# 基于改进蚁群算法的车辆路径优化问题研究*

陈迎欣
(哈尔滨工程大学 经济管理学院, 哈尔滨 150001)

**摘　要**: 物流活动中需要找出各个配货节点之间的最短路径,用以指导物流车辆调度,进而节约物流成本。提出解决车辆路径优化问题的方法,针对蚁群算法的缺点,分别对信息素更新策略、启发因子进行改进,并引入搜索热区机制,有效解决了蚁群算法的缺陷。最后,以哈尔滨市局部地图为原型,应用 MATLAB 软件对改进蚁群算法求解车辆路径优化问题的性能进行仿真,并与基本蚁群算法对比分析,验证了改进蚁群算法的有效性和可行性。

**关键词**: 蚁群算法; 车辆路径优化; 信息素; 物流
**中图分类号**: TP301　　**文献标志码**: A　　**文章编号**: 1001-3695(2012)06-2031-04
doi:10.3969/j.issn.1001-3695.2012.06.007

## Study on VRP based on improved ant colony optimization

CHEN Ying-xin
(*School of Economics & Management*, *Harbin Engineering University*, *Harbin* 150001, *China*)

**Abstract**: Logistics activities need to find different distribution node of the shortest path, to instruct the logistics vehicle scheduling, and then save the logistics cost. This paper proposed the solution of vehicle routing optimization problem. In order to conquer the defects and improve the basic ant colony optimization, it improved pheromones updating strategy, stimulating factor and the introduction of search hotspots, solved the defects of ant colony optimization effectively. With the help of Harbin city map as the prototype and the MATLAB software, it carried out simulation to check the improved ant colony optimization. The result verifies the feasibility and effectiveness of the improved ant colony optimization.

**Key words**: ant colony optimization; vehicle routing problem(VRP); pheromone; logistics

## 0　引言

车辆路径问题(vehicle routing problem, VRP)是物流系统优化环节中关键的一环。对运输车辆进行优化调度,通过选择车辆的最佳运输路径,合理安排车辆调度顺序,可以有效减少车辆的空驶率和行驶距离。VRP 已经应用到牛奶配送、报纸和快件投递、垃圾车的线路优化及连锁商店的送货线路优化等众多社会领域[1]。

由于该问题属于求解较为困难的组合优化 NP-hard 问题而且具有强有力的应用背景,长期以来吸引着大量学者对其不断地进行研究探索,研究人员从最初的精确算法开始,逐步将目光投向更广阔的启发式算法,如禁忌搜索算法、遗传算法、蚁群算法等。钟石泉等人[2]对软、硬时间窗约束的 VRP 进行了研究,并用禁忌搜索算法分别进行求解;Chao[3]利用禁忌搜索算法求解了多车型的 VRP;Chen 等人[4]利用遗传算法来处理钢铁领域采用半挂车运输车辆进行物流配送中的车辆时间表制定及路径选择问题;唐炉亮等人[5]通过对蚁群算法的改进解决了 GPS 数据的公众出行路径优化问题。这些算法在 VRP 的求解上取得了可喜的成果,但也存在着不少问题,如禁忌搜索算法存在对初始解有较高的依赖性的缺点;遗传算法有局部搜索能力不强、易陷入早熟、总体上可行解的质量不是很高的缺点。

蚁群算法自从被提出就受到了广泛的关注,该算法在解决路径优化、任务分配等问题时表现出了良好的性能,但是蚁群算法也存在一定的缺陷,如在解决大规模问题时,收敛速度慢、运算时间长、容易陷入局部最优解等[6]。本文将基本蚁群算法进行改进,克服了以上缺点,使算法性能更好,得到了运输效率更高、运算结果更精确的车辆路径优化算法。

## 1　VRP 的数学模型

VRP 的数学模型表述如下:假设客户点共有 $k$ 个(1,2,…,$k$),将第 $i$ 个客户点的货物需求量设为 $g_i$($i=1,2,\cdots,k$),配送中心派出车辆承担运输任务,其载重量为 $q$。设 $g_i < q$,求满足货运需求的最短路线。

为了方便地安排路线,可以事先估算需要使用的车辆数目。在现实情况中,车辆数目的需求量取决于货物装车(或卸车)的复杂程度,装卸越复杂,约束条件就越多,车辆的实际载货量就越小。用户自主选择的车辆数目可以由下列公式求得:

$$m = [\sum g_i / \alpha q] + 1 \tag{1}$$

其中:$m$ 表示所需的车辆数;[ ]表示取整;$\alpha$ 是参数,且 $0 < \alpha < 1$。约束条件越多的情况下,货物装车、卸车越复杂,$\alpha$ 值就越小,一般取 $\alpha = 0.85$。

用 $c_{ij}$ 表示点 $i$ 到点 $j$ 的运输成本(如时间、路程、花费等),

**收稿日期**: 2011-11-07; **修回日期**: 2011-12-12　　**基金项目**: 黑龙江省哲学社会科学基金资助项目(08E004);中央高校基本业务费专项基金资助项目(HEUCF120910)

**作者简介**: 陈迎欣(1975-),女,河北景县人,副教授,博士,主要研究方向为建模仿真、危机管理(hexieshehui2011@163.com).

配送中心的编号为 0,各客户点的编号为 $i(i=1,2,\cdots,k)$,现定义如下变量:

$$x_{ijs}=\begin{cases}1 & \text{车辆 } s \text{ 从 } i \text{ 点行驶到 } j \text{ 点}\\ 0 & \text{otherwise}\end{cases} \tag{2}$$

$$y_{ijs}=\begin{cases}1 & \text{客户 } i \text{ 由车辆 } s \text{ 负责完成}\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

可以建立下列目标函数的数学模型:

$$\min Z=\sum_{i=0}^{k}\sum_{j=0}^{k}\sum_{s=1}^{m}c_{ij}x_{ijs} \tag{4}$$

限制条件如下:

$$\sum_{i=1}^{k}g_i y_{is}\leqslant q \quad s=1,2,\cdots,m \tag{5}$$

$$\sum_{i=1}^{m}y_{is}=\begin{cases}1 & i=1,2,\cdots,k\\ m & i=0\end{cases} \tag{6}$$

$$\sum_{i=0}^{k}x_{ijs}=y_{js} \quad j=1,2,\cdots,k;s=1,2,\cdots,m \tag{7}$$

$$\sum_{j=0}^{k}x_{ijs}=y_{js} \quad i=1,2,\cdots,k;s=1,2,\cdots,m \tag{8}$$

$$x_{ijs}=0/1 \quad i,j=0,1,\cdots,k;s=1,2,\cdots,m \tag{9}$$

$$y_{is}=0/1 \quad i=1,2,\cdots,k;s=1,2,\cdots,m \tag{10}$$

式(5)约束了车辆的容量;式(6)是为了保证每个客户点的运输任务仅由一辆车来完成,而所有运输任务则由 $m$ 辆车协同完成;式(7)和(8)保证了到达和离开某一客户点的货车有且仅有一辆。

## 2 改进的蚁群算法

自然界中的蚂蚁个体虽然简单,但是蚂蚁群体通过合作能够完成复杂的任务,经过研究发现,蚂蚁个体可以释放一种被称为信息素的物质,并通过感知该物质的浓度进行交流。蚂蚁的初始运动是随机的,但是会受到信息素浓度的影响,根据感知到的信息素的浓度来决定下一步的运动方向[7]。在觅食过程中,蚁群总是能够找到蚁穴到食物源之间的最短路径。

本文在考虑改进蚁群算法时,基于前人的研究成果提出了以下几项改进措施,包括信息素更新策略的改进、启发因子的改进、搜索热区的引入等。

### 2.1 信息素更新策略的改进

在基本蚁群算法研究中,有两种信息素更新策略:实时更新和全局更新。前者是指蚂蚁从一个节点到达另一个节点就立刻更新蚂蚁所走路径的信息素,后者是指蚂蚁遍历过所有节点后再更新整条路径上的信息素。两种方式相比,全局的信息素更新策略可以提高收敛速度。目前研究表明全局性的更新效果更好,但是也存在一定的缺陷,这种更新方法会导致全局过早收敛,使蚂蚁更快地集中于一条路径上,无法发现更好的解,即陷入局部最优。

在信息素的更新中,通常会对一只蚂蚁的信息素进行更新,这只蚂蚁是找到最优路径的蚂蚁。针对这只蚂蚁的信息素更新也有两种方式,一种是找出本次循环中表现最好的蚂蚁,这种方式收敛速度慢,但是不会迅速收敛到某路径上,蚁群依旧会继续探索新的路径,这种方法更易于发现更好的解;另一种是找出运算以来表现最好的蚂蚁,这种更新方式能够提高收敛速度,使蚁群迅速收敛到较优的解中,但是也会阻碍蚁群发现更优解,容易造成蚁群被困在相对较差的路径上。

本文针对这两种方式各自的优缺点,提出了一种新的混合更新策略,即在搜索的前十次循环中,采用本次迭代最优进行更新,即利用本次循环中表现最好的蚂蚁来进行信息素更新,这种更新方式有利于发现更多较优的解,以避免蚁群陷入较差解中;十次固定循环以后,再采用全局最优解进行更新,即利用运算以来表现最好的蚂蚁进行信息素的更新。采用混合的信息素更新策略后,蚁群算法首先会收敛到较优的解集中,找到更多的可行解,而且还能保持继续搜索较优解的能力,不会限制对最优解的进一步搜索,并且保持了较快的收敛速度,克服了单一采用全局最优解更新时容易陷入局部最优的缺陷。

改进后的信息素更新规则如下:

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{\text{best}} \tag{11}$$

其中:$\Delta\tau_{\text{best}}=1/f(s_{\text{best}})$;$f(s_{\text{best}})$ 表示本次迭代中的最优解 $s_{\text{ib}}$ 或是全局最优解 $s_{\text{ob}}$。

另外为了防止搜索停滞(即由于某条路径上信息素浓度过高导致的所有蚂蚁都集中于一条路径上,而不再进行搜索),需要将路径上的信息素限定在某个范围内。对所有的信息素 $\tau_{ij}(t)$ 都有 $\tau_{ij}(t)\in[\tau_{\min},\tau_{\max}]$,若 $\tau_{ij}(t)\geqslant\tau_{\max}$,则将 $\tau_{ij}(t)$ 设置为 $\tau_{ij}(t)=\tau_{\max}$,若 $\tau_{ij}(t)\leqslant\tau_{\min}$,则将 $\tau_{ij}(t)$ 设置为 $\tau_{ij}(t)=\tau_{\min}$。若在迭代中出现了某个解,其信息素的量为 $\tau_{\max}$,而其他路径上信息素的浓度为 $\tau_{\min}$,则算法出现收敛。

信息素的改进可以按照以下策略动态地进行:

a)在初始状态下,信息素尚未得到更新时,采用式(6)(7)来确定初始更新范围,$L(s_{\text{best}})$ 表示全局最优解或本次迭代最优解,$1/L(s_{\text{best}})$ 表示每次迭代路径上新增的信息素最高值。

$$\tau_{\max}(t)=\frac{1}{2(1-\rho)}\times\frac{1}{L(s_{\text{best}})} \tag{12}$$

$$\tau_{\min}(t)=\frac{\tau_{\max}(t)}{20} \tag{13}$$

b)一旦信息素更新以后,则开始采用式(14)来确定 $\tau_{\max}(t)$,$\tau_{\min}(t)$ 仍采用式(13)来确定。

$$\tau_{\max}(t)=\frac{1}{2(1-\rho)}\times\frac{1}{L(s_{\text{best}})}+\frac{1}{L(s_{\text{best}})} \tag{14}$$

### 2.2 启发因子的改进

蚁群算法中,启发函数设为 $\eta_{ij}(t)$,通常取为 $1/d_{ij}$,其中 $d_{ij}$ 表示城市 $i$ 与城市 $j$ 之间的距离。启发函数的作用是帮助蚂蚁在下一步选择路径时,能够以更大的概率选择较短路径,两个节点之间的路径越长,$\eta_{ij}(t)$ 的值越小,则蚂蚁会以更小的概率选择该路径。但是 $\eta_{ij}(t)=1/d_{ij}$ 用以表示城市 $i$ 与城市 $j$ 之间对应的距离关系,这种更新方式存在一定缺陷。由于没能从全局着眼,只能表达出两节点之间的关系,而不能体现出现在节点与出发点之间的关系,因此会降低全局最优解的搜索效率。本文将启发函数改为

$$\eta_{ij}(t)=1/d_{0j} \tag{15}$$

其中:的 $d_{0j}$ 表示出发原点到当前节点 $j$ 之间的距离,这一改进不仅能够从全局着眼,也能为蚁群搜索全局最优解指明方向,找到从原点到各节点之间的关系,从而提高搜索效率与精确程度。

### 2.3 搜索热区的引入

所谓的搜索热区是指以出发原点和目标点为对角而画出的矩形区域,最短路(最优解)落在热区内的概率要远远大于落在热区以外的概率。因此可以先行判断当前路径是否落在热区内,根据是否在搜索热区内部来确定下一步选择该路径的权重。引入搜索热区后,蚁群转移的概率变为

$$p_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)\phi_{ij}(t)}{\sum_{r\in\text{allowed}_k}\tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}(t)} & j\in\text{allowed}_k\\ 0 & \text{otherwise}\end{cases} \tag{16}$$

其中:$\phi_{ij}(t)$的值会根据在 $t$ 时刻 $i,j$ 之间的路径是否落在搜索热区内而取不同的值。

$$\phi_{ij}(t)=\begin{cases}x & (x>1)\text{路径}(i,j)\text{在搜索热区内}\\ y & (y\leqslant 1)\text{路径}(i,j)\text{不在搜索热区内}\end{cases} \tag{17}$$

引入搜索热区后可以提高蚁群搜索的效率,并且防止蚁群陷入局部最优解。

## 3　改进后的蚁群算法实现步骤

针对本文中对蚁群算法的以上改进,整理改进后的蚁群算法的实现步骤如下:

a)参数初始化。设 $N_c=0$($N_c$ 为迭代次数);每条路径上的信息素浓度 $\tau_{ij}(0)=c$($c$ 为常数),并且 $\Delta\tau_{ij}=0$;随机将 $m$ 个蚂蚁放置到初始点上。

b)设置循环次数,即 $N_c=N_c+1$。

c)设置禁忌表,将蚂蚁禁忌表的索引号设为 $k=1$。

d)设置蚂蚁数目,蚂蚁数目的变化为 $k=k+1$。

e)判断路径是否在搜索热区内。按照式(15)~(17)计算出每只蚂蚁应当移动到下一条路径的概率,并实现蚂蚁的移动。

f)当蚂蚁从 $i$ 点出发到达 $j$ 点时,对其所经过的路径上的信息素进行更新并修改禁忌表,将禁忌表指针按照当前状况进行修改,即将新城市 $j$ 置于禁忌表 $\text{tabu}_k$ 中。

g)执行步骤 b)~f),直到每只蚂蚁都找到了一条包含所有节点的可行路径。

h)在新生成的所有可行解中找到一条最短路径,这条最短路径就是本次迭代中的最优路径。

i)判断循环次数,若循环次数小于十次,则对蚂蚁找到的最优路径按照本次迭代最优值进行信息素更新;若循环次数超过十次,则按照全局信息素进行更新。

j)对所有蚂蚁经过的每一条路径,按式(11)~(14)进行一次全局信息素的更新。

k)循环执行 b)~j),直到连续多次的迭代中没有出现更优解或 $N_c$ 的值已经达到最大迭代次数,停止运算,输出当前值作为最优解。

## 4　实例验证

为验证改进蚁群算法的有效性,本文以哈尔滨市地图为例,在 MATLAB 平台上进行实例仿真实验。

从哈尔滨市地图上截取一段平面图,将交叉路口作为节点,将每个节点位置进行编号,如图 1 所示。本张地图共有 76 个节点,利用蚁群算法和改进的蚁群算法来遍历哈尔滨地图上的 76 个节点,观察仿真结果。

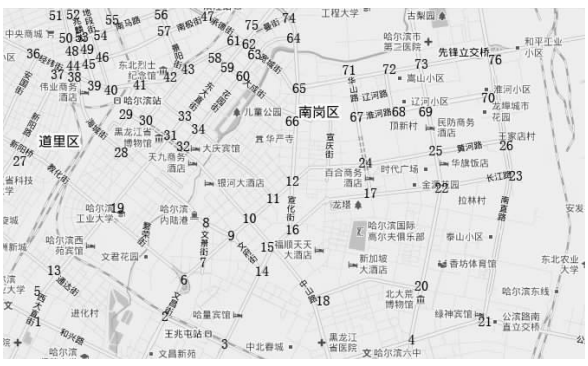


**图1　哈尔滨市部分交通平面图**

从网络地图上取到每个节点的经纬度,节点的经纬度如表 1 所示。

表 1　节点经纬度

| 节点 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 纬度 | 45.732 8 | 45.732 7 | 45.728 9 | 45.729 4 |
| 经度 | 126.615 7 | 126.642 2 | 126.653 9 | 126.690 8 |
| 节点 | 5 | 6 | 7 | 8 |
| 纬度 | 45.736 2 | 45.737 8 | 45.740 4 | 45.745 9 |
| 经度 | 126.618 0 | 126.645 9 | 126.649 7 | 126.649 9 |
| 节点 | 9 | 10 | 11 | 12 |
| 纬度 | 45.743 9 | 45.746 1 | 45.749 0 | 45.751 3 |
| 经度 | 126.655 5 | 126.658 9 | 126.663 8 | 126.667 4 |
| 节点 | 13 | 14 | 15 | 16 |
| 纬度 | 45.732 4 | 45.739 4 | 45.742 5 | 45.744 6 |
| 经度 | 126.620 4 | 126.661 2 | 126.663 4 | 126.666 9 |
| 节点 | 17 | 18 | 19 | 20 |
| 纬度 | 45.749 6 | 45.735 1 | 45.747 0 | 45.736 6 |
| 经度 | 126.682 9 | 126.671 3 | 126.630 6 | 126.692 7 |
| 节点 | 21 | 22 | 23 | 24 |
| 纬度 | 45.732 7 | 45.750 6 | 45.752 4 | 45.753 8 |
| 经度 | 126.704 9 | 126.696 7 | 126.710 1 | 126.681 8 |
| 节点 | 25 | 26 | 27 | 28 |
| 纬度 | 45.755 5 | 45.757 1 | 45.756 2 | 45.755 6 |
| 经度 | 126.695 7 | 126.709 1 | 126.613 3 | 126.633 7 |
| 节点 | 29 | 30 | 31 | 32 |
| 纬度 | 45.760 8 | 45.759 5 | 45.757 4 | 45.755 9 |
| 经度 | 126.634 9 | 126.638 0 | 126.642 5 | 126.645 2 |
| 节点 | 33 | 34 | 35 | 36 |
| 纬度 | 45.760 2 | 45.758 6 | 45.762 5 | 45.768 7 |
| 经度 | 126.645 7 | 126.648 7 | 126.641 6 | 126.616 9 |
| 节点 | 37 | 38 | 39 | 40 |
| 纬度 | 45.766 9 | 45.766 0 | 45.765 0 | 45.764 5 |
| 经度 | 126.622 2 | 126.624 6 | 126.628 0 | 126.630 8 |
| 节点 | 41 | 42 | 43 | 44 |
| 纬度 | 45.765 0 | 45.765 6 | 45.766 9 | 45.767 3 |
| 经度 | 126.637 3 | 126.642 8 | 126.646 4 | 126.624 3 |
| 节点 | 45 | 46 | 47 | 48 |
| 纬度 | 45.767 7 | 45.768 3 | 45.774 0 | 45.759 2 |
| 经度 | 126.626 9 | 126.629 3 | 126.650 4 | 126.623 7 |
| 节点 | 49 | 50 | 51 | 52 |
| 纬度 | 45.769 5 | 45.771 0 | 45.775 2 | 45.775 7 |
| 经度 | 126.626 1 | 126.623 2 | 126.621 9 | 126.623 7 |
| 节点 | 53 | 54 | 55 | 56 |
| 纬度 | 45.771 3 | 45.771 7 | 45.772 6 | 45.774 9 |
| 经度 | 126.625 4 | 126.627 8 | 126.632 0 | 126.641 2 |
| 节点 | 57 | 58 | 59 | 60 |
| 纬度 | 45.772 1 | 45.769 0 | 45.767 5 | 45.765 9 |
| 经度 | 126.642 8 | 126.651 5 | 126.654 3 | 126.657 0 |
| 节点 | 61 | 62 | 63 | 64 |
| 纬度 | 45.771 0 | 45.770 6 | 45.769 2 | 45.771 2 |
| 经度 | 126.657 2 | 126.657 9 | 126.660 6 | 126.667 9 |
| 节点 | 65 | 66 | 67 | 68 |
| 纬度 | 45.764 3 | 45.760 5 | 45.760 2 | 45.761 1 |
| 经度 | 126.668 4 | 126.668 2 | 126.680 2 | 126.687 2 |
| 节点 | 69 | 70 | 71 | 72 |
| 纬度 | 45.762 0 | 45.763 7 | 45.765 7 | 45.766 5 |
| 经度 | 126.694 1 | 126.707 8 | 126.678 9 | 126.685 8 |
| 节点 | 73 | 74 | 75 | 76 |
| 纬度 | 45.767 4 | 45.774 2 | 45.772 0 | 45.769 1 |
| 经度 | 126.692 7 | 126.666 4 | 126.659 6 | 126.706 3 |

按照基本蚁群算法进行计算,设置参数为:蚂蚁数量 =20,初始信息素 $\tau_{ij}(0)=20$,信息素挥发因子 $\rho=0.5$,信息素对选

择概率上的影响参数 $\alpha = 0.5$，路径长度对选择概率的影响参数 $\beta = 5$，蚂蚁每次更新的信息素数量为 1，最高迭代次数为 100 次。利用基本蚁群算法仿真的结果如图 2 和 3 所示。

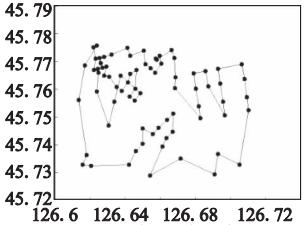


图2 用基本蚁群算法的仿真路径

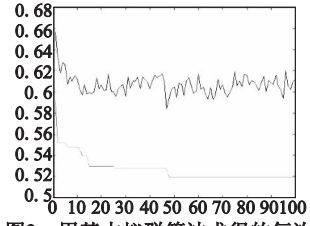


图3 用基本蚁群算法求得的每次循环平均路径长度和最短路径长度

结果显示，仿真求得的最短路径为

columns 1 through 38

62→61→75→63→60→59→58→47→57→56→55→54→53→49→45→46→39→40→29→30→35→42→43→33→31→32→34→41→28→19→48→38→44→37→50→51→52→36

columns 39 through 76

27→5→1→13→2→6→7→8→9→10→11→12→16→15→14→3→18→4→20→21→23→26→70→76→73→69→25→22→68→72→71→67→24→17→66→65→64→74

shortest_length = 0.5192

同样的参数设置，利用改进的蚁群算法对哈尔滨市地图上的交通节点进行遍历仿真，得到的仿真结果如图 4 和 5 所示。

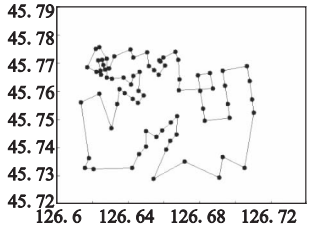


图4 用改进蚁群算法的仿真路径

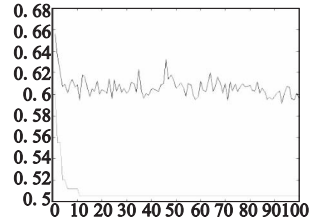


图5 用改进蚁群算法求得的每次循环平均路径长度和最短路径长度

结果显示，仿真求得的最短路径为

columns 1 through 38

4→20→21→23→26→70→76→73→69→25→22→17→24→67→71→72→68→66→65→64→74→75→62→61→63→60→59→58→47→57→56→55→46→45→49→53→50→44

columns 39 through 76

38→37→36→51→52→54→39→40→41→35→42→43→33→34→32→31→30→29→28→19→48→27→5→1→13→2→6→7→8→9→10→11→12→16→15→14→ 3→18

shortest_length = 0.5057。

从实例仿真中可以看出，改进的蚁群算法相对于基本蚁群算法迭代次数更少，能够以更快的速度收敛到全局最优解；求解方面，改进的蚁群算法找到的最短路径为 0.5057，基本蚁群算法找到的最优解为 0.5192，其解的质量也有所提高。说明改进的蚁群算法跟一般的蚁群算法相比不仅操作简单，而且收敛效果好、收敛速度快、全局收敛率高。

## 5 结束语

本文对基本蚁群算法进行改进，包括参数的改进、信息素更新策略的改进、信息素挥发因子的改进等，设计出新的改进蚁群算法。利用改进的蚁群算法求解 VRP，并利用 MATLAB 软件进行仿真，以哈尔滨市地图为模型验证了改进蚁群算法的有效性。仿真结果表明，改进的蚁群算法是可行、有效的。

**参考文献：**

[1] 宋留勇，王锐，周永旺，等. 动态城市交通网络优化模型研究及算法设计[J]. 测绘科学，2011，36(1)：134-136.

[2] 钟石泉，贺国光. 有时间窗约束车辆调度优化的一种禁忌算法[J]. 系统工程理论方法应用，2005，14(6)：523-527.

[3] CHAO Yi-ming. A tabu search method for the truck and trailer routing problem[J]. Computer and Operations Research，2002，29(1)：33-51.

[4] CHEN Yao-rong，LIANG Bo，ZHOU Mei-hua. Optimization for vehicle scheduling in iron and steel works based on semi-trailer swap transport[J]. 中南大学学报：英文版，2010，17(4)：873-879，

[5] 唐炉亮，常晓猛，李清泉，等. 基于蚁群优化算法与出租车 GPS 数据的公众出行路径优化[J]. 中国公路学报，2011，24(2)：89-95.

[6] 杨中秋，张延华. 改进蚁群算法在交通系统最短路径问题的研究[J]. 科学计算与信息处理，2009，32(8)：76-78.

[7] DORIGO M，GAMBARDELLA L M. Ant colony system：a cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computations，1997，1(1)：67-68.