-6-3-0	0-4-5-7-8-0	35. 221	43. 062	3. 3/4. 9/4. 6
11	0-1-10-0	0-2-6-9-3-0	0-8-7-5-4-0	35. 221	42. 866	3. 3/4. 9/4. 6
12	0-8-7-5-0	0-4-2-1-10-0	0-3-6-9-0	35. 666	39. 262	3. 4/4. 8/4. 6
13	0-8-7-5-0	0-4-2-1-10-0	0-3-9-6-0	35. 666	37. 265	3. 4/4. 8/4. 6

由于本文实例的节点间实际地理位置较接近( 7 km 以内) 及客户节点较少( 10 个客户) 的原因 ,表 2 中关于“节点间直线行驶路程”和“实际道路行驶路程”的对比差距不太明显. 若在实际应用中 ,随着配送客户数量的增多且地理位置较分散的情况下 ,二者之间的差距会更明显 ,更能凸显使用实际道路行驶最短路程作为最优解的优越性.

另外 ,关于改进后的蚁群算法收敛性方面 ,本文实例在迭代次数达到 50 次以上基本趋于收敛稳定 ,如图 2 所示 ,之所以出现进化不太明显 ,依然是本文实例的节点间实际地理位置较接近的原因. 而当客户数量增加到 20 个后( 实际配送水平) ,迭代 200 次左右就基本收敛稳定 ,进化较为明显 ,如图 3 所示. 说明算法的收敛性较好、可接受.

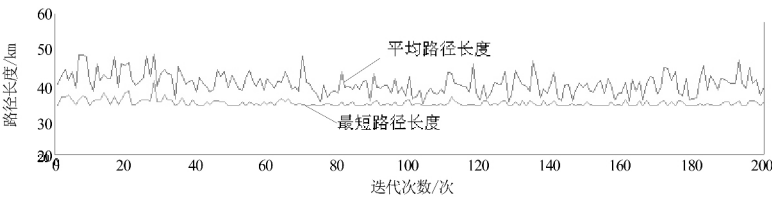


图 2    本文实例的收敛进化图

Fig. 2    Convergent evolution diagrams in this paper

6    结语

在追踪前人的研究中发现 ,采用节点间直线距离之和作为最短路径最优求解存在难于应用的不足 ,因此提出用百度地图 API 获取的道路节点间实际距离 ,并提交给改进蚁群算法进行最优

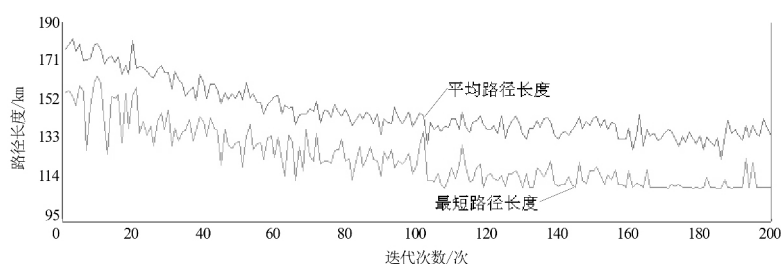


图3 客户数增加到20个的收敛进化图

Fig.3 20 customers convergent evolution diagram

求解的方法,最后通过实例分析,表明该方法具有可行性及实际应用价值。

#### 参考文献:

- [1] 杨燕霞,伍岳庆,姚宇,等.带时间窗车辆调度问题的启发式算法研究与应用[J].计算机应用,2013,33(S1):59-61.
- [2] 唐俊.时间窗约束下的配送车辆调度问题研究[J].计算机工程与应用,2011,47(21):243-248.
- [3] Dantzing G, Ramser J. The truck dispatching problem[J]. Management Science, 1959, 10(6): 80-91.
- [4] 谈晓勇,刘秋菊.应急配送车辆调度优化研究综述与展望[J].计算机应用研究,2012,29(9):3212-3215, 3220.
- [5] 孙丽君,胡祥培,王征.车辆路径规划问题及其求解方法研究进展[J].系统工程,2006,24(11):31-37.
- [6] Asvin G, Volker G. A general vehicle routing problem[J]. European Journal of Operational Research, 2008, 191: 650-660.
- [7] 王旭坪,张凯,胡祥培.基于模糊时间窗的车辆调度问题研究[J].管理工程学报,2011,25(3):148-153.
- [8] 杨湘燕.基于WebGIS的物流配送路径规划系统的设计与实现[D].厦门:厦门大学,2009.
- [9] 袁建清,修建新,王泽彬.基于GPS/GIS/GPRS技术的动态车辆调度系统设计与实现[J].哈尔滨师范大学自然科学学报,2010,26(6):67-70.
- [10] 杨森.RFID射频识别技术在物流车辆调度系统中的应用[J].物流技术,2014,33(4):338-341.
- [11] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-66.
- [12] Juyoung W, Byun I K, Seongbae K. The rollon-roll of waste collection vehicle routing problem with time windows[J]. European Journal of Operational Research, 2013, 224: 466-476.
- [13] 段海滨.蚁群算法原理及其应用[M].北京:科学出版社,2005:85-97.
- [14] 刘哲,李建国.基于带时间窗口车辆路径问题的蚁群算法[J].控制工程,2006,13(2):127-134.
- [15] 张潇,王江晴.混合蚁群算法在车辆路径问题中的应用[J].计算机工程,2011,37(24):190-192.
- [16] 百度地图Web服务.Route matrix API手册[EB/OL].(2013-12-19)[2014-10-20].<http://developer.baidu.com/map/index.php?title=webapi/route-matrix-api>.
- [17] Nagy G, Salhi S. Heuristic algorithms for the single and multiple depot vehicle routing problem with pickups and deliveries[J]. European Journal of Operational Research, 2005, 162: 126-141.
- [18] 张维泽,林剑波,吴洪森,等.基于改进蚁群算法的物流配送路径优化[J].浙江大学学报:工学版,2008,42(4):574-579.

[责任编辑:唐诚焜]