

# 用蚁群优化算法求解中国旅行商问题\*

燕忠, 袁春伟

(东南大学 生物医学工程系分子与生物分子电子学教育部重点实验室, 江苏 南京 210096)

**摘要:** 中国旅行商问题是一个组合优化问题, 是一个 NP 问题。本文提出用蚁群优化算法去解决, 同时提出了两种改进的方法, 其中, Ant-F 能够增强系统的搜索能力, 使系统避免早熟, 具有正负反馈的功能, 仿真简单, 容易理解; 而 ACS+ 是在 Ant Colony System (ACS) 的基础上改进而成, 它使系统在演化的后期能够通过适当增大系统区分信息素对比强度的方法, 尽快找到最优的解。和它的几种蚁群优化算法、遗传算法和模拟退火算法相比较, 实验表明, ACS+ 是本文提及的几种算法中最优的一种, 它能加快系统收敛的速度, 找到问题的最优值。

**关键词:** 蚂蚁系统; 蚁群优化算法; 旅行商问题

**中图分类号:** TP391.41 **文献标识码:** A

## 1 引言

旅行商问题就是指旅行商按一定的顺序访问  $N$  个城市的每个城市, 使得每个城市都能被访问且仅能被访问一次, 最后回到起点, 而使花费的代价最小。它是图论中具有代表性的优化组合问题, 是一个 NP 问题, 问题的求解随着问题规模的扩大 (城市数目的增加) 而变得越来越复杂, 所需的计算时间也越来越长, 采用某些传统的算法, 例如穷举搜索法, 是无法实现的, 所以寻找更新的优化算法是大家所追求的。这样, 就出现了一些新的随机的搜索算法, 像遗传算法 (Genetic Algorithm, GA)、贪婪算法 (Greedy Search)、模拟退火 (Simulated Annealing, SA) 等。但是这些算法对系统参数要求较高, 例如遗传算法就必须选择种群数目、杂交概率、变异概率等, 参数选得好, 就会得到比较好的实验结果, 反之, 可能导致实验的失败。

中国旅行商问题是多种旅行商问题中的一个, 它是一种对称式的旅行商问题。目前, 有关文献对这个问题进行了求解, 例如利用几何算法求得的解为 15492 公里<sup>[1]</sup>, 采用人机结合的优质穷举法求得的解为 15449 公里<sup>[2]</sup>, 用二叉树描述中国旅行商问题而所求得的解为 15404 公里<sup>[3]</sup>, 当然, 还有其它比较早方法, 其结果见文献[4,5]。在这里, 介绍一种蚁群优化算法, 它不仅具有很好的鲁棒性, 良好的正反馈特性, 而且具有并行分布计算的特点, 适合于解决旅行商问题。将其应用到中国旅行商问题上, 取得很好的实验结果。

## 2 蚁群优化算法

蚁群优化算法 (Ant Colony Optimization, ACO)<sup>[6]</sup>是由意大利学者 Dorigo M, Maniezzo V, Colomi A 等人根据蚂蚁群体具有智能的特点而首先提出来的, 当时称之为蚁群系统 (Ant System, AS)<sup>[7]</sup>, 并利用该方法去解决离散优化问题<sup>[8]</sup>, 取得成功, 后来 Dorigo M 等为了其他学者研究的方便, 将各种蚂蚁算法统称为蚁群优化算法, 并为该算法提出了一个统一的框架结构模型。蚁群优化算法是九十年代初期才提出的一种新型的进化算法, 虽然不很成熟, 但是对蚁群优化算法的研究已引起了国际上学者们的广泛关注。

研究表明, 蚁群优化算法具有很强的发现较好解的能力, 这是因为该算法不仅利用了正反馈原理, 在一定程度上可以加快进化过程, 而且是一种本质上分布并行的算法。但是这种算法也存在一些缺陷, 如搜索时间较长, 容易出现停滞现象 (stagnation behavior), 即搜索进行到一定程度后, 所有

\* 收稿日期: 2003-05-26 修订日期: 2003-11-04  
基金项目: 国家自然科学基金重点资助项目 (69831010)

个体所发现的解完全一致,不能对解空间进一步进行搜索,不利于发现更好的解,所以,不少学者提出了自己的改进算法,其中有 Dorigo M 提出的 Ant-Q System<sup>[9]</sup>、Ant Colony System<sup>[10]</sup>, Thomas 等人提出的 Max-Min Ant System<sup>[11]</sup>

蚁群优化算法的第一个应用是旅行商问题。之所以选择 TSP, 不仅因为该问题为大家所熟悉易懂, 是 NP 问题, 而且也较容易阐明蚁群优化算法的原理。用蚁群优化算法来解决旅行商问题的过程如下: 首先, 蚁群从同一地点或不同地点同时出发, 按照按照下面的概率公式逐次访问各个城市节点:

$$P_{ij}^k = \begin{cases} \frac{([\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta)}{\sum_{l \in J_k} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta} & \text{如果 } j \in J_k \\ 0 & \text{否则} \end{cases} \quad (1)$$

其中禁忌列表  $Tabu_k$  是保存了每只蚂蚁  $k$  已经访问过的城市的集合,  $J_k = \{N - Tabu_k\}$ ,  $\alpha$ 、 $\beta$  是系统参数, 分别表示信息素、距离对蚂蚁选择路径的影响程度;  $\eta_{ij}$  表示由城市  $i$  到城市  $j$  的期望程度, 可根据某种启发算法具体确定, 一般为  $1/d_{ij}$ ;  $\tau_{ij}$  为边  $L(i, j)$  上的信息素强度。

当蚁群完成了所有的节点的访问后, 在原路返回的过程中, 根据所得的解的好坏去修改路径上的信息素强度, 以此来引导其他蚂蚁对该路径的选择, 从而达到群体协作的目的, 最后判断系统是否满足停止的条件(停止条件可以是最大的迭代次数, 计算机运行时间, 或者是达到系统所要达到的数据精度等), 如果条件不满足, 则蚁群又重新开始搜索路径, 建立新的解; 否则, 系统将退出运行, 将所得的结果输出。从上面可以看出, 蚁群优化算法的基本思想就是质量越好的解和距离越短的路径就越能吸引更多的蚂蚁。蚁群正是通过这种反复记忆和学习的过程, 得到了最短路径, 即全局最优解。

Ant-Q System 与标准的蚁群优化算法(AS)相比, 有如下两点不同:

1) 选择下一个城市的规则不同, 存在勘探与开采的区分:

$$j = \begin{cases} \arg \max_{u \in J_k} \{[\tau_{iu}]^\alpha [\eta_{iu}]^\beta\}, & \text{如果 } q \leq q_0 \\ \text{按公式(1)选择 } j, & \text{否则} \end{cases} \quad (2)$$

$q_0$  是系统进行开采(exploration)和勘探(exploitation)的分界点,  $q_0$  的取值范围为  $[0, 1]$ ,  $q_0$  越小, 那么系统随机选择下一个城市的概率就越大。

2) 是修改信息素强度的公式不同, 有一个额外的信息素强化项:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho(\Delta\tau_{ij} + \gamma \max_{l \in J_k} \tau_{il}) \quad (3)$$

其中  $\rho$  为信息素的挥发速率, 为小于 1 的正数。之所以这样做, 其理由是一方面是为了防止信息素的无穷积累, 另一方面也是为了提高系统搜索更好的可行解的能力, 以免较早地失去探索新路径的能力;  $\Delta\tau_{ij}$  表示蚂蚁在本次运行中留在路径  $L(i, j)$  上的信息素强度:

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (4)$$

而  $\Delta\tau_{ij}^k$  表示蚂蚁  $k$  放置在边  $L(i, j)$  上的信息素强度,  $\gamma$  的意义与  $\rho$  相同。

Ant Colony System (ACS) 是 Dorigo M 等人在 1996 年基于 Ant-Q 算法的基础上提出的一种改进的蚁群算法, 在解决对称式的 TSP 和非对称的 TSP 上, 取得了很好的效果。它与 Ant-Q 相比, 既具有 Ant-Q 的特点, 能够区分系统的勘探与搜索, 又具有不同点于 Ant-Q 的特点, 这主要表现在信息素强度的修改上, 它使用了局部修改和全局修改的规则。

局部修改的规则是当蚂蚁个体找到一个解时, 就修改此个体所走路径上的信息素的强度, 公式为:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (5)$$

$\tau_0$  为一个很小的数, 经实验观察为  $\tau_0 = (n \cdot L_{nn})^{-1}$ ,  $n$  为城市数目,  $L_{nn}$  为接近优化值的一个值, 实验中取到目前为止系统所得的最优解。

全局修改的规则为: 当所有蚂蚁都完成了旅行, 访问了所有的城市并回到了起点, 系统这时就从这个蚁群中选择具有最好解的蚂蚁个体, 并将其所走路径上的信息素强度进行修改, 其公式为:

$$\tau(i, j) = (1 - \gamma)\tau(i, j) + \gamma(L_{best-iter})^{-1} \quad (6)$$

$L_{best-iter}$  为本次运行中蚁群中具有的最好解。这种修改规则有利于那些具有短路径的边上具有较高的信息素强度。

Max-Min system 的基本思想是仅让每一代中的最好个体所走路径上的信息素强度作调整, 以加快收敛速度, 这样便出现早熟的现象。为了避免早熟, 对信息素增强量上面施加限制。由于确定信息素强度的上下限比较复杂, 因此实践中很少应用它, 在下面的实验中就不进行仿真了。

### 3 改进方法

按照本文提出的改进方案, 首先, 按照蚁群优化算法寻找路径, 当所有的蚂蚁完成了搜索的路径的任务后, 累加所有蚂蚁个体在不同路径的边上所留下的信息素强度, 然后判断各条边上是否已经被蚂蚁个体访问过, 若已被访问过, 就根据下面公式修改其信息素强度:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum \Delta\tau_{ij}^k \quad (7)$$

$\sum \Delta\tau_{ij}^k$  为所有蚂蚁在这条边  $L(i, j)$  上的信息素的累加。若没有被访问过, 则不按信息素挥发公式计算, 也就是说, 这条边的信息素强度在本次的迭代中不产生变化。其目的是为了增大那些没有被访问到的边的搜索概率, 增强系统的搜索能力, 不致于使系统过早陷入局部最小值, 这是系统所具有的负反馈功能方面。所以这种系统具有正负反馈的功能, 将这种方法记为 Ant-F。

同时, 在实验中观察发现, ACS 到了系统演化的后期, 系统的信息素强度都比较小, 区分它们并不容易, 在搜索最优值方面很难有什么起色, 所以为了区别它们的信息素强度, 使系统能尽快找到最优的解, 本文提出在演化的后期, 适当增大系统区分信息素对比强度的方法, 记为 ACS+。在下面的实验中采取了在系统迭代的最大次数的 3/4 处开始增大系统参数  $\alpha$  为原来的 5 倍。

### 4 仿真

用这几种蚁群优化算法来解决中国旅行商问题 (CTSP31)。在以下的实验过程中, 采取了保留最优

解的策略。实验的有关参数为: 整个系统中蚂蚁的个数等于 TSP 中城市的个数, 每个城市中分布蚂蚁的个数为 0~3;  $\alpha=1$ ,  $\beta=2$ ,  $\rho=0.9$ ,  $\gamma=0.3$ ,  $q_0=0.9$ , 系统信息素强度的初始值为 10.0, 算法对不同问题求解的最大迭代次数为表 1、2 中的圆括号里面的数值, 如 GR17 (500), 表示 17 个城市的 TSP, 每次求解的最大迭代次数为 500 次 (最大迭代次数或者已经找到最优解作为算法的停止条件), 算法对不同问题求解的统计次数为 15 次。统计的实验结果如表 1 所示。需说明的是, 这里的标准 ACO 是指蚁群系统算法 (AS)。从表 1 中可以发现, 总体来说, ACS+改进的效果十分明显, 搜索到了要找到的当前最优值 (最优解的路线与文献 3 所找的路线一致), 找到最优值的概率达到了 73.3% (运行 15 次, 11 次找到了最好的结果), 是最好的一种算法 (其它的均不能在规定的时间内求出最优解)。

其它几种, 在最优值上, ACS 是比较好的, 其次是 Ant-F、标准的 ACO、Ant-Q; 在反映系统搜索最优值的稳定性方面上: 在平均值方面, 标准 ACS 比较好, 然后是标准的 ACO、Ant-F、Ant-Q; 在均方差方面, Ant-F 表现最差, 这也反映增大那些没有被访问城市 (或被访问的次数较少) 而被系统选择的概率的思想。同时在图 1 反映各种蚁群优化算法某一次实验演化的过程中, 可以看出, ACS+不仅摆脱前面求解停滞状态, 在第 3078 次迭代时使解的质量 (为 15415) 发生的变化, 在第 3550 迭代处找到最优解, 而且也说明了 ACS+在系统演化的最后阶段增大系统区分信息素对比强度  $\alpha$ , 能加快系统收敛。

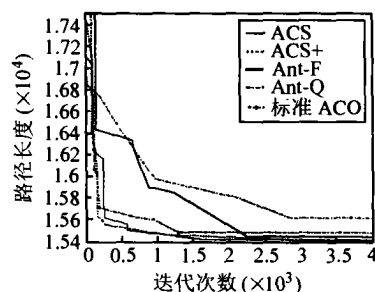


图 1 几种蚁群优化算法在解决 CTSP31 问题上的演化过程

表 1 几种改进的蚁群优化算法在 CTSP31 (4000) 问题上的实验结果

	标准 ACO	Ant-Q	ACS	Ant-F	ACS+
最优值	15483.0	15620.0	15420.0	15448.0	15404.0
平均值	15566.2	15686.0	15455.27	15685.4	15428.0
均方差	41.71	87.68	23.54	106.10	43.10

ACS+在解决 CTSP 问题上要比其它的几种方法要强,下面来看在解决其它问题上的比较。选用 TSPLIB (<http://softlib.rice.edu/softlib/tsplib/>) 中 TSP 问题来对这几种改进的方法进行实验,参数选取与前面的一样。将结果和遗传算法、模拟退火算法进行比较。其中,遗传算法 (Genetic Algorithm, GA) 采用的是基于排名的选择 (ranking selection)、次序杂交和倒位变异的遗传算法<sup>[12]</sup>,相关参数为:种群数目为 50,杂交概率为 0.001,变异概率为 0.8。模拟退火<sup>[13]</sup>中的参数为:在某个温度下产生的变换个数为 100,温度按 1 进行递减。两种算法以最大演化代数 10000 作为停止条件。实验结果如表 2 所示。从表 2 综合起来看可以发现,ACS+最优,其次是 ACS、Ant-F、Ant-Q。这说明 ACS+具有良好的性能,在统计的过程中,除了 Gr48 外,均能找到最优的值(这里说算法找到了最优的值是指找到了到目前为止国际社会对问题求解的最好的结果),并且每次都能找到最优的值。虽然在 Gr48 中,系统经过了 2000

次迭代后,没有找到最优的值,但是它找到的结果要比其它方法所找到的结果要强,接近最优值。尽管有随机的因素的影响存在,但是这种算法总能找到最优的结果,这也就说明了它的先进之处;Ant-F 较 Ant-Q 要强,较 ACS、ACS+ 要弱,但其仿真简单,容易让人理解,也是一种值得借鉴的方法。同时,就旅行商问题而言,蚁群优化算法的收敛性比 GA、SA 要好得多,此外,在解决维数较低的旅行商问题,例如 Gr17、Gr24、Swiss42、Hk48

表 2 几种改进方法在不同问题上的实验结果

		Ant-Q	ACS	Ant-F	ACS+	GA	SA
GR17 (500)	最优值	2085.0	2085.0	2085.0	2085.0	2386.0	2496.0
	平均值	2090.47	2085.0	2089.4	2085.0	2598.87	2570.20
	均方差	10.3	0.0	4.88	0.0	81.95	57.94
	最快迭代次数	147	93	117	101	10000	10000
	平均迭代次数	306.4	136.73	357.07	153	10000	10000
GR21 (1000)	最优值	2707.0	2707.0	2707.0	2707.0	4325.0	4239.0
	平均值	2749.6	2707.0	2707.0	2707.0	4523.33	4453.10
	均方差	38.49	0.0	0.0	0.0	124.21	92.91
	最快迭代次数	351	116	160	107	10000	10000
	平均迭代次数	879.47	154.13	337.7	135.4	10000	10000
GR24 (1500)	最优值	1272.0	1272.0	1272.0	1272.0	2033.0	2105.0
	平均值	1275.6	1272.0	1277.2	1272.0	2169.47	2190.20
	均方差	2.94	0.0	2.04	0.0	67.0	45.89
	最快迭代次数	98	102	220	96	10000	10000
	平均迭代次数	770.87	298.3	937.87	369.07	10000	10000
SWISS42 (1500)	最优值	1281.0	1273.0	1279.0	1273.0	3227.0	3215.0
	平均值	1291.6	1273.0	1297.8	1273.0	3403.73	3386.4
	均方差	3.46	0.0	10.53	0.0	98.4	67.94
	最快迭代次数	2000.0	222	2000.0	147.0	10000	10000
	平均迭代次数	2000.0	350.73	2000.0	275.13	10000	10000
GR48 (2000)	最优值	5269.0	5150.0	5228.0	5058.0	14475.0	14316.0
	平均值	5307.8	5213.13	5303.73	5092.13	14957.3	14795.2
	均方差	23.28	35.93	43.01	35.02	328.04	307.01
	最快迭代次数	2000.0	2000.0	2000.0	2000.0	10000	10000
	平均迭代次数	2000.0	2000.0	2000.0	2000.0	10000	10000
HK48 (2000)	最优值	11470.0	11461.0	11461.0	11461.0	32727.0	31497.0
	平均值	11603.4	11461.0	11527.9	11461.0	34223.0	33222.7
	均方差	75.29	0.0	59.41	0.0	654.90	947.34
	最快迭代次数	2000.0	115	1023	98	10000	10000
	平均迭代次数	2000.0	323.93	1880.0	222.8	10000	10000

上,ACS+的收敛性达到了 100%,当然,这种收敛性与问题的复杂程度,各个城市的分布及算法本身的参数设置都有关,一种算法不可能放之四海皆准。在收敛速度上,从图 1 中的平均迭代次数和找到最优值的最快迭代次数两个方面考察,可以得出结论,那就是 ACS+的收敛速度明显要比其他算法收敛速度要快,其次是 ACS、Ant-F、Ant-Q。

## 5 结论

本文在用蚁群优化算法解决中国的旅行商问题的过程中,将几种改进的蚁群优化算法进行对比,提出两种改进的方法。其中 Ant-F 通过增大那些没有被访问到的边将被搜索的概率,增强系统的搜索能力,不致于使系统过早陷入局部最小值,此过程利用了系统的负反馈能力,使系统具有正负反馈的功能,仿真简单,容易理解;ACS+是在 ACS 的基础上改进而成的,因为 ACS 到了系统演化的后期,系统信息素的强度大部分已经挥发,导致系统信息素强度都比较小,难以区分它们,在搜索最优值方

面,又很难有起色,所以为了区别它们的信息素强度,使系统能够尽快找到最优解,提出了在演化的后期适当增大系统区分信息素对比强度的方法。和其它的几种改进的蚁群优化算法和遗传算法、模拟退火算法相比较,实验表明,ACS+是本文提及的算法中最优的一种,它能加快系统收敛,找到问题的最优值。虽然蚁群优化算法的研究在国内刚刚起步,但是,随着人们对它的了解不断深入,必将得到更广泛的应用。

### 参考文献:

- [1] 周培德. 求解货郎担问题的几何算法 [J]. 北京理工大学学报, 1995, 15(1): 97-99.
- [2] 杨忠, 鲍明, 张阿舟. 人机结合求解中国旅行商问题 [J]. 模式识别与人工智能, 1995, 8(4): 373-376.
- [3] 徐伯庆, 宣国荣, 柴佩琪. 中国旅行商问题的二叉树描述及其求解 [J]. 模式识别与人工智能, 2000, 13(2): 373-376.
- [4] 靳蕃等. 神经网络与神经计算机 [M]. 成都: 西南交通大学出版社, 1991, 94-104; 375-376.
- [5] 杨忠, 鲍明, 张阿舟. 求解中国旅行商问题的新结果 [J]. 数据采集与处理, 1993, 8(3): 177-184.
- [6] Dorigo M, Caro G Di. Ant colony optimization: a new meta-heuristic [A]. *Proc. 1999 Congress on Evolutionary Computation* [C]. 1999-07, 1470-1477.
- [7] Dorigo M, Maniezzo V, Colnari A. The ant system: optimization by a colony of cooperating agents [J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 1996, 26(1): 29-41.
- [8] Dorigo M, Caro G Di, Gambardella L M. Ant algorithms for discrete optimization [J]. *Artificial Life*, 1999, 5(2): 137-172.
- [9] Gambardella L M, Dorigo M. Ant-Q: a reinforcement learning approach to the traveling salesman problem [A]. *Proc. 12th International Conference on Machine Learning* [C]. Tahoe City, CA, 1995, 252-260.
- [10] Gambardella L M, Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies [A]. *Proc. IEEE International Conference on Evolutionary Computation* [C]. 1996, 622-627.
- [11] Stützle T, Hoos H. The MAX-MIN ant system and local search for the traveling salesman problem [A]. *Proc. IEEE International Conference on Evolutionary Computation* [C]. 1997, 309-314.
- [12] 潘正君, 康立山, 陈毓屏编著. 演化计算 [M]. 北京: 清华大学出版社, 广西科学技术出版社, 1998.
- [13] 康立山等. 非数值并行计算(第一册)——模拟退火算法 [M]. 北京: 科学出版社, 1998.

**作者简介:** 燕忠 (1973-), 男, 东南大学生物医学工程系博士生, 研究方向为生物计算、人工智能、模式识别等;  
袁春伟 (1955-), 男, 生物医学工程系教授, 博士生导师, 研究方向为仿生计算、人工智能、纳米材料等, 承担了国家自然科学基金重点项目、国防科技预研项目、国家自然科学基金面上项目等6项。

## Applying Ant Colony Optimizations to Chinese Traveling Salesman Problem

YAN Zhong, YUAN Chun-wei

( National Laboratory of Molecular & Biomolecule Electronics, Department of Biomedical Engineering,  
Southeast University, Nanjing 210096, China )

**Abstract:** The Chinese Traveling Salesman Problem (CTSP) is one of combinatorial optimization problems, and is a NP. Based on Ant Colony Optimizations (ACOs), two improved versions of ACOs are proposed to solve this problem. One is Ant-F, in which system's search probability is increased to avoid premature. Such system with negative and positive feedback can be simply simulated and easily understood. The other is ACS+, which can be used to speed up system's convergence at the last stage of evolution by means of magnifying the contrast between the intensity of pheromone laid on every edge. The empirical experiments indicate that performance of ACS+ is better when compared with other ACOs, GA(Genetic Algorithm) and Simulated Annealing.

**Key words:** ant system; ant colony optimization; traveling salesman problem