

自适应蚁群算法求解最短路径和 TSP 问题

易正俊¹, 李勇霞¹, 易校石²

(1. 重庆大学 数学与统计学院, 重庆 401331;
2. 重庆师范大学 数学科学学院, 重庆 401131)

摘要:对传统蚁群算法的初始化信息素浓度加入方向引导,避免蚁群在初始阶段盲目地随机搜索浪费较多的时间;在全局信息素更新过程中加入双曲正切函数作为动态因子,自适应地更新每次迭代较优解路径的信息素浓度,增大算法获取全局最优解的可能性。两个算例采用改进的蚁群算法进行优化,优化的结果与实际情形具有良好的一致性,说明了改进算法的有效性和实用性。

关键词:蚁群算法;最短路径;方向引导;动态因子;旅行商问题

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2016)12-0001-05

doi:10.3969/j.issn.1673-629X.2016.12.001

Solving of Shortest Path Problem and TSP with Adaptive Ant Colony Algorithm

YI Zheng-jun¹, LI Yong-xia¹, YI Xiao-shi²

(1. College of Mathematics and Statistics, Chongqing University,
Chongqing 401331, China;

2. College of Mathematics and Statistics, Chongqing Normal University,
Chongqing 401131, China)

Abstract: Direction guiding is utilized in the initial pheromone avoiding ant colony in the initial stage to blindly random search and to waste more time. Moreover, a dynamic factor (hyperbolic tangent function) is invited in the global renewal process to update adaptively the pheromone concentration on the optimal path, in which way the possibility of obtaining the global optimal solution is increased. Then two examples are optimized with the improved algorithm, and the optimization results are in step with the actual, illustrating the effectiveness and practicability of the improved algorithm.

Key words: ant colony algorithm; shortest path; direction guiding; dynamic factor; TSP

0 引言

蚁群算法(Ant Colony System, ACS)是由意大利学者Dorigo等于20世纪90年代初期提出的一种基于种群的启发式随机搜索算法^[1-4]。蚁群算法具有正反馈、鲁棒性、并行性等优点,有很多文献采用蚁群算法计算最短路径的问题。最短路径问题在交通运输、物流配送、网络分析、管道铺设、厂区选取等领域有广泛的应用。如交通运输往往需要尽可能降低运输成本,等价于搜索运输网中两节点的最短路径;但传统的蚁群算法在计算较复杂的网络图中两节点间的最短距离

时容易陷入局部最优,且随着网络图的复杂程度的增加其收敛速度慢,还有可能出现停滞现象^[5-6]。针对这些问题,有学者对传统的蚁群算法做了一些相应改进。有的将信息素浓度控制在一定范围内变化,避免蚂蚁都选择同一条路径,避免算法中的停滞现象。Stuzle等^[7]提出“最大最小蚁群系统”(Max-Min Ant System);Clornei等^[8]将蚁群算法和遗传算法相结合,增强了算法的寻优能力;郑卫国等^[9]提出带奖罚的蚁群算法,对较优信息素浓度进行奖励,对较差信息素浓度进行惩罚。这些改进的蚁群算法虽然获得了较好的

收稿日期:2016-01-27

修回日期:2016-05-11

网络出版时间:2016-11-21

基金项目:国家自然科学基金资助项目(69674012);重庆市科技攻关计划(CSTC2009AC3037)

作者简介:易正俊(1963-),男,博士,教授,全国专业学位研究生教育教指委专家,研究方向为人工智能、仿真系统、数据挖掘;李勇霞(1988-),女,硕士生,研究方向为智能算法。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20161121.1641.026.html>

最优解,但初始搜索时间较长,全局更新规则中没有考虑较优解。

文中对初始信息素浓度进行方向的改进:距离目标方向越近的路径,就提高此路径上信息素浓度比例,这样就会促进蚂蚁在寻找食物源时以最大概率沿着信息素浓度高的路径,用最短时间内接近最优食物源的快速通道,避免初始搜索时间较长的缺陷;由于传统蚁群算法的动态因子在调节全局信息素浓度时不具有自适应性,因此引入双曲线正切函数在全局的信息素浓度更新中作为其自适应动态因子,每次迭代最优解路径的信息素浓度时具有一定的自适应性,使最优路径的信息素增大,从而增大了算法获取最优解的可能性,缩短了算法的收敛时间。

1 最短路径问题的描述

设一个带权重的有向图 $G = (V, \{L\})$, 其中 V 是含有 n 个节点的集合, L 是边的集合, 每条边 $(i, j) \in L$ 都赋予一个非负权重 d_{ij} , 表示节点 i 到节点 j 间的距离大小, $i, j \in V$ 。设 B, E 分别为 V 中的任意两点, 最优路径问题就是在带有权重的有向图中, 寻找从点 B 到 E 的一条具有最小权重总和的路径。

2 蚁群算法的改进

一种仿生算法—蚁群算法, 是模拟自然界蚂蚁寻找食物过程而得出。在此过程中, 它们总能找到从巢穴到食物源之间的一条最短路径。其原因在于蚂蚁在路径上会释放一种信息激素 (pheromone) 进行信息传递, 蚂蚁在运动过程中能够感知这种物质, 由它来指导运动方向。当它们碰到一个陌生路口时, 就会随机地选出一条路径进行前行, 同时蚂蚁会在这条路径上释放一定量的信息素, 此时路径上的信息素浓度增大, 后来的蚂蚁再次碰到这个路口时, 就会选择信息素浓度高的路径, 这样就形成一种正反馈。最优路径上的激素浓度越来越大, 最终整个蚁群就会搜索出一条最短路径。

2.1 算法初始时刻浓度改进

传统的蚁群算法在初始迭代时给每条路径上的浓度是一个常数, 那时蚂蚁不知道每条路径的长度, 它只能随机选择, 产生了大量无关紧要的路径, 阻碍最优路径搜索过程, 同时对信息素浓度局部更新也会产生影响。但在求最短路径时, 节点与节点之间的距离是已知的, 因此在用蚁群算法求最短路径时, 没有必要进行盲目选择, 可以对迭代开始时的初始浓度进行精确赋值。

设在网络图中需要求出从点 B 到 E 的一条最短路径。其中 $\tau_{ij}(0)$ 为节点 i 和节点 j 之间路径上的初始信息素浓

度为 $\tau_{ij}(0)$, 反映靠近最短距离的程度, 越靠近最短距离, 浓度应该越大。用 d_{BE} 表示起点 B 到终点 E 的直线距离, d_{Bj} 表示节点 B 与节点 j 的直线距离, d_{jE} 表示节点 j 到终点 E 的直线距离, $d_{Bj} + d_{jE}$ 表示 $B \rightarrow j \rightarrow E$ 的曲线距离, $d_{BE}/(d_{Bj} + d_{jE})$ 中 $d_{Bj} + d_{jE}$ 越趋向于 d_{BE} , $d_{BE}/(d_{Bj} + d_{jE})$ 越接近于 1, 此条路径上的浓度越大, 蚂蚁就越偏向选择这些节点作为移动方向。因此节点 i 和节点 j 之间路径上的初始信息素浓度为 $\tau_{ij}(0)$ 定义为:

$$\tau_{ij}(0) = \frac{d_{BE}}{d_{Bj} + d_{jE}} \quad (1)$$

算法的这种改进使得蚂蚁在搜索过程产生了比较合理的方向引导, 进而对无关路径起到抑制作用, 避免蚂蚁的盲目搜索, 提高了搜索全局最优解的速率。

2.2 全局更新规则的改进

传统蚁群算法在全局更新规则中只更新每次迭代的最优路径上的信息素浓度。这样就导致某些较短路径信息素浓度过分增加, 全局最优路径与单次迭代的最优路径相关性较大^[10-12], 易陷入局部最优。在全局更新规则中, 引入自适应动态因子 σ ($\sigma \in (0, 1)$), 这样不仅可以实时动态更新每次迭代最优路径的全局信息素浓度, 而且也要动态更新较优的局部最优解。经过自适应控制单次迭代最优解的信息素浓度在当前最优解中的比重, 从而较优路径信息素浓度发挥了更大的作用, 更好地反映了路径信息, 最优解逐渐凸现, 加快了全局最优解的收敛速度。

传统算法的动态因子为 $f(x) = \frac{1}{2} \operatorname{sgn}(x) + \frac{1}{2}$, 可以看出它仅仅在全局最优路径上取 1, 也就是说只加强最优路径上信息素浓度, 而较优路径上的信息素浓度没有及时更新。故对较优路径信息素浓度更新不具有自适应性, 因此文中选用的自适应动态因子为双曲正切函数 $f(x) = \frac{1}{2} \tanh(\omega x) + \frac{1}{2}$, 输出为平滑曲线, 取值范围为 $[0, 1]$ 。其中 ω 为控制双曲正切函数形状的参数。不仅最优路径信息素浓度得到更新, 更重要的是较优路径也得到了自适应更新。在完成每次迭代后, 更新本次迭代最优解路径信息素浓度, 规则如下:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sigma \mu \Delta \tau_{ij} \quad (2)$$

$$\sigma = \frac{1}{2} \tanh\left(\omega \frac{L_{\text{localmin}} - L_{\text{min}}}{\bar{L} - L_{\text{min}}}\right) + \frac{1}{2} \quad (3)$$

其中, \bar{L} 为当前局部最优解的平均长度; L_{localmin} 为此次迭代的最优路径长度; L_{min} 为当前全局最优解; μ 为给定参数。

对于自适应因子 σ , 当 L_{localmin} 越大, 即搜索路径越

长时,动态因子 σ 越趋于0;而当 L_{localmin} 越小,搜索路径越短时,动态因子 σ 越趋于1。所以搜索到的路径长度越短,全局更新时它所遍历的路径信息素浓度越强,信息量增加就会较快。当 ω 取不同值时,双曲正切函数曲线以及与传统蚁群算法动态因子的对比图如图1所示。

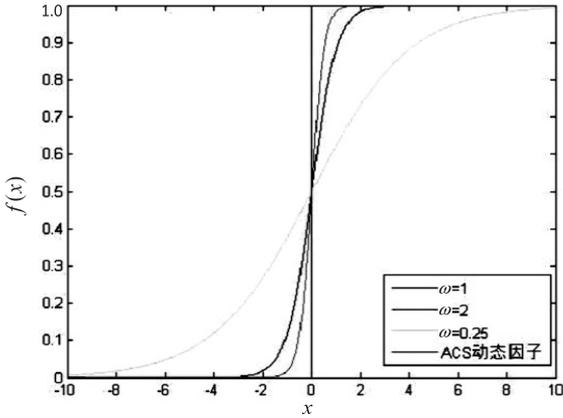


图1 动态因子函数曲线图

对于双曲正切函数 $f(x)$,当 $x \in (-1, 1)$ 时,可以看到 $f(x)$ 变化率最大,那么敏感度就较高,这样有效区分了不同迭代最优解对信息素加成的影响,加强局部最优解中路径更短的信息素浓度,加快收敛的全局最优解的速度;当 $x < -1$ 时, $f(x)$ 缓慢趋向于0,这样不但对不利解对信息素加成起到抑制作用,同时保留了其部分影响,进而也增加了解空间的多样性;当 $x > 1$ 时, $f(x)$ 缓慢趋于定值1,这样更好地避免部分较短但不是全局最优路径被过度增强,有效地防止算法陷入局部最优解。与此同时,对于自适应动态因子 σ 是平滑过渡,故不会对信息素浓度造成波动影响。从图中可以看出 ω 越大, $f(x)$ 对 x 的灵敏度就越高。

3 自适应蚁群算法的步骤

以上是对传统蚁群算法的改进,改进后的算法实现步骤如下:

步骤1:初始化信息启发因子 α (轨迹的相对重要性),期望启发因子 β ,路径启发因子 $\eta_{ij}, \eta_{ij} = 1/d_{ij}$ 。其中, d_{ij} 为 $i \rightarrow j$ 的欧氏距离; q_0 为蚂蚁状态转移的阈值。设最大迭代次数 N ,选定蚂蚁数 m 。

步骤2:计算节点距离矩阵,初始化信息素浓度 $\tau_{ij}(0)$,将 m 只蚂蚁放在起点 B ,将各蚂蚁的初始节点 B 置于当前解集 tabu_k 中,第 k 只蚂蚁从节点 i 转移到下一个节点 j 的规则:在0与1之间随机生成一个实数 q ,把它与事先给定的阈值 $q_0 \in (0, 1)$ 进行比较:

当 $q \leq q_0$ 时,第 k 只蚂蚁选取下一个节点 j 为与 i 相连的边 ij 的权重 $\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)$ 最大值所对应的节点,即 $j =$

$$\arg \max_{s \in \text{allowed}_i^k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)。$$

当 $q > q_0$ 时,以轮盘赌的方式计算第 k 只蚂蚁由 i 转移到下一个节点 l 的概率:

$$p_{kil}(t) = \begin{cases} \frac{\tau_{il}^\alpha(t) \eta_{il}^\beta(t)}{\sum_{s \in \text{allowed}_i^k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, & l \in \text{allowed}_i^k \\ 0, & l \notin \text{allowed}_i^k \end{cases} \quad (4)$$

其中, $\text{allowed}_i^k = \{\text{与} i \text{ 相连的节点}\} - \{\text{蚂蚁} k \text{ 已经访问过的节点}\}$,即每只蚂蚁对每个节点最多访问一次。选取概率最大的节点 j ,即: $j = \arg \max_{i \in \text{allowed}_i^k} p_{kij}(t)$ 。

步骤3:信息素浓度的局部更新。在蚂蚁完成一次搜索后,信息素一方面要挥发掉一部分,另一方面蚂蚁在走过的路径上要释放一定量的信息素。设挥发和增加信息素的系数均为 ρ ,更新规则如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) \quad (5)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{kij}(t) \quad (6)$$

其中, $\Delta\tau_{kij}(t)$ 为本次循环中蚂蚁 k 对边 (i, j) 的信息素增量,若蚂蚁没有经过此路径,则 $\Delta\tau_{kij}(t) = 0$,否则,用 Q 表示增加浓度的总和,可视为一个常数,用 L_k 为第 k 只蚂蚁在本次循环搜索中搜索到的路径长度,则定义 $\Delta\tau_{kij}(t) = \frac{Q}{L_k}$ 。式(5)表示下一次迭代路径的信息素浓度为已挥发剩下的浓度与新释放的信息素浓度之和。

步骤4:重复步骤2和3,直到所有蚂蚁都到达终点 E ,此时得到 m 条由初始节点到终点 E 的路径 $\{l_1, l_2, \dots, l_m\}$ 。

步骤5:比较 m 条路径的长度有 $l'_1 < l'_2 < \dots < l'_m$,记本次最优解为 L_{localmin} ,同时可求出 m 条路径的平均解 \bar{L} ,记已迭代的当前最短路径为 L_{min} ,按式(2)和式(3)对此路径上的信息素浓度进行全局更新。

步骤6:再将 m 只蚂蚁放置于起点 B ,按照步骤2~4进行搜索,这样一直重复进行,直到迭代 N 次,此时可以得到全局最优解。

4 算法仿真

4.1 最短路径问题求解

为检验文中自适应蚁群算法的有效性,以哈尔滨市部分交通路线图为例,以道路交通事故现场作为起点,以哈尔滨市人民医院作为终点,在事故和医院之间寻找一条最短路径,其目的是快速将伤员送往医院,最大可能地减少人员伤亡和财产损失。

选取的参数如表1所示,分别采用文中的自适应蚁群算法(M-ACO)、传统蚁群算法(Ant Colony Optimization,ACO)、最大最小蚁群算法(Max-Min Ant Sys-

tem, MMAS), 节点与节点间的距离数据来源于文献[13]。

表 1 3 种算法的公共参数设置

参数名称	参数值
Q-信息素	100
m-蚂蚁数	20
α -信息启发因子	1
β -期望启发因子	1
ρ -信息素浓度挥发因子	0.2
q_0 -给定参数	0.5
N-最大迭代次数	200
$\tau_{ij}(0)$ -参考信息素初始值	1
τ_{max} -信息素浓度上限	0.9
τ_{min} -信息素浓度下限	0.01
ω -给定参数	1

选取交通事故点离医院有 20 个路口即对 20 个节点网络进行仿真实验, 实际最短路径长度为 3 186.25 km, 用三种方法搜索的最短距离结果如表 2 所示。

表 2 20 路口网络实验数据统计结果

算法	最优解	平均解	搜索到最优解几率/%	平均迭代次数
M-ACS	3 186.25	3 192.61	96	46.12
ACS	3 186.25	3 652.74	8	21.03
MMAS	3 186.25	3 571.36	19	34.74

从表 2 可以看出: ACS 经历 21 次迭代后迅速陷入局部最优, 并且在 100 次中只有 8 次搜索到最优路径; 而搜索结果在平均解意义上有所提高的是 MMAS, 但平均迭代次数明显升高, 并且搜索到的最优解的几率只有 19%; 文中改进的蚁群算法的搜索结果在最优解和平均最优解上都有显著改善, 搜索到最优解的几率为 96%, 基本上每次都能找到最优解。

为进一步验证文中算法对路口数量多的适用性, 现扩大交通事故点与医院的距离, 路口增加即研究的网络节点数增加, 使得研究的网络图变得更复杂。分别对 20、30、40、50、60 个路口进行仿真, 如表 3 和表 4 所示。

表 3 最优解搜索率统计结果

算法	20	30	40	50	60
M-ACS	96	89	80	57	51
ACS	8	22	10	8	5
MMAS	19	38	41	19	11

由表 3 可知, 当路口数增加时, 改进的 M-ACS 算法仍然具有较好的路径搜索效果, 30 路口时最优路径搜索率为 89%, 远远优于其他两种算法。当路口数为 60 时, 由于网络拓扑结构较为复杂, 计算量较大,

参考算法搜索率较低, 而文中仍然有 51% 的搜索率。表 4 中, M-ACS 算法平均迭代次数略大于其他算法, 并且在路口数较少时, 迭代数增加并不明显, 但带来的搜索性能提升较大。

表 4 平均迭代次数统计结果

算法	20	30	40	50	60
M-ACS	46.12	50.01	62.73	94.06	146.11
ACS	21.03	17.55	40.88	29.90	66.17
MMAS	34.74	9.37	42.17	33.74	51.63

3 种算法对不同节点的收敛时间对比如图 2 所示。

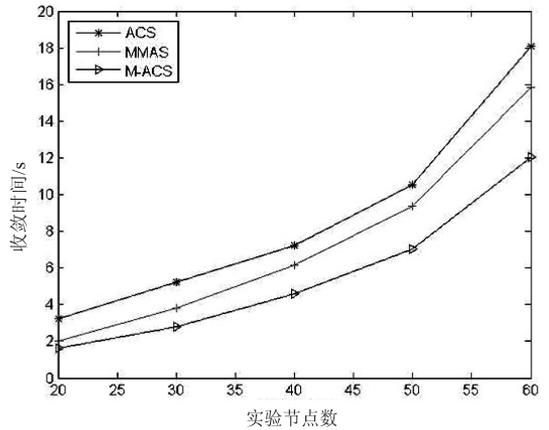


图 2 收敛时间对比图

从图 2 可以看出: 在不同复杂程度的网络图中, 3 种算法中收敛时间最短的是文中提出的自适应蚁群算法, 主要因为 M-ACS 在迭代初期, 对路径的信息素浓度进行了重新分配, 并给予了一定的方向指导, 节约了蚁群盲目的随机搜索时间。在迭代后期全局信息素浓度更新时加入了自适应动态因子, 动态更新每次迭代最优路径的全局信息素浓度, 使其最优路径上的信息素浓度进一步加强, 从而提高了算法的收敛速度。因此, 在不同节点的收敛时间均小于其他 2 种算法。

4.2 TSP 问题求解

旅游线路的优化问题是旅行商问题(TSP)的一种典型代表。近几年来对于 TSP 的求解提出了许多优化算法, 仿生算法是研究热点, 它有传统算法不可替代的优势。文中引用文献[14]中设计旅行线路为例, 直观地反映自适应蚁群算法与其他算法的优劣。为验证文中算法的性能, 引入遗传算法和传统的蚁群算法进行求解。选取参数如表 1, 设置出发点都为重庆, 且必须经过每一个城市且只有一次, 最后回到重庆。

由文献[14]可知, 利用遗传算法得出的最优路线如图 3 所示, 对应最优值为 18 997.8 km。图 4 为传统蚁群算法得出旅行路径的结果, 其对应的最优值为 18 696.1 km。从优化效果来看, 传统蚁群算法的寻优

效果比遗传算法的效果更好。

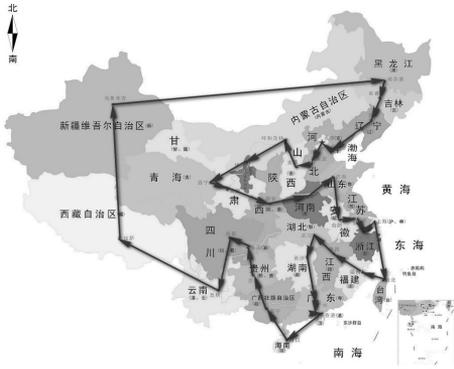


图 3 遗传算法运行结果

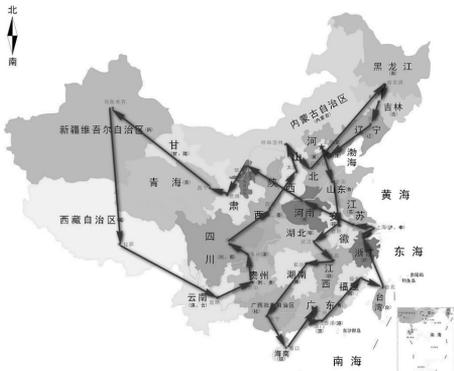


图 4 传统蚁群算法运行结果

图 5 为采用自适应蚁群算法设计旅行路径的结果。对应的线路如下:重庆→贵州→南宁→海口→澳门→香港→广州→长沙→合肥→南京→上海→杭州→台北→福州→南昌→武汉→郑州→太原→石家庄→济南→哈尔滨→长春→沈阳→天津→北京→呼和浩特→西安→银川→兰州→西宁→乌鲁木齐→拉萨→昆明→成都→重庆;所对应的最优值为 17 887.4 km。从效果来看,自适应蚁群算法的寻优效果比遗传算法和传统的蚁群算法的效果更好。

通过三种算法的比较可以得出:文中在迭代过程中采用自适应蚁群算法,在实际应用中能够得到更佳的效果且求解精度更高。

5 结束语

针对传统蚁群算法存在容易陷入局部最优解且收敛速度慢的缺陷,提出了自适应蚁群算法。通过在初始化信息素浓度时加入方向引导,在全局信息素更新过程中加入双曲正切函数作为动态因子,自适应地调整每次迭代最优解的信息素更新的改进策略。实验结果表明:自适应蚁群算法提高了求解最短路径的收敛速度和精度,增强了算法的寻优能力。为验证文中算法的有效性,将其用于 34 个城市旅游线路的优化,并与传统的蚁群算法和遗传算法进行比较。结果表明:文中算法寻优精度高,得到的线路最优。

参考文献:

- [1] Colomi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies [C]//Proceeding of the first European conference of artificial life. Paris:Elsevier Publishing, 1991.
- [2] Dorigo M, Ganbardella 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.
- [3] Colomi A, Dorigo M, Maniezzo V. Ant system for job-shop scheduling [J]. Belgian Journal of Operations Research Statistics and Computer Science, 1994, 34 (1): 39-53.
- [4] Dorigo M, Maniezzo V, Colomi A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on System Man and Cybernetics -Part B, 1996, 26 (1): 29-41.
- [5] 王 健, 刘衍珩, 朱建启. 全局自适应蚁群优化算法 [J]. 小型微型计算机系统, 2008, 29 (6): 1083-1087.
- [6] 付 宇, 肖健梅. 动态自适应蚁群算法求解 TSP 问题 [J]. 计算机辅助工程, 2008, 15 (4): 10-13.
- [7] Stuzle H. MAX-MIN ant system [J]. Future Generation Computer System, 2000, 16: 889-914.
- [8] Clomei I, Kyriakides E. Hybrid ant colony-genetic algorithm (GAAP) for global continuous optimization [J]. IEEE Transactions on Systems Man and Cybernetics-Part B, Cybernetics, 2012, 42 (1): 234-245.
- [9] 郑卫国, 田其冲, 张 磊. 基于信息素强度的改进蚁群算法 [J]. 计算机仿真, 2010, 27 (7): 191-193.
- [10] Shah S, Kothari R, Chandra S. Debugging ants: how ants find the shortest routs [C]//8th international conference on information, communications and signal processing. [s. l.]: IEEE, 2011: 1-5.
- [11] 吴虎发, 李学俊, 章玉龙. 改进的蚁群算法求解最短路径问题 [J]. 计算机仿真, 2012, 29 (8): 215-218.
- [12] 张学敏, 张 航. 基于改进蚁群算法的最短路径问题研究 [J]. 自动化技术与研究, 2009 (6): 4-7.
- [13] 哈尔滨市统计年鉴 [R]. 哈尔滨: 哈尔滨市统计局, 2014.
- [14] 走遍全中国方案的研究 [EB/OL]. 2014-05-30. <http://www.docin.com/p-105402921.html>.

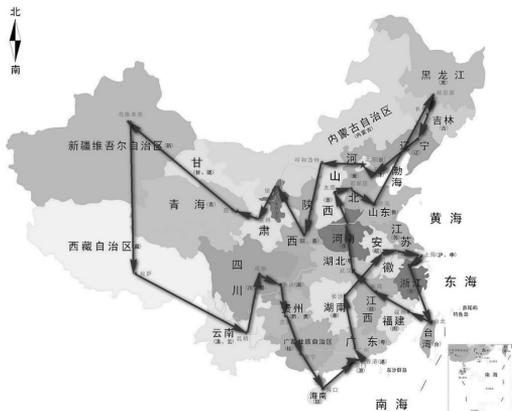


图 5 自适应蚁群算法运行结果