

# 改进蚁群算法在快递配送路径中的应用

陆缘缘, 高华成, 崔 衍

(南京邮电大学 物联网学院, 江苏 南京 210023)

**摘 要:** 蚁群算法作为一种启发式算法,最初是用来解决 TSP(traveling salesman problem)问题,基本思想来源于自然界蚂蚁觅食的最短路径原理,目前在快递配送中已被广泛使用。但其前期信息素匮乏导致搜索速度较慢和计算最优路径时迭代次数过多的问题尚未较好地解决。针对这一问题,提出了一种改进蚁群算法。首先,通过借鉴遗传算法启发函数的思想,对蚁群算法信息素初始值引入改进启发函数进行动态更新,针对不同情况进行寻优节点选择,解决前期搜索速度较慢的问题。其次,对信息素更新公式进行改进,引入自适应计算公式,根据每个节点对应的其他节点情况进行信息素更新,使蚁群更有目的的进行路径选择,加快蚁群算法搜索速度并减少迭代次数。最后,在蚁群算法迭代中引入变异和交叉操作,避免陷入局部最优解。通过仿真实验可以看出,与传统蚁群算法相比,改进的蚁群算法具有更好的寻优能力,且在收敛速度和迭代次数计算值中有明显提升。

**关键词:** 蚁群算法;遗传算法;启发函数;最优解;信息素

**中图分类号:** TP391

**文献标识码:** A

**文章编号:** 1673-629X(2021)11-0015-06

doi:10.3969/j.issn.1673-629X.2021.11.003

## Application of Improved Ant Colony Algorithm in Express Delivery Route

LU Yuan-yuan, GAO Hua-cheng, CUI Yan

(School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

**Abstract:** As a heuristic algorithm, the ant colony algorithm was originally used to solve the TSP (traveling salesman problem). The basic idea is derived from the shortest path principle of ants foraging in nature. It has been widely used in express delivery. However, the lack of pheromone in the early stage leads to slow search speed and too many iterations when calculating the optimal path, which have not been solved. Aiming at this problem, an improved ant colony algorithm is proposed. Firstly, by drawing on the idea of heuristic function of genetic algorithm, the initial value of ant colony algorithm pheromone is introduced into an improved heuristic function for dynamically updating, and the optimal node selection is carried out according to different situations to solve the problem of slow search speed in the early stage. Secondly, the pheromone update formula is improved, an adaptive calculation formula is introduced, and the pheromone is updated according to the situation of other nodes corresponding to each node, so that the ant colony can make more purposeful path selection, speed up the ant colony algorithm search and reduce the number of iterations. Finally, mutation and crossover operations are introduced in the ant colony algorithm iteration to avoid falling into the local optimal solution. Through the simulation experiment, it can be seen that compared with the traditional ant colony algorithm, the improved ant colony algorithm has better optimization, and the convergence speed and the calculated value of the number of iterations are significantly improved.

**Key words:** ant colony algorithm; genetic algorithm; heuristic function; optimal solution; pheromone

## 0 引 言

随着快递业的飞速发展,快递单量也在飞速增加,如何能够准时地进行快递配送已经成为现在快递行业的一个重要评价标准。在史晓原<sup>[1]</sup>的研究中发现,近年来快递投诉有效率高达 95%,且其最突出的问题就是快递晚点问题。如何提高末端快递配送速度成为重

中之重。对于这一问题,近年来国内外学者提出了多种优化算法对快递配送路径进行优化,包括蚁群算法、遗传算法、粒子群算法等。杨从平<sup>[2]</sup>将信息素释放量与各个节点的蚂蚁数量联系在一起,采用动态的释放函数进行各个节点的信息素释放。胡春阳等<sup>[3]</sup>对两点之间的搜索进行优化,对蚁群算法进行改进,提出双向

收稿日期:2020-12-29

修回日期:2021-04-29

基金项目:江苏省研究生科研与实践创新计划项目(KYCX19\_0985);中国博士后科学基金面上项目(2020M671554)

作者简介:陆缘缘(1993-),女,硕士研究生,研究方向为物联网技术在通信中的应用。

搜索机制并引入惩戒因子,可以提高蚁群的搜索能力。李桃迎等<sup>[4]</sup>对外卖配送问题改进遗传算法,引入聚类算法根据配送点的属性进行聚类分析,并在其后的遗传算法计算中对编码形式进行改进,加速优化速度。赵静等<sup>[5]</sup>对启发函数和信息素挥发因子进行改进以提高蚁群算法的搜索能力,采用连接目标方式改进启发函数使搜索更具有目的性,并使信息素挥发因子自适应变化,保证在找到最优解时可以快速收敛。Jiao Z 等<sup>[6]</sup>对蚁群算法进行改进,提出多态蚁群优化算法。对路径上的信息素进行自动变化原则,使得蚁群在进行路径搜索时更容易找到最优路径。黄国锐等<sup>[7]</sup>通过对蚁群算法路径选择进行研究,认为可以通过相互合作进行路径偏好选择,相邻蚂蚁可以相互影响其信息素释放。Deng W 等<sup>[8]</sup>针对蚁群算法前期信息素匮乏的问题,提出可以通过粗略搜索获得初始化信息素,然后再通过蚁群算法进行最优路径的搜索。马贵平等<sup>[9]</sup>对蚁群算法各个节点之间的道路上信息素的最大最小值进行限制,从而可以改变路径选择的概率,加快迭代速度。

以上文献中都对现代启发算法进行改进,但考虑得还不够全面。如蚁群算法在改进的过程中有的只考虑局部最优解问题<sup>[10]</sup>,或是只考虑优化速度问题,并没有同时解决这两个问题。对于平均最优路径都没有进行改变,说明最终在搜索方面,蚂蚁的搜索路径仍较为分散。

如何对快递末端配送进行优化,都雪静等<sup>[11]</sup>提出了一种快递自提柜放置在地铁口的设计理念,通过这种办法降低运输成本、减少货物到达客户手中的时间等。但是将快递放在地铁口会让用户从家出来拿快递花费的时间更长,并不会减少用户所等待的时间,没有方便用户。李玲玉等<sup>[12]</sup>则提出在快递末端配送的类 TSP 问题上采用 C-W 节约算法对快递配送路径进行优化,所找出的最佳路径平均缩短了 7.8% 的里程,但是在搜索速度上还有进一步的优化空间。李凤坤等<sup>[13]</sup>对蚁群算法多目标搜索问题,选择将多目标变为单目标进行搜索遍历,最后利用遗传算法对目标路径进行优化,找出最优解。但此算法中由于多目标合成单目标是避免了极值的影响,也失去了路径优化过程中的多样性,容易在遗传算法后期陷入局部最优解,冗余过大。叶威惠等<sup>[14]</sup>对快递配送过程中的旅行商问题进行研究,对遗传算法染色体的产生进行改进。采用当前节点最近点作为染色体中的一个染色体进行初始化,从而在搜索过程中可以快速找到最优解。Qi Chengming 等<sup>[15]</sup>在使用蚁群算法计算路径优化问题时,结合 PLS(pareto local search)算法对蚁群算法找到的路径做进一步优化。但是 PLS 是一个针对单目标

问题的本地搜索算法的拓展,在多目标搜索处理上并不能有效且快速地找到最优路径。

为此,文中提出一种改进蚁群算法来解决快递配送问题中的 TSP 问题,采用遗传-蚁群相结合的方式,对遗传算法的启发函数进行改进。首先,针对蚁群算法前期搜索较慢的问题引入启发函数来进行信息素更新。其次,对蚁群算法易于陷入局部最优解的问题,对蚂蚁搜寻的路径进行交叉、变异操作。最后,对蚁群算法的信息素挥发因子进行改进,利用遗传算法的启发函数使信息素挥发函数自适应变化,从而快速找到最优解。

## 1 相关算法分析

### 1.1 传统遗传算法

传统的遗传算法(GA)<sup>[16]</sup>是对生物界所存在的生物进化过程进行模拟而提出的一种搜索最优解的方法。最早于 1975 年提出,其本质是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型。在计算机仿真时,对求解的问题进行转换,转换成为类似生物体内染色体基因的形式,每一个节点就是一个染色体。并根据“物竞天择、适者生存”原则对染色体进行进化,引入选择、交叉、编译三个基本操作算子,加快其搜索能力。

遗传算法将需要进行搜索的问题进行演化变为一个种群,再将这个种群经过基因编码成为一条染色体。在每一次进行演化时,引入适应度函数对染色体个体进行计算,引入选择、交叉、编译操作算子,产出代表新的种群。这样,经过一系列操作所产生的种群会更有竞争力,其最优个体可以认为是最优解。但是遗传算法也存在一些问题,如搜索速度慢、易陷入“早熟”,导致后期冗余过大。因此常对启发函数进行改进以提高搜索能力,避免过早陷入“早熟”。所以,目前对于遗传算法的改进一直在进行,如:Mei 和 Wang 等<sup>[17]</sup>提出了一种称为蜂王演化遗传算法,实现了使用运算符序列来进行迭代执行以寻找最优路径。

### 1.2 传统蚁群算法

蚁群算法(ACO)<sup>[2-3]</sup>是对自然界中蚂蚁群体觅食行为进行模拟而提出的一种启发式搜索算法,于 20 世纪 90 年代首先提出。算法因可进行并行计算,鲁棒性好和正反馈机制被广泛地应用在车辆调度、路径优化、网络路由等问题上。生物界中蚂蚁进行觅食,刚开始会随机进行路径的选择。但是蚂蚁在路径中进行爬行时会分泌一种物质,这种物质会影响其他蚂蚁在觅食时对路径的选择。路径上存在的信息素越多,那么蚂蚁选择这条路径的概率就越大,这样最终可以找到最优的觅食路径。在蚁群算法中,首先设置初始蚂蚁种

群,通过轮盘赌方法将其分布在各个点,并根据其状态转移函数对蚂蚁下一步移动节点进行计算。其公式如下:

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allow}_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, & s \in \text{allow}_k \\ 0, & s \notin \text{allow}_k \end{cases} \quad (1)$$

其中,  $p_{ij}^k$  表示蚂蚁  $k$  在  $t$  时刻从节点  $i$  到节点  $j$  的状态转移概率;  $\tau_{ij}(t)$  表示  $i, j$  这条路径上的信息素浓度;  $\alpha$  表示对  $\tau_{ij}(t)$  重要程度的一个表示,值越大,说明在蚁群进行路径选择时,路径选择偏好作用就越大;  $\beta$  表示启发函数重要程度因子,值的大小与启发函数所起到的作用挂钩;  $\eta_{ij}(t)$  表示启发函数,表示蚁群对  $i, j$  这条路径选择的期望值,公式如下:

$$\eta_{ij}(t) = 1/d_{ij} \quad (2)$$

其中,  $d_{ij}$  表示节点  $i$  到节点  $j$  的距离。  $\text{allow}_k$  表示蚂蚁  $k$  还未访问的节点集合,在蚂蚁  $k$  进行搜索的过程中,已访问的节点会放到禁忌表  $\text{tabu}_k$  中,未访问的放到  $\text{allow}_k$  中。由状态转移概率公式可以看出,距离当前节点信息素越高的节点下次被访问到的概率就越大。所以在蚁群算法中需要对路径上的信息素进行更新,信息素会进行挥发,常设  $\rho$  来代表挥发因子。在信息素进行挥发的同时还需要再将在该时刻上其他蚂蚁所释放的信息素进行累加。其公式如下:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}, 0 < \rho < 1 \\ \Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \end{cases} \quad (3)$$

其中,  $\Delta\tau_{ij}^k$  表示第  $k$  只蚂蚁在  $i, j$  这条路径上释放的信息素浓度,  $\Delta\tau_{ij}$  则表示所有蚂蚁在  $i, j$  这条路径上释放的信息素总和。对于  $\Delta\tau_{ij}^k$  的计算公式分为三种模型:蚁周系统、蚁密系统、蚁量系统。主要区别在于信息素释放的时机不同。三种系统用公式表示如下:

蚁周系统:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{第 } k \text{ 只蚂蚁从节点 } i \text{ 到节点 } j \\ 0, & \text{其他} \end{cases} \quad (4)$$

蚁密系统:

$$\Delta\tau_{ij}^k = \begin{cases} Q, & \text{第 } k \text{ 只蚂蚁从节点 } i \text{ 到节点 } j \\ 0, & \text{其他} \end{cases} \quad (5)$$

蚁量系统:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}}, & \text{第 } k \text{ 只蚂蚁从节点 } i \text{ 到节点 } j \\ 0, & \text{其他} \end{cases} \quad (6)$$

其中,  $Q$  为信息素总量,  $L_k$  表示蚂蚁  $k$  在一次迭代中所通过路径的总长度,  $d_{ij}$  表示节点  $i$  到节点  $j$  之间的

距离。

蚁群算法较其他启发算法,搜索能力更强,所以在路径搜索方面使用蚁群算法较好。但是,在 TSP 问题的搜索上,虽然可以取得良好的效果,能够找到一条可以遍历所有节点的最短路径,却仍存在一些缺点:

- (1) 当节点过多时,计算时间较长。
- (2) 前期由于信息素缺乏,搜索速度较慢。

(3) 本质上要求所有蚂蚁选择同一条路,该线路即为最优线路。但是在实际的使用中,给定迭代次数,很难达到这个条件,使找到最优路径在某种情况下可能只是局部最优解。

## 2 改进蚁群算法

### 2.1 信息素初始矩阵更改

传统蚁群算法信息素初始矩阵一般设置为 0 或者 1,这样会造成前期蚁群算法信息素匮乏的问题,导致前期搜索过慢,因此本算法对信息素初始矩阵进行改进。由于遗传算法因其启发函数的存在,导致前期搜索速度较快,设计信息素初始化启发函数对蚁群算法信息素初始矩阵进行设置。对遗传算法启发函数公式的改进如下:

$$\text{Tau}(i, j) = \sum_{i=1}^n \sum_{j=1}^n \frac{\text{maxval} - \text{aviDistances}(i, j) + \text{min}}{\text{maxval} - \text{minval} + \text{min}} \quad (7)$$

其中,  $n$  为节点的个数,  $\text{aviDistances}$  为每个节点到给定点之间的距离占总路径长度的概率,  $\text{maxval}$  为  $\text{aviDistances}$  中的最大值,  $\text{minval}$  为  $\text{aviDistances}$  中的最小值,  $\text{min}$  则为弥补精度采用的一个极小值。

$\text{aviDistances}$  公式为:

$$\text{aviDistances}(i) = \sum_{i=1}^n \sum_{j=1}^n \frac{D(\text{st}, j)}{\text{sumDistances}(i)} \quad (8)$$

其中,  $D$  为各个节点之间的距离,  $\text{st}$  为给定起始点,  $\text{sumDistances}$  为总路径长度,公式如下:

$$\text{sumDistances}(i) = \sum_{i=1}^n \sum_{j=1}^n D(\text{st}, j) \quad (9)$$

通过计算出起始点到其他各个节点的概率,其距离越近,概率越大,符合蚁群系统中蚂蚁根据信息素寻找路径的规则。所以将此结果  $\text{Tau}$  作为蚁群算法信息素初始矩阵有助于蚁群算法避免前期搜索较慢的问题。

### 2.2 迭代中引入变异、交叉操作

在蚁群算法进行最优路径的搜索时,蚂蚁会根据路径上信息素的大小来进行路径的选择。但是,可能存在所选的最优路径只是某局部的最优路径。这时,如果问题本身属于凸优化问题,则可认为这个解是整体最优解。如果问题是非凸优化问题,则无法保证这个解是整体最优解。所以,引入交叉、变异操作,对迭



代中的路径尽可能多样化,突破局部最优解的问题。交叉操作会随机选择将两条路径进行一定的交叉形成一条新的路径,变异操作会对路径中的某一个节点进行替换操作,突破当前的搜索限制,更容易找到最优解。为此,在本算法中设置轮盘赌大小 tournamentSize 的值。然后进行 tournamentSize 次的轮盘赌选择,选择出随机的 tournamentSize 个路径,选择其中距离最小的路径。重复两次,这样将两个最小路径进行交叉操作。交叉的结果进行变异,将变异后的结果存储。

整个过程重复  $m$  次,这样最终用经过交叉、变异的路径替代原有的路径,避免陷入局部最优解的问题。

### 2.3 信息素更新规则更改

传统蚁群算法当节点过多时迭代次数较多,计算时间较长且其计算结果的平均路径较为分散。所以,对蚁群算法的信息素更新公式进行改进,解决所存在的问题。传统的蚁量、蚁密和蚁周系统局限性太大,信息素的更新容易陷入局部最优解,且在一定的迭代次数中,蚂蚁难以找到一条最优路径进行共同遍历,平均距离较大,说明蚂蚁寻找的路径波动较大。对这一问题,文中对蚁群算法的信息素更新公式进行改进,公式如下:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}, 0 < \rho < 1 \\ \Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \end{cases} \quad (10)$$

$$\Delta\tau_{ij}^k = \frac{Q}{\text{aviDistances}(i,j)} \quad (11)$$

其中,  $\text{aviDistances}(i,j)$  计算的是起点到每个点的概率,其公式为:

$$\text{aviDistances} = \sum_{i=1}^n \sum_{j=1}^n D[\text{Table}(i,j), \text{Table}(i,j+1)] \quad (12)$$

其中,  $D$  为各个节点之间的距离,  $\text{Table}$  为当前蚁群所走的路径。对信息素更新公式进行自适应计算,有助于蚁群快速找到最优解。在多节点计算时,可以更快地得到最优解,减少迭代次数和迭代时间。

### 2.4 改进蚁群算法实现步骤

(1) 利用公式(9)计算每条路径上的总路径长度,之后利用公式(8)计算每两个节点之间的距离所占总路径的比例,计算出其选中的概率。再之后利用公式(7)计算出初始信息素矩阵  $\tau(i,j)$  的值。

(2) 设置蚁群算法的初始种群蚂蚁数量  $m$ , 迭代次数  $\text{iter}$  初始值及最高迭代次数  $\text{iter\_max}$ 。设置信息素启发函数系数  $\alpha$ , 期望启发函数  $\beta$  以及信息素挥发系数  $\rho$ 。设置常系数  $Q$ , 起点位置, 初始化禁忌表, 设置交叉和变异概率的值。

(3) 寻找路径, 将起点加入到禁忌表中, 根据状态转移概率公式计算下一可达节点的概率, 根据轮盘赌方法选择下一到达节点, 直到节点全部加入禁忌表。

(4) 利用交叉、变异对禁忌表进行操作。

(5) 利用改进信息素更新公式更新信息素矩阵。

(6) 判断当前迭代次数是否大于最大迭代次数, 如果大于则输出最优路径, 否则继续进行迭代搜索。

(7) 得到最优路径, 算法终止。

改进蚁群算法具体流程如图1所示。

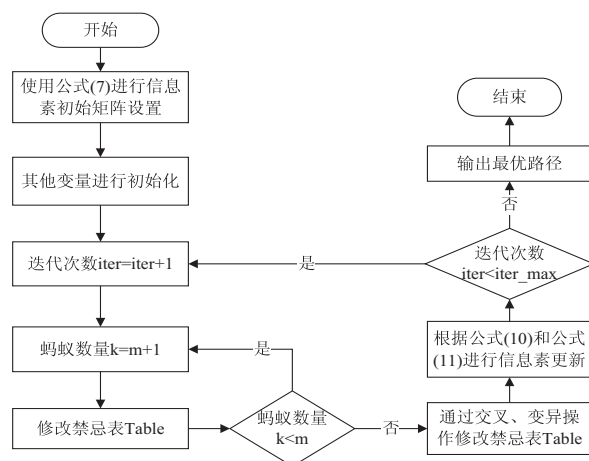


图1 改进 ACO 算法流程

## 3 实验仿真与结果分析

针对快递末端配送问题,对提出的算法进行仿真实验。仿真使用 Matlab 工具进行仿真,随机设置 31 个点的坐标,其中一个坐标为快递物流转运站,其他点是各个小区快递柜。设置蚁群算法各变量值,其中蚂蚁数量 50,最大迭代次数为 100。信息素重要因子  $\alpha$  经过测试设置为 1,当值为 1 时搜索性能最好。启发函数重要程度因子  $\beta$  值过小,会导致局部最优解,经过测试  $\beta$  取值 5 其搜索性能最好。信息素挥发因子  $\rho = 0.1$ ,交叉概率为 0.8,变异概率为 0.1。建立的模型如图2所示。

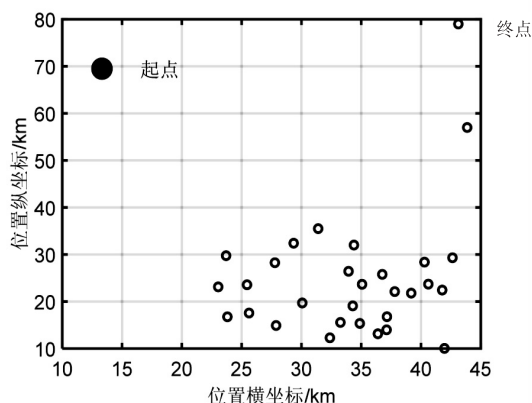


图2 地点分布

图中黑色实心点是快递公司物流转运站,其他点

是 30 个小区快递柜坐标。使用蚁群算法以快递站为起点进行 TSP 搜索遍历,对改进蚁群算法和传统蚁群算法的搜索效率进行对比。

图 3 是传统蚁群算法迭代 100 次搜索的最短路径,其最短路径路程长度是 229.94 km。图 4 为改进蚁群算法迭代 100 次搜索的最优路径,其最短路径路程长度为 224.88 km,通过计算可知,改进算法比传统算法在搜索总长度上优化了 2.2%。

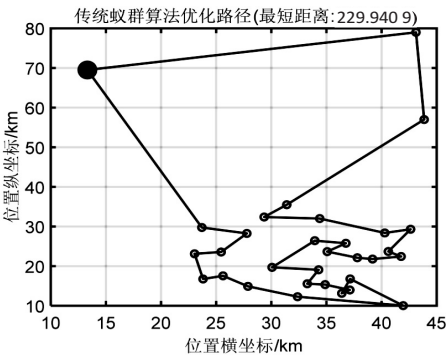


图 3 传统蚁群算法最优路径

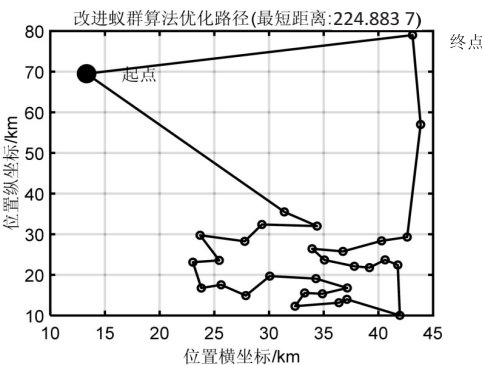


图 4 改进蚁群算法最优路径

对两种算法的实验结果进行比较,最优路径对比如图 5 所示。从图中可以看出,改进蚁群算法比传统蚁群算法更早地找到了最优解,且突破了传统蚁群算法的局部最优解问题。通过计算得知,改进蚁群算法比传统蚁群算法的效率提高 64%。

在图 6 中可以看出,在平均路径方面改进蚁群算法比传统蚁群算法性能更好。改进蚁群算法从第 15

次开始,其平均路径在 230 左右波动,而传统蚁群算法在第 50 次才趋于稳定波动,其波动值在 260 左右。改进蚁群算法比传统蚁群算法在波动时间方面优化了 70%,在波动值稳定性方面优化了 11.5%。

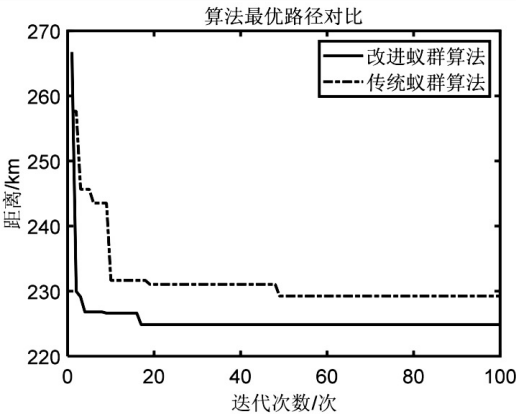


图 5 最优路径对比

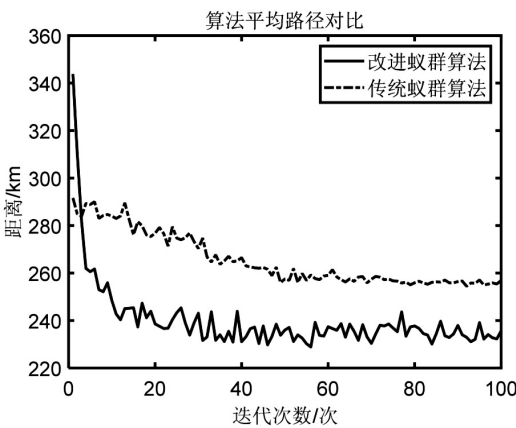


图 6 平均路径对比

在最终迭代结果中查看最后一次禁忌表,查看其 50 只蚂蚁搜索路径,将两种算法进行对比。图 7 是传统蚁群算法最后迭代禁忌表,可以看出其蚂蚁搜索仍较为分散,不符合最优解的规定。图 8 是改进蚁群算法最后迭代禁忌表,可以看出其蚂蚁搜索已经趋于一致,符合最优解的规定,解决了 2.2 节提出的传统蚁群算法在一定迭代次数中无法解决蚁群搜索路径一致性的问题。

15	9	10	28	27	30	31	1	29	11	24	20	25	21
15	9	10	28	27	30	31	29	1	12	14	13	7	6
15	9	10	28	27	30	31	1	29	11	23	16	5	2
15	9	10	28	27	30	31	29	1	12	14	13	7	6
15	9	10	28	27	30	31	29	1	12	14	13	11	23
15	9	10	28	27	30	31	29	1	12	14	13	7	6
15	9	10	28	27	30	31	1	12	14	13	11	24	20
15	9	10	28	27	30	31	29	1	12	14	13	7	6
15	9	10	28	27	30	31	1	29	11	23	16	5	6
15	9	10	28	27	30	31	1	29	11	24	20	25	26
15	10	28	27	30	31	29	1	12	14	13	11	23	19
15	9	10	26	21	22	18	3	17	19	24	20	25	30
15	9	10	28	27	30	31	1	29	11	23	6	5	2
15	9	10	28	27	30	31	29	1	14	12	13	11	23
15	9	10	28	27	30	31	29	1	12	14	13	7	6
15	9	10	28	27	30	31	29	1	14	12	13	7	6

图 7 传统蚁群算法最后迭代禁忌表

