

基于蚁群算法的中国旅行商问题满意解

伍文城,肖 建

(西南交通大学电气工程学院,四川 成都 610031)

摘要: 蚁群算法是基于群体合作的一类仿生算法,适合于解困难的离散组合优化问题。本文对其做了适当的改进,以克服其求解速度过慢、容易出现停滞的缺陷,并将其用于解决中国旅行商问题,找到了目前已知的最好的解,同时指出了进一步提高蚁群算法效率还需解决的问题和方向。

关键词: 模拟进化算法; 蚁群算法; 旅行商问题; 组合优化

中图分类号: TP301.6 **文献标识码:** A

Satisfactory Solution of Chinese Travelling Salesman Problem Based on Ant Colony Algorithm

WU Wen-cheng, XIAO Jian

(Electrical School of Southwest Jiaotong University, Chengdu 610031, China)

Abstract: Ant colony algorithm is a novel simulated evolutionary algorithm based on group cooperation and can be applied to solve hard discrete combinatorial optimization problem. This paper gives a brief introduction on the ant colony algorithm and modifies it to solve the Chinese travelling salesman problem. Result shows that the performance of ant colony algorithm is improved and the present best solution is found. It also points out that the ant colony algorithm still has to be modified in many ways.

Keywords: simulated evolutionary algorithm; ant colony algorithm; travelling salesman problem; combinatorial optimization

0 引 言

旅行商问题可简单描述为:求一条通过全部 N 个城市一次且仅一次的最短旅行路线。可将 TSP 定义为网络 $G = [V, A, C]$, 其中: $V = \{(i, j) | i, j = 0, 1, \dots, N-1, i \neq j\}$ 为弧集, 表示旅行商可能经过的路段集合; $C = \{C_{ij} | (i, j) \in A\}$ 为费用矩阵, C_{ij} 表示旅行商经过对应弧 (i, j) 所花费的时间、距离或运输成本等。

中国旅行商问题 (Chinese Travelling Salesman Problem, CTSP) 是一个真实的地理问题, 由靳藩教授在文献[1]中提出。它可简单表述为:求一条从北京出发经过中国 31 个省会/直辖市最后又回到北京的最短回路。CTSP 的求解情况如下:靳藩在 1991 年的著作^[1]提到用神经网络方法求出的一个满意解为 $L_{\min} = 15904$ 公里, 在 2000 年的著作^[2]中提到用神经网络、遗传算法求出的满意解均为 $L_{\min} = 15404$ 公里, 王攀^[4]等用改进了的混合遗传算法求得的满意解也

为 $L_{\min} = 15404$ 公里。

TSP 的算法难度是 NP 完全难度的, 目前尚无很好的求解方法。虽然有枚举法、分支定界法、割平面法、动态规划法等精确算法^[3], 但需时甚长; 对于一个仅有 30 个城市的对称 TSP 问题, 使用枚举法求解, 需搜索 $29! / 2$ 条路径, 在每秒运算 1000 亿次的巨型计算机上求解也需费时 4×10^{17} 年。

模拟进化算法就是针对这一类难题而涌现的仿生算法。这一类算法有模拟退火算法、遗传算法以及本文采用的蚁群算法等。蚁群算法在求解离散的复杂的组合优化问题上有优异的表现, 在求解 TSP 问题上它甚至可以与目前最好的一些专用算法相媲美^[7,8]。因此, 我们用蚁群算法来求解中国旅行商问题, 这一问题已为其他算法所攻克, 可以以此来检验蚁群算法的效能。在求解过程中, 我们对蚁群算法进行了改进, 较好地解决了蚁群算法存在的过早收敛、计算速度较慢等两大缺点。

收稿日期:2002-02-05

作者简介:伍文城(1977-), 男, 广东郁南人, 西南交通大学电气工程学院硕士研究生, 研究方向:智能控制、计算机控制技术、集成电路设计等。

1 蚁群算法的基本思想

蚁群算法是由意大利学者 M. Dorigo, A. Colomi, V. Manizzo 等^[5,6]在 90 年代初提出的一类新的仿生算法。蚁群算法的特点是模拟自然界中蚂蚁的群体行为。科学家们发现,蚁群总是能够发现从蚁巢到食物源的最短路径。经研究发现,蚂蚁在行走过的路上留下一一种挥发性的激素,蚂蚁就是通过这种激素进行信息交流。蚂蚁趋向于走激素积累较多的路径。找到最短路径的蚂蚁总是最先返回巢穴,从而在路上留下了较多的激素。由于最短路径上积累了较多的激素,选择这条路径的蚂蚁就会越来越多,到最后所有的蚂蚁都会趋向于选择这条最短路径。基于蚂蚁这种行为而提出的蚁群算法具有群体合作、正反馈选择、并行计算等三大特点,并且可以根据需要为人工蚁加入前瞻、回溯等自然蚁所没有的特点。自然蚁的行为特点及它与人工蚁的不同之处见 Dorigo 等人的著作^[9]。

在使用蚁群算法求解现实问题时,先生成具有一定数量蚂蚁的蚁群,让每一只蚂蚁建立一个解或解的一部分,每只人工蚁从问题的初始状态出发,根据“激素”浓度来选择下一个要转移到的状态,直到建立一个解,每只蚂蚁根据所找到的解的好坏程度在所经过的状态上释放与解的质量成正比例的“激素”。之后,每只蚂蚁又开始新的求解过程,直到寻找到满意解。为避免停滞现象,引入了激素更新机制。

2 解决 TSP 的蚁群算法描述

用于解决 TSP 问题的蚁群算法原理如下:随机生成 m 个城市节点 ($1 \leq m \leq N$),将 m 只蚂蚁分别放在每一个城市上去,每只蚂蚁通过“状态转移规律”选择下一个要访问的城市,每只蚂蚁趋向于访问具有较高“激素”浓度的路径。当所有的蚂蚁完成了一次巡回后,即启动全局激素更新机制及激素挥发机制,而每只蚂蚁在它走过的路径的每一条弧上释放与它所找到的路径长度成反比的“激素”。之后,每只蚂蚁又开始新的巡游,直到满足停止条件为止。详细的蚁群算法描述可参考相关文献^[7,9]。

本文在求解 CTSP 过程中也对这一算法作了改进,下文将有论述。

2.1 状态转移规则

在城市 i 的第 k 只蚂蚁选择转移到城市 j 的概率计算方法为:

$$P_k(i, j) = \begin{cases} \frac{[\tau(i, j)] \cdot [\eta(i, j)]^\beta}{\sum_{u \in J_k(i)} [\tau(i, u)] \cdot [\eta(i, u)]^\beta} & \text{if } j \in J_k(i) \\ 0 & \text{others} \end{cases} \quad (1)$$

其中, $\tau(i, j)$ 表示边 (i, j) 的适应值,即“激素”; $\eta(i, j)$ 为距离 $C(i, j)$ 的倒数; $J_k(i)$ 为在城市 i 的蚂蚁 k 可以访问的城市集合; β 为一个参数,它表示激素与距离的相对重要程度。

在蚁群算法中,选择方式为:

$$j = \begin{cases} \arg \max_{u \in J_k(i)} \{[\tau(i, u)] \cdot [\eta(i, u)]^\beta\} & \text{if } q < q_0 \\ J & \text{others} \end{cases} \quad (2)$$

其中, q 为均匀分布在 $[0, 1]$ 上的一个随机变量, q_0 为在 $[0, 1]$ 上的参数,而 J 则是根据等式(1)计算出来的概率分布来进行选择,这种做法类似于遗传算法的“轮盘选择”。本文没有采用这种方式,而是采用当 $q > q_0$ 时使用随机选择的方式。我们认为这两种方式本质上是一致的,采用随机选择方式相当于降低了 q_0 值,从而加大了搜索空间。实验结果也表明这两种做法没有太大的差别,由于随机选择方式实现起来更简便、执行速度也略快、求出的解的质量也较高,本文最终采用了随机选择方式。

2.2 全局更新规则

在蚁群算法中只有生成了全局最优解的蚂蚁(即从一开始到现在构建了最短路径的蚂蚁)才有机会进行全局更新。

更新规则为:

$$\tau(i, j) \leftarrow (1 - \alpha) \cdot \tau(i, j) + \alpha \cdot \Delta \tau(i, j) \quad (3)$$

而

$$\Delta \tau(i, j) = \begin{cases} (L_{gb})^{-1} \\ 0 \end{cases} \quad (4)$$

α 为位于 $[0, 1]$ 上的“激素”挥发因子,而 L_{gb} 为到目前为止所找到全局最短路径长度,当然,也可以采用本次巡回最优的路径来进行更新。但实验表明,采用全局最短路径更新规则略好,因此本文采用这种方式。

2.3 局部更新规则

每只蚂蚁建立一个解的过程中也同时进行激素迹的更新,规则如下:

$$\tau(i, j) \leftarrow (1 - \rho) \cdot \tau(i, j) + \rho \cdot \Delta(i, j) \quad (5)$$

其中, ρ 为 $[0, 1]$ 间的参数, $\Delta \tau(i, j) = \tau_0$, τ_0 为激素迹的初始水平。

按激素的更新时刻分有在线更新、在线延迟更新、离线更新等三种方式。本文采用在线延迟更新方

式,即每只蚂蚁建立了一条回路以后再回去更新所走的边上的“激素”水平。

2.4 参数设置

采用 Dorigo 在文献[8]中提出的建议值: $\beta = 2$, $q_0 = 0.9$, $\alpha = \rho = 0.1$, $\tau_0 = (n \cdot L_{\min})^{-1}$, 而 L_{\min} 是用贪心法算出的巡回路径长度, n 为城市数目, 蚂蚁数目取为 $m = 10$ 。

2.5 路线优化方法

在使用蚁群算法解决 CTSP 过程中,我们发现蚁群算法容易出现“停滞”现象:当找到一个较好解以后,再增加循环次数也无法使解得到改善。因此本文引入路线改进法,即通过对边(弧)进行交换,在解的领域进行调整,每次调整可使可行解得到改进,直到解在领域内不能改进为止。现有线路改进法有 2-交换、3-交换、or-交换等。本文没有采用文献[7]所用的 3-交换法而采用了 2-交换法,因为 2-交换法实现起来较简单一些,而效果几乎一样。我们只对本次循环中产生的最好的解进行 2-交换,极大地节约了时间。

2-交换法如下:以 (i, j) 、 $(i+1, j+1)$ 代替 $(i, i+1)$ 、 $(j, j+1)$, 交换后线路中的路径 $(i+1, \dots, j)$ 被反向,如图 1 所示:

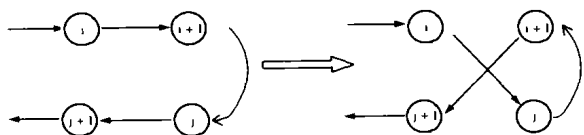


图 1 2-交换示意图

当交换后线路长度缩短,即满足以下条件时:

$$C_{i,j} + C_{i+1,j+1} < C_{i,i+1} + C_{j,j+1} \quad (6)$$

交换后可使可行解得到改进。

2.6 停止条件

可采用固定循环次数或当进化趋势不明显使停止迭代作为停止条件,为简便起见,这里采用固定循环次数的方法。

3 求解结果

根据上述算法,采用 Matlab6.0 为编程工具,在 CPU 为 PIII 733MHz,内存为 128MB,操作系统为 Windows Me 的计算机上求解。结果如下:

(1)基本蚁群算法求解结果。

本文采用基本蚁群算法求解求得最好结果为 $L_{\min} = 15810$ 公里,巡回路径为:

北京→呼和浩特→太原→石家庄→郑州→西安→银川→兰州→西宁→乌鲁木齐→拉萨→成都→昆

明→贵阳→南宁→海口→广州→长沙→南昌→武汉→合肥→福州→台北→杭州→上海→南京→济南→天津→沈阳→长春→哈尔滨→北京

此后,再怎样增加循环次数也不能使解得到改善:即使经过 100000 次循环、长达 12397 秒的搜索后,它仍然未能找到目前公认的最好的解!

(2)改进后的蚁群算法求解结果。

引入了随机选择、2-交换法后,我们经过多次实验发现,只需 100 次循环、仅 31 秒的时间就能发现 $L = 15516 \sim 15426$ 公里这样的满意解,且经过 10000 次循环长达 3198 秒搜索后得到一个最满意解为 $L_{\min} = 15404$ 公里,巡回路径为:

北京→呼和浩特→太原→石家庄→郑州→西安→银川→兰州→西宁→乌鲁木齐→拉萨→成都→昆明→贵阳→南宁→海口→广州→长沙→武汉→南昌→福州→台北→杭州→上海→南京→合肥→济南→天津→沈阳→长春→哈尔滨→北京

这和文献[2]求解的结果一致,而文献[4]给出的巡回路径中少了上海,估计是排版错误。这在另一侧面证实了 $L_{\min} = 15404$ 公里为 CTSP 最优解的猜测。

4 结束语

本文通过引入随机选择、2-交换法,使得蚁群算法的性能得到较大提高,在使用它求解中国旅行商问题中也取得让人满意的结果,能在 30 秒时间内发现长度为 15426 公里这样的满意解。但是蚁群算法还是容易陷入局部最优解:往往要耗费让人难以等待的时间才能找到目前已知的最好的解,有时还可能找不到。Dorigo 等^[7]指出,引入新的路线改进法,如 Lin-Kenighan 法,有可能进一步改善蚁群算法的性能,我们正在进行这一步工作。

蚁群算法的最优停止条件更是值得我们研究。据掌握的现有资料来看,这一方面的工作还没有专门的论述,现在所用的停止条件大多是带有尝试性质的。确定性的停止条件有助于我们减少尝试错误的次数。

蚁群算法要取得与其他仿生算法相比较的优势,解决这两个问题尤为关键。

参考文献:

- [1] 靳藩,范俊波,谭永东.神经网络与神经计算机[M].成都:西南交通大学出版社,1991.375~377.

(下转第 11 页)

表 1 简单多边形的计算结果

顶点数	150	300	450	600
梯形数	147	276	372	558
合并梯形数	12	9	15	24
优化后梯形数	135	247	357	534
初始化 CPU 时间(ms)	0.3	0.7	1.0	1.4
梯形运算 CPU 时间(ms)	3.5	9.2	15.6	28.2
优化 CPU 时间(ms)	0.9	0.8	0.8	1.0
总 CPU 时间(ms)	4.7	10.7	17.4	30.6

为了考察本算法的性能,在 CPU 为 AMD 雷鸟 750、内存 128MB 的计算机上进行了计算。编程语言为 Visual C++ 5.0,用人工生成的数据进行了运算。表 1、表 2 中的多边形是不带孔的,表 1 的多边形几乎是单调的,表 2 的多边形带有许多凹面。表中的时间为单次运算的时间,正如第 2 节所预计的,梯形化所费时间最多,时间的增加也不象 $O(n^2 \log_2 n)$ 那样快,实际上,时间大约是按 $O(n \log_2 n)$ 数量级增加的。表 2 的多边形远比表 1 的复杂,所以生成了更多的梯形,处理所费时间显著增加,可以并接的梯形数量几乎是线性增加的,同样所费的时间也线性增加,但是远较第 2 节中所说的最恶劣情况的时间少。

表 2 带有许多凹面的多边形的计算结果

顶点数	150	300	450	600
梯形数	441	891	1251	1743
合并梯形数	129	276	390	579
优化后梯形数	312	615	861	1164
初始化 CPU 时间(ms)	0.5	0.6	0.8	1.1
梯形运算 CPU 时间(ms)	5.6	13.1	21.2	36.4
优化 CPU 时间(ms)	1.2	2.5	3.3	4.9
总 CPU 时间(ms)	7.3	16.2	25.3	42.4

表 3 中多边形带有凹面和不同的孔数,所用的外环是相同的,都有 100 个顶点,每个内环都有 10 个顶点。由于内环的存在而造成外环内部的破碎情况,可清楚地反映在所进行的优化程度上。比较算法不同阶段的运算时间并不合理,因为它们与拓扑上的不同有关,但是总时间是可以比较的,所以表 3 中给出了总时间。由表可见,本算法可以有效地、经济地处理大量复杂的多边形,经优化处理后,可减少约 34% 的梯形个数。

表 3 带有许多凹面和孔的多边形的计算结果

孔数	0	1	2	5	10
顶点数	300	330	360	450	600
梯形数	978	1188	1386	2031	2898
合并梯形数	702	858	1008	1506	2112
优化后梯形数	276	330	378	525	786
总 CPU 时间(ms)	15.4	17.9	22.3	33.4	46.3

4 结束语

本文提出了用梯形来剖分多边形的通用算法,多边形可以是非单调的。在计算机上进行的计算表明,本算法可以有效、经济地处理大量复杂的多边形。对复杂的多边形,梯形化处理所费时间显著增加,但是远比最差情况的 $O(n^2 \log_2 n)$ 少。

参考文献:

- [1] 叶琳,邱龙辉.一种剖分平面多边形的通用算法描述[J].计算机与现代化,2002(7).
- [2] Narkhede A, Manocha D. Fast Polygon Triangulation Based on Seidel's Algorithm[A]. Paeth AW. Graphics Gems V[C]. Boston: Academic Press, 1995.

(上接第 8 页)

- [2] 靳藩.神经计算智能基础[M].成都:西南交通大学出版社,2000.300~308.
- [3] 马良.旅行推销员问题的算法综述[J].数学实践与认识,2000,30(2):156~165.
- [4] 王攀,商海燕,潘利群,徐承志.基于混合遗传算法的中国旅行商问题满意解[J].航空计算技术,2000,30(1):19~21.
- [5] M Dorigo, V Maniezzo and A Colomni. The Ant System: An Autocatalytic Optimizing Process[R]. Technical Report No. 91-016 Revised, Politecnico di Milano, Italy, 1991.
- [6] Marco Dorigo, etc. The Ant System: Optimization by a colony of

- cooperating agents[J]. IEEE Transactions on System, Man, and Cybernetics-part B, 1996, 26(1):29~41.
- [7] Macro Dorigo and Luca Maria Gambardella. Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem[J]. IEEE Transaction On Evolutionary Computation, 1997, 1(1):53~66.
- [8] Macro Dorigo and L M Gambardella. Ant Colonies for the Travelling Salesman Problem[J]. BioSystems, 1997, 43:73~81.
- [9] M Dorigo, G D Caro and L M Gambardella. Ant Algorithms for Discrete Optimization[J]. Artificial Life, 1999, 5(2):137~172.