

# 基于改进蚁群算法的物流配送路径优化

段爱民, 陈泽琳, 陈海波

(华南理工大学 计算机科学与工程学院, 广东 广州 510510)

**摘 要:** 随着社会的不断进步, 配送车辆最短路径优化问题已广泛应用于交通运输、网络购物、物流配送等与生产生活息息相关的问题, 然而配送车辆路径优化的计算比较复杂。文中建立在带约束条件的多车辆物流配送问题模型的基础上, 运用改进的蚁群算法解决物流配送过程中的路径选择问题。通过对信息素的全局和局部更新规则进行改进, 和传统的最值蚁群算法进行比较, 算法的收敛速度和全局搜索能力得到提高。文中最后成功将改进后的蚁群算法应用于多车辆物流调度路径优化问题。结果表明该优化算法性能更优。

**关键词:** 蚁群算法; 蚁群系统; 物流配送; 路径优化

**中图分类号:** TP392

**文献标识码:** A

**文章编号:** 1673-629X(2011)12-0178-04

## Path Optimization for Logistics Distribution Based on Improved Ant Colony Algorithm

DUAN Ai-min, CHEN Ze-lin, CHEN Hai-bo

(College of Computer Science and Engineering, South China University of Technology, Guangzhou 510510, China)

**Abstract:** The computation of the vehicle path optimization for logistics distribution is complicated. Based on the multiple vehicles' logistics distribution model with constraints, it takes advantage of improved ant colony algorithm to solve the problem of selecting the path in the process of logistics distribution. By improving the global and local updating rules for pheromone, the convergence rate and global search capability is increased, compared with the traditional ant colony algorithm. Finally, the improved ant colony algorithm is successfully applied on path optimization of scheduling multiple vehicles in the logistics distribution.

**Key words:** ant colony algorithm; ant colony system; logistics distribution; path optimization

### 0 引言

随着经济的高速发展和城市扩张, 物流配送车辆调度优化问题是制约很多大型物流公司发展的的问题之一。对物流货运车辆的行驶路径进行优化, 不但可以实现物流科学化, 而且可以提高物流的经济效益。

物流配送车辆调度的优化目标就是让所有运送车辆在能完成客户对产品需求的前提下, 尽量使得所有车辆行驶距离最小, 继而达到总费用最低且节省能源的目标。

### 1 蚁群算法基本原理

蚁群算法是专家学者在 20 世纪 90 年代对自然界中真正的蚂蚁群体进行不断观察而总结出来的一种算法。当蚂蚁在食物源和巢穴之间爬行的时候, 它们在地面上释放一种特殊的分泌物——信息素, 蚂蚁通过遗留在地面上的信息素轨迹, 按照概率优先选择信息

浓度大的路径, 去发现觅食的最短归途。试验表明, 蚂蚁能够发现觅食最短路径直接原因就是跟踪信息素轨迹<sup>[1]</sup>。自从蚂蚁算法在著名的旅行商问题(TSP)<sup>[2,3]</sup>和工件排序<sup>[4]</sup>上取得成效以来, 蚂蚁算法也逐渐渗透到了其它问题领域。如: 车辆调度问题<sup>[5]</sup>、通讯网络中的负载均衡问题<sup>[6]</sup>、图着色问题<sup>[7]</sup>等方面都表示出了良好的性能。

目前, 人们已从蚂蚁觅食过程中总结出了一些简单的规则。蚂蚁  $k$  在  $t$  时刻在路径  $(i, j)$  上节点  $i$  的信息素浓度为  $\tau_{ij}(t)$ , 此时可以计算出它转移到下一个节点  $j \in N_i$  的概率  $P_{ij}^k(t)$ 。

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

式(1)中,

$\alpha$  为信息启发因子, 表示路径的相对重要性;

$\beta$  为期望启发因子, 表示能见度的相对重要性;

收稿日期: 2011-05-23; 修回日期: 2011-09-02

作者简介: 段爱民(1982-), 男, 硕士研究生, 研究方向为人工智能; 陈泽琳, 副教授, 硕士生导师, 研究方向为人工智能。

$\eta_{ij}(t)$  为  $t$  时刻的能见度,表示蚂蚁从节点  $i$  转移到节点  $j$  的可期望程度;

$\text{allowed}_k = \{0, 1, \dots, n-1\} - \text{tabu}_k$  是节点集合,表示蚂蚁  $k$  当前可以选择的所有节点; $\text{tabu}_k$  是用于记录蚂蚁当前已走过节点的禁忌表;

同时规定,在一定时间段完成一次循环之后,对残留的信息进行一次更新,避免残留信息素过多会引起残留信息淹没启发信息。

路径  $(i, j)$  的信息素强度  $\tau_{ij}(t)$  的更新方程为:

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

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

式(2)中,  $\rho$  ( $0 < \rho < 1$ ) 为信息素的持久系数,那么  $(1 - \rho)$  为信息素的挥发系数; $\Delta\tau_{ij}$  则代表完成一次循环后路径  $(i, j)$  上的信息素增量;

式(3)中,  $\Delta\tau_{ij}^k$  表示路径  $(i, j)$  上第  $k$  只蚂蚁在本次循环中移动的信息量。

当迭代次数到达上限时,将取蚂蚁所走路径中的最短路径作为最终路径。只要有足够多的蚂蚁,经过大量的这种地毯式搜索后,这种寻路方式就能逐渐收敛,最终达到全局最优解<sup>[8]</sup>。

蚁群算法提出以后,有不少的学者对它进行了深入的研究改进,例如德国学者 Thomas Stutzle 与 Holger Hoos<sup>[9]</sup> 改进了基础蚁群算法,提出“最大最小蚁群系统”(Max-Min Ant System, MMAS)。Gendreau, Hertz & Laporte<sup>[10]</sup> 则是使用插入法、贪婪法 (Greedy Method) 对路线进行切割,进而对旅行商问题 (TSP) 求解,产生初始解。Willard<sup>[11]</sup> 在车辆路线问题上应用了禁忌搜索法,并在重复的虚拟物流中心基础上,成功的将车辆路线问题转换成了相对简单的旅行商问题,最后再利用 2-opt 或 3-opt 方法来求解。段海滨<sup>[12]</sup> 对自适应蚁群算法进行了介绍,在引入自适应转移策略和信息素更新策略后,结合节约法,改进了基本蚁群算法易局部收敛和计算时间长等不足。

## 2 物流配送模型

物流配送车辆路径问题属于 NP-HARD 问题,在定义模型前,有如下假设:

- 配送中心拥有不同车型的车辆,即载重不同(文中不考虑容积限制);

- 每个客户点的位置和需求均已知,距离计算公式为:  $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$  (假设客户 A 坐标为  $(x_i, y_i)$ , 客户 B 坐标为  $(x_j, y_j)$ );

- 每条线路上的客户点需求量之和不超过汽车载重量;

- 每个客户点的需求必须且只能由一辆汽车来完

成。

配送系统所要达到的目标:

- 按照客户要求准时送货以提高服务质量,增加满意度。

- 在保证客户需求量不超过车辆负载,同时车辆到达客户点的行驶距离最短的前提下,使得物流配送总成本最低。

参数表:

$K$  为车辆总数; $Z$  为客户数目。

$Q_k$  为车辆  $k$  的最大载重量。

$n_i$  为客户  $i$  的产品需求重量。

$t_{ijk}$  为表征车辆  $k$  是否从节点  $i$  到节点  $j$  ( $t_{ijk}$  为 0 表示不经过,  $t_{ijk}$  为 1 时表示经过)。

$m_k$  为表征本次配送是否使用车辆  $k$  ( $m_k$  为 0 表示不使用,  $m_k$  为 1 时表示使用)。

基本约束模型:

(1) 参与配送的车辆数目不超过车辆总数。

$$\sum m_k \leq K \text{ 其中 } i = 0$$

(2) 每个客户的货物只能由一辆车送达。

$$\sum_{k=0}^{K-1} \sum_{j=0, j \neq i}^{Z-1} t_{ijk} = 1 \text{ 其中 } i = 1, 2, \dots, N$$

(3) 在车辆的配送过程中,客户的产品需求量之和不超过车辆的最大载重。

$$\sum_{k=0}^{K-1} n_i \sum_{j=0, j \neq i}^{Z-1} t_{ijk} \leq Q_k$$

## 3 改进蚁群算法的设计

### 3.1 改进算法数学模型

基于以上模型,同时兼顾车辆物流配送系统的特点,文中结合最值蚂蚁系统,并对其算法进行改进优化。现将改进算法的数学模型描述如下:

全局信息素更新规则:

为了增强蚂蚁的全局搜索能力,防止过快局部收敛,不同蚂蚁在同一路径上遗留的信息素对访问该路径的其他蚂蚁进行抑制,由此,  $t$  时刻在路径  $(i, j)$  上的信息素可按如下规则进行调整:

$$\tau_{ij}(t, n) = \rho * \tau_{ij}(t-1, n) + \Delta\tau_{ij}(t, n) - \xi_0/n - 1 * \sum_{u=1}^{n-1} \Delta\tau_{ij}(t, u) \quad (4)$$

$$\Delta\tau_{ij}(t, u) = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (5)$$

式(4)中:  $\rho$  ( $0 < \rho < 1$ ) 为信息素持久化系数;

$\Delta\tau_{ij}(t, n)$  为第  $n$  次迭代后蚂蚁留在路径  $(i, j)$  上的信息素增量;

$\xi_0 \in [0, 1]$  表示一个系数;

$\sum_{u=1}^{n-1} \Delta\tau_{ij}(t, u)$  表示完成一次迭代后属于前面  $n-1$

次迭代中蚂蚁留在路径 $(i,j)$ 上的信息素增量之和;

这样, $\xi_0/n-1*\sum_{u=1}^{n-1}\Delta\tau_{ij}(t,u)$ 表示属于前面迭代中蚂蚁释放信息素加权抑制因子。

式(5)中: $\sum_{k=1}^m\Delta\tau_{ij}^k$ 表示在第 $n$ 次迭代后所有蚂蚁留在路径 $(i,j)$ 上的信息量总和, $\Delta\tau_{ij}^k$ 的取值采用 Ant-Cycle 模型,值为:

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & \text{if the } k \text{ at go through the path of } (i,j) \text{ in the cycle} \\ 0 & \text{else} \end{cases} \quad (6)$$

式(6)中: $L_k$ 表示第 $k$ 只蚂蚁在本次循环中所走路径的总长度; $Q$ 是信息素强度,它会影响算法的收敛速度。

局部信息素更新规则:

与蚁群系统算法类似,在这里,设置局部信息素更新的作用是,根据路径爬行次数占总爬行次数的比例来进一步控制该路径上信息素的大小,从而影响蚂蚁选择该路径的概率,因此,当蚂蚁从由区域 $i$ 行驶到区域 $j$ 时,

$$\tau_{ij}(t,n) = (1 - \xi_2) * \tau_{ij}(t-1,n) + \xi_1 * \tau_0 \quad (7)$$

$$\xi_2 = \rho/\eta \quad (8)$$

式(7)中: $\eta(0 < \eta < 1)$ 是前 $n-1$ 次迭代所有蚂蚁经过路径 $(i,j)$ 的次数与前 $n-1$ 迭代参与搜寻的蚂蚁总数的比值,则 $(1-\xi)$ 反应了路径 $(i,j)$ 的重要程度。 $\xi_2(0 < \xi_2 < 1)$ 是一个参数,而 $\tau_0$ 为算法初始化时路径上信息素的浓度。

### 3.2 改进后的算法流程

以 TSP 为例,采用 Ant-Cyle 模型,改进蚁群算法的具体实现步骤如下:

(1)参数初始化。初始化各个参数值,设置迭代次数 $N_c=0$ ,设置最大迭代次数为 $N_{\max}$ ,设置蚂蚁数目 $m$ 以及初始化信息素强度 $Q$ 、最小信息素强度 $\tau_{\min}$ 、最大信息素强度 $\tau_{\max}$ 和 $\alpha$ 、 $\beta$ 的参数值。

(2)清空禁忌表,并且 $N_c = N_c + 1$ 。

(3)蚂蚁根据式(1)的路径选择概率将蚂蚁移动到下一个城市,然后修改禁忌表,把该城市增加到该蚂蚁个体的禁忌表中。

(4)若是第一次迭代中的蚂蚁则循环执行步骤(3)和步骤(4),否则蚂蚁根据式(7)进行局部信息素更新,直到每只蚂蚁都生成一条路径。

(5)计算第 $k$ 只蚂蚁所走路径的总长度 $L_k$ ,若是第一次迭代中的蚂蚁则根据式(2)进行全局信息素更新,否则蚂蚁根据式(8)进行所有路径上的全局信息素更新。

(6)若循环次数 $N_c \geq N_{\max}$ ,则循环结束,输出蚂蚁走过的路径,为最优路径,否则转步骤(2)。

## 4 实验仿真结果

通过模拟湖南邮政配送过程,来验证算法的实际可行性。文中的城市之间的距离通过调用 Google Map API 获取,测试假设为湖南省 10 个地级市的配送案例。这 10 个地级市为长沙、株洲、岳阳、衡阳、耒阳、郴州、浏阳、常德、怀化、邵阳(依次编号为 0,1,2,...,9),在这里的配送案例相对于实际情况有如下假设:

(1)配送中心到客户点所在城市之间的距离以及城市需要配送的任务量已知。

(2)不考虑天气、路况等因素对运输的影响,且认为各个城市之间都存在可互相连通的道路。

(3)设定所有的客户需求形式都相同,运输方式也都相同,采用陆运。

表 1 为各个城市的任务表,任务总量为 2347。

表 1 各个城市的任务表

| 城市编号 | 城市名称 | 任务表 | 城市编号 | 城市名称 | 任务表 |
|------|------|-----|------|------|-----|
| 0    | 长沙   | 0   | 5    | 郴州   | 214 |
| 1    | 株洲   | 125 | 6    | 浏阳   | 191 |
| 2    | 岳阳   | 44  | 7    | 常德   | 470 |
| 3    | 衡阳   | 56  | 8    | 怀化   | 273 |
| 4    | 耒阳   | 384 | 9    | 邵阳   | 590 |

表 2 为通过 Google Maps API 获取的城市之间距离矩阵表(单位为公里)。

表 2 城市之间距离矩阵表

|   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     |
|---|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0     | 71.3  | 156.2 | 194.7 | 232.0 | 321.5 | 74.4  | 166.3 | 428.2 | 226.8 |
| 1 | 71.3  | 0     | 200.2 | 141.2 | 178.5 | 268.0 | 73.7  | 232.8 | 418.2 | 216.8 |
| 2 | 156.2 | 200.2 | 0     | 322.8 | 360.1 | 449.6 | 202.7 | 320.4 | 571.5 | 370.0 |
| 3 | 194.7 | 141.2 | 322.8 | 0     | 90.6  | 180.1 | 241.5 | 355.5 | 331.1 | 129.6 |
| 4 | 232.0 | 178.5 | 360.1 | 90.6  | 0     | 100.4 | 279.5 | 393.5 | 422.2 | 220.7 |
| 5 | 321.5 | 268.0 | 449.6 | 180.1 | 100.4 | 0     | 366.3 | 480.3 | 508.9 | 307.5 |
| 6 | 74.4  | 73.7  | 202.7 | 241.5 | 279.5 | 366.3 | 0     | 239.3 | 490.5 | 289.0 |
| 7 | 166.3 | 232.8 | 320.4 | 355.5 | 393.5 | 480.3 | 239.3 | 0     | 283.6 | 383.0 |
| 8 | 428.2 | 418.2 | 571.5 | 331.1 | 422.2 | 508.9 | 490.5 | 283.6 | 0     | 210.8 |
| 9 | 226.8 | 216.8 | 370.0 | 129.6 | 220.7 | 307.5 | 289.0 | 383.0 | 210.8 | 0     |

现假设前提运输车辆为 3 台:A 车辆的最大载重量为 800kg,最大行驶距离是 1000km;B 车辆的最大载重量为 1200kg,最大行驶距离是 800km;C 车辆的最大载重为 800kg,最大行驶距离为 800km。

系统的各个参数为 $Q=100$ , $\alpha=6$ , $\beta=4$ , $\rho=0.7$ ,蚂蚁数量为 15,迭代次数为 320 次,全局更新规则参数 $\zeta_0=0.3$ ,局部更新规则参数 $\zeta_1=0.4$ ,最小信息素强度 $\tau_{\min}=0.1$ ,最大信息素强度 $N_{\max}=10$ ,用改进的蚁群算法对其进行模拟计算结果为:

经过计算,较优路径长度为 2328.0km,较优路径

中各车辆的配送情况如下:

车辆 A 的行驶距离为 991.5km,分配货物重量为 698kg,配送城市为长沙-耒阳-郴州-衡阳-岳阳-长沙。

车辆 B 的行驶距离为 1003.9km,分配货物重量为 1179kg,配送城市为长沙-浏阳-株洲-邵阳-怀化-长沙。

车辆 C 的行驶距离为 332.6km,分配货物重量为 470kg,配送城市为长沙-常德-长沙。

## 5 结束语

文中首先对蚁群算法进行了详细介绍。然后在最优蚂蚁系统的基础上,结合物流配送的特点,对蚂蚁所走路径的信息素进行局部更新。同时改进其全局更新策略,提高了算法的收敛速度和搜索能力,也提高了算法的效率。并在最后将改进算法和实际物流配送系统相结合,有效地解决了物流配送的路径优化问题,证明了改进算法在车辆路径优化问题中的可行性。

### 参考文献:

- [1] 胡 娟,王常青,韩 伟. 蚁群算法及其实现方法研究[J]. 计算机仿真,2004(7):10-14.
- [2] Dorigo M, Bonabeau E, Theraulaz G. Ant algorithm and stigmergy[J]. Future Generation Computer Systems, 2000, 16

(9):851-871.

- [3] Dorigo M, Gambardella L M. Ant colonies for the traveling salesman problem[J]. BioSystems, 1997, 43:73-81.
- [4] 龚雨兵,李泉水. 蚂蚁算法在工件排序问题中的应用[J]. 桂林电子工业学院学报,2004(2):68-71.
- [5] 崔雪丽,马 良,范炳全. 车辆路径问题的蚂蚁搜索算法[J]. 系统工程学报,2004(4):418-422.
- [6] Schoonderwoerd R, Holland O, Bruten J. Ant-like agents for load balancing in telecommunications networks[C]//Proc of Agents 97. Marina del Rey, CA:ACM Press, 1997:209-216.
- [7] 张 丽,马 良,石丽娜. 图着色问题的蚂蚁算法研究[J]. 上海工程技术大学学报,2009(4):328-332.
- [8] 赵 虎,李 睿. 蚂蚁算法在车间作业调度问题中的应用[J]. 计算机工程与应用,2003(22):6-8.
- [9] Stutzle T, Hoos H. The MAX-MIN ant system and local search for the traveling salesman problem[C]//Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC 97). Piscataway, NJ:IEEE Press, 1997.
- [10] Gendreau M, Hertz A, Laporte G. A Tabu Search Heuristic for the Vehicle Routing Problem[J]. Management Science, 1994, 40(10):1276-1290.
- [11] Willard J A G. Vehicle routing using P-optimal tabu search [D]. London: The Management School, Imperial College, 1989.
- [12] 段海滨,王道波,朱家强,等. 蚁群算法理论及应用研究的进展[J]. 控制与决策,2004,19(12):1321-1326.

(上接第 177 页)

可见,优化后的模型视觉效果大为改善,更直观清晰。

## 3 结束语

文中介绍了 CT 断层图像的分割与三维重建过程用到的原理及方法,实验结果也满足了双下肢三维建模的要求。三维重建弥补了 CT 二维成像的不足,能直观、立体、清楚地显示出骨骼立体形态以及空间关系。医务人员可根据重建得到的三维图像进行术前诊断、手术设计等操作。CT 三维重建为数字骨科学奠定了基础,随着该技术的不断成熟与发展,其在医学领域的应用也将更为广泛。

### 参考文献:

- [1] 裴国献. 数字骨科学[M]. 北京:人民卫生出版社,2009.
- [2] 陈家新. 医学图像处理及三维重建技术研究[M]. 北京:科学出版社,2010.
- [3] 卫阿盈,杨 磊. 图像分割在医学图像处理中的应用[J]. 医学信息,2005,18(12):1629-1631.
- [4] Tian Jie, Xue Jian, Dai Yakang. A novel software platform for medical image processing and analyzing[J]. IEEE Transac-

tions on Information Technology in Biomedicine, 2008, 12 (6):800-812.

- [5] 王志兵. 基于模糊聚类的图像分割技术研究[D]. 重庆:西南大学,2009.
- [6] 王东明. 模糊聚类在 CT 图像骨骼分割中的应用[D]. 西安:西安电子科技大学,2006.
- [7] Lorensen W E, Cline H E. Marching cubes: A high resolution 3D surface construction algorithm[J]. ACM Computer Graphics, 1987, 21:38-44.
- [8] Cline H, Lorensen W. Two algorithms for the 3D reconstruction of tomograms[J]. Medical Physics, 1988, 15(3):320-327.
- [9] 姚均营. 医学影像三维立体可视化系统研究[D]. 南京:南京航空航天大学,2007.
- [10] 洪 锋,梅 炯,李明禄. 医学图像三维重建技术综述[J]. 中国图象图形学报,2003,8(z1):785-787.
- [11] 唐占红,於时才. 面绘制三维重建原理及其改进算法研究[J]. 计算机工程与设计,2009,30(9):2225-2228.
- [12] 郑 莹. 下颌骨系统模型重建及受力分析的研究[D]. 上海:同济大学,2010.
- [13] 谢小棉,李树祥,江贵平,等. 基于 MC 的医学三维等值面的平滑与归并[J]. 中国图象图形学报,2001,6(8):806-809.
- [14] 李 华,蒙培生,王 乘. 医学图像重建 MC 算法三角片的合并与实现[J]. 计算机应用,2003(6):104-106.