文章编号:1000-6788(2004)08-0102-05

圆排列问题的蚁群模拟退火算法

高　尚[1,2]　杨静宇[1]，吴小俊[2]，刘同明[2]

(1. 南京理工大学计算机系，江苏 南京 210094；2. 江苏科技大学电子信息学院，江苏 镇江 212003)

摘要：首先把圆排列问题转化为旅行商问题，然后利用模拟退火算法求解此问题. 针对模拟退火算法对选择试验解比较敏感这一问题，文章提出六种找领域解算法. 算法的分析和测试表明，利用了城市间距离大小的信息的蚁群模拟退火算法Ⅱ是一种简单有效的算法.

关键词：圆排列问题；旅行商问题；模拟退火算法；蚁群算法

中图分类号：TP301.6　　**文献标识码**：A

Solving Circle Permutation Problem with Ant Colony -Simulated Annealing Algorithm

GAO Shang[1,2], YANG Jing-yu[1], WU Xiao-jun[2], LIU Tong-ming[2]

(1. Department of Computer, Nanjing University Science and Technology, Nanjing 210094, China; 2. School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

Abstract: Circle permutation problem is translated into traveling salesman problem (TSP) firstly, then the simulated annealing algorithm (SA) is used to solve the TSP. The quality of the annealing solution is very sensitive to the way in which the trial solutions are selected. In order to search neighborhood of the trial solution, six algorithms are put forward. By analysis and test, it is proved that Ant Colony-Simulated Annealing Algorithm Ⅰ, which makes use of the information of distance between cities, is a simple and effective algorith.

Key words: circle permutation problem; traveling salesman problem; simulated annealing algorithm; ant colony algorithm

1　引言

实际工程中经常涉及到工件切割问题，如把一个矩形钢板切割成半径不等的圆，尽可能节省材料，这就是圆排列问题. 所谓圆排列问题[1]是指给定 n 个大小不等的圆 $c_1, c_2, \cdots, c_n$，现要将这 n 个圆排进一个矩形框中，且要求与矩形的底边相切. 圆排列问题要求从 n 个圆的所有排列中找出有最小长度的圆排列. 文献[1]用回溯法解决此问题，其最坏情况接近于枚举法，时间复杂性为 $O((n+1)!)$，属于 NP-完全问题，目前没有有效的算法解此问题. 圆排列问题有很强的应用背景，如铺设半径大小不等的电缆管道、下水道等等，都可以转化为圆排列问题.

2　与旅行商问题等价

已知圆 c_i 的半径 r_i，$(i = 1, 2, \cdots, n)$，假如排列方式为 $i_1, i_2, \cdots, i_n$，则长度为

$$D = r_{i_1} + 2\sqrt{r_{i_1} r_{i_2}} + 2\sqrt{r_{i_2} r_{i_3}} + \cdots + 2\sqrt{r_{i_{n-1}} r_{i_n}} + r_{i_n} \tag{1}$$

收稿日期：2003-04-26

资助项目：江苏科技大学青年基金(Q2002313)

作者简介：高尚(1972－)，男，江苏镇江人，硕士，副教授，主要从事系统理论与优化等方面的研究，Email：sys_gao@163.net

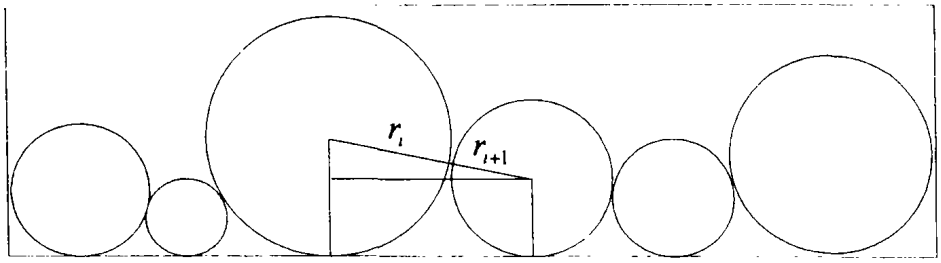

图 1　圆排列问题

圆排列问题要求所有排列中使长度找出最小. 所有的排列有 $n!$ 个，去掉对称的排列，如 1，2，…，$n-1$，n 与 n，$n-1$，…，2，1 对称，共有 $\frac{1}{2}n!$ 种排列.

旅行商问题(TSP)是指有 n 个城市，城市 i，j 之间的距离为 d_{ij}，一个旅行商从城市 1 出发到其他每个城市去一次且只去一次，最后回到城市 1，旅行商问题要求从 $2-n$ 个城市的所有排列中找出总路线最短的路线.

假设把 $1 \sim n$ 个圆分别放置在 $1 \sim n$ 个城市中，城市 i，j 之间的距离 $d_{ij}(i=1,2,\cdots,n;j=1,2,\cdots,n)$ 为

$$d_{ij} = 2\sqrt{r_i r_j} \tag{2}$$

再增加一个城市 0，它与城市 j 的距离 $d_{0j}(j=1,2,\cdots,n)$ 为

$$d_{0j} = r_j \tag{3}$$

因此圆排列问题与旅行商问题等价，一个旅行商从城市 0 出发到其他每个城市去一次且只去一次，最后回到城市 0，旅行商问题要求从 $1 \sim n$ 个城市的所有排列中找出总路线最短的路线. 所以解圆排列问题就变为解旅行商问题.

旅行商问题也是一个 NP-完全问题，目前求解旅行商问题的主要方法有动态规划方法[1]，分枝限界法[1]，模拟退火算法[2-5]，遗传算法[2,6]，启发式搜索法[2]，神经网络算法[7]，蚁群算法[8-10]等. 各种算法各有千秋，模拟退火算法最早思想由 Metropolis 在 1953 年提出，1983 年 Kirkpatrick 等成功地将退火思想引入组合优化领域. 模拟退火算法是局部搜索算法的扩展，理论上来说，它是一个全局最优算法. 如何在初始解附近找出一个“好的解”是一项关键技术，它直接影响算法的收敛速度. 本文推荐采用蚁群算法思想的模拟退火算法来解决此问题，其主要思想利用点的邻接关系，距离近的邻点以较大的概率选为下一个访问点.

3　模拟退火算法

模拟退火算法用于优化问题的出发点是基于物理中固体物质的退火过程与一般优化问题的相似性. 算法的基本思想是从一给定解开始的，从邻域中随机产生另一个解，接受准则允许目标函数在有限范围内变坏，它由一控制参数 t 决定，其作用类似于物理过程中的温度 T，对于控制参数 t 的每一取值，算法持续进行“产生新解—判断—接受或舍弃”的迭代过程，对应着固体在某一恒定温度下趋于热平衡的过程. 经过大量的解变换后，可以求得给定控制参数 t 值时优化问题的相对最优解. 然后减小控制参数 t 的值，重复执行上述迭代过程. 当控制参数逐渐减小并趋于零时，系统亦越来越趋于平衡状态，最后系统状态对应于优化问题的整体最优解，该过程也称冷却过程. 由于固体退火必须缓慢降温，才能使固体在每一温度下都达到热平衡，最终趋于平衡状态，因此，控制参数的值必须缓慢衰减，才能确保模拟退火算法最终趋于优化问题的整体最优解. 模拟退火算法要从邻域中随机产生另一个解，对于旅行商问题，它的邻域是指两条路径除局部有差别外，大多数路径相同.

解 TSP 问题的模拟退火算法的框架[5]：给定起、止"温度" T 、T_0和退火速度 α，初始一条路径 C_0；

While ($T > T_0$) do

在 C_0 的邻域内产生另一条路径 C_1；

计算两条路径所引起的目标函数(能量)值的变化 ΔE；若 $\Delta E \leq 0$，接受新值，否则若 $\exp(-\Delta E/T) > rand(0,1)$ ($rand(0,1)$ 表示 0 ～ 1 之间的随机数)，也接受新值，否则就拒绝；

确定新的参数值，若扰动被接受，则 $C_0 \leftarrow C_1$，否则 C_0 不变；

若接受新值，降温 $T \leftarrow \alpha T$，否则不降温；

End

4 几种算法的比较

模拟退火算法是依赖邻域结构的迭代方法，如何找领域的解直接影响收敛速度和最优解. 按照上面方法把圆排列问题转化为旅行商问题. 针对这个问题，在上面算法中，在 C_0 的邻域内产生另一条路径 C_1，这里提出 6 种算法. 为便于说明，假设本问题为 $n = 9$ 的圆排列问题.

算法 A 在第 1 ～ n 个访问的城市中随机地选取第 j_1 次和第 j_2 次访问的城市，在路径 C_0 中交换第 j_1 次和第 j_2 次访问的城市，其余不变，此时的路径为 C_1.

比如 $C_0 =$ (0 2 3 4 1 5 7 9 8 6)，$j_1 = 2$ (第 2 次访问的城市是城市 3)，$j_2 = 7$ (第 7 次访问的城市是城市 9)，则 $C_1 =$ (0 2 9 4 1 5 7 3 8 6).

算法 B 在第 1 ～ n 个访问的城市中随机地选取第 j_1 次访问的城市，在路径 C_0中交换第 j_1 次和第 $j_1 + 1$ 次访问的城市，其余不变，此时的路径为 C_1.

比如 $C_0 =$ (0 2 3 4 1 5 7 9 8 6)，$j_1 = 2$，则 $C_1 =$ (0 2 4 3 1 5 7 9 8 6)

算法 C 也称逆转算法，在第 1 ～ n 个访问的城市中随机地选取第 j_1 次和第 j_2 次访问的城市，在路径 C_0 中第 j_1 次到第 j_2 次访问的城市之间的子路径以反方向插入，其余不变，此时的路径为 C_1.

比如 $C_0 =$ (0 2 3 4 1 5 7 9 8 6)，$j_1 = 2$，$j_2 = 7$，则 $C_1 =$ (0 2 9 7 5 1 4 3 8 6)

算法 D 在第 1 ～ n 个访问的城市中随机地选取第 j_1 次和第 j_2 次访问的城市，假设 $j_1 < j_2$，在路径 C_0 中将第 j_1次访问的城市安排到第 j_2 次访问的城市之后，其余不变，此时的路径为 C_1.

比如 $C_0 =$ (0 2 3 4 1 5 7 9 8 6)，$j_1 = 2$，$j_2 = 7$，则 $C_1 =$ (0 2 4 1 5 7 9 3 8 6)

蚁群模拟退火算法 I 上面算法没有利用城市间距离大小的信息，蚁群模拟退火算法 I 将利用点的邻接关系，依据蚁群算法的思想，距离近的邻接点以较大的概率被选为下一个访问点，所以在局部调整时依据此思想.

设 $d(i,j)$ 表示城市 i 与城市 j 的距离，在 1 ～ n 的城市中随机地选取城市 i_1，离城市 i_1 最远的城市的距离为：$d_{\max} = \max_j d(i_1, j)$，为了排除下一个访问点为它自己，令 $d(i_1, i_1) = d_{\max}$，则下一个访问点为城市 j 的概率为：

$$p_j = \frac{d_{\max} - d(i_1, j)}{\sum_{k=1}^{n} (d_{\max} - d(i_1, k))} \tag{4}$$

假设以(1)式的概率选取的是城市 j_1，在路径 C_0 中将城市 j_1 安排到城市 i_1 之后，其余不变，此时的路径为 C_1. 比如 $C_0 =$ (0 2 3 4 1 5 7 9 8 6)，$i_1 = 3$，$j_1 = 7$，则

$$C_1 = (0\ 2\ 3\ 7\ 4\ 1\ 5\ 9\ 8\ 6)$$

蚁群模拟退火算法 II　在蚁群模拟退火算法 I 中是在 $1 \sim n$ 的城市中随机地选取城市 i_1，为了使路径总长度之和达到最小，优先解决薄弱环节，这里采用路径中相邻城市之间的距离大的两个城市以较大的概率被选取，在它们之间插入其他城市．用 $l(n)$ 数组记录路径 C_0 相邻城市之间的距离，具体数据如下：

$$l(k) = d[c(k), c(k+1)], k = 1, 2, \cdots, n-1$$

$$l(n) = d[c(n), c(1)]$$

选取城市 i 的概率为：

$$p_i = \frac{l(i)}{\sum_{k=1}^{n} l(k)} \tag{5}$$

按(2)式以概率选取的是城市 i_1，后面的步骤同蚁群模拟退火算法 I，按(1)式以概率选取的是城市 j_1，在路径 C_0 中将城市 j_1 安排到城市 i_1 之后，其余不变，此时的路径为 C_1．

5　算例分析

假设圆排列问题 n 取 30，50，100，$r_i = i$ ($i = 1, 2, \cdots, n$)．假设算法的参数相同，起始温度 $T = 100000$，终止温度 $T_0 = 1$，退火速度 $\alpha = 0.99$，各种算法各随机测试 100 次，结果如表 1－表 3 所示．

表 1　$n = 30$ 的结果比较

算法	结果比较			
	平均时间/s	平均值	最好解	最差解
算法 A	0.29	768.3933	758.8744	787.3225
算法 B	0.28	775.9391	764.7513	789.9046
算法 C	0.35	766.1576	756.4350	778.0075
算法 D	0.27	767.8782	756.8798	782.2558
蚁群模拟退火算法 I	0.49	767.7527	755.1965	776.5480
蚁群模拟退火算法 II	0.67	765.7780	750.7518	770.4598

表 2　$n = 50$ 的结果比较

算法	结果比较			
	平均时间/s	平均值	最好解	最差解
算法 A	0.63	2066.8	2049.1	2071.5
算法 B	0.58	2070.7	2064.2	2099.2
算法 C	0.71	2067.6	2050.5	2086.5
算法 D	0.58	2075.3	2044.6	2087.4
蚁群模拟退火算法 I	0.78	2047.4	2041.7	2054.3
蚁群模拟退火算法 II	0.98	2041.5	2037.5	2045.5

从表 1－表 3 可以看出，蚁群模拟退火算法 I 与蚁群模拟退火算法 II 都是采用蚁群算法思想的模拟退火算法，蚁群模拟退火算法 II 是最好的算法，其次是蚁群模拟退火算法 I 和算法 C，算法 C 采用逆转策略，可以使迭代过程突破局部最优圈而跳到另一个搜索空间．算法 A、算法 B 和算法 D 花费的时间少，说明它们很容易落入局部最优解，效果较差．随着规模 n 的增大，运行时间都增大．特别当 n 增大时，采用蚁群模拟退火算法 II 的效果更好，优点更显著．

表 3 $n = 100$ 的结果比较

算法	结果比较			
	平均时间/s	平均值	最好解	最差解
算法 A	0.71	8107.3	8099.4	8116.4
算法 B	0.69	8124.7	8105.2	8149.2
算法 C	0.74	8086.6	8073.4	8099.7
算法 D	0.67	8097.3	8096.7	8116.3
蚁群模拟退火算法 Ⅰ	0.82	8057.5	8042.3	8072.5
蚁群模拟退火算法 Ⅱ	1.05	8023.1	8019.8	8049.7

6 结束语

本文把园排列问题转化为 TSP 问题，TSP 问题是组合优化领域中的一个典型的问题，解决此问题有较大的现实意义. 对于其他类似问题，如印刷电路板的钻空路线方案，连锁店送货路线问题等都可简化为 TSP 问题. 本文讨论了 6 种算法，推荐采用蚁群模拟退火算法 Ⅱ，若把这几种算法混合在一起，或与遗传算法的思想融在一起，可能效果会更好，因此还有许多工作有待研究.

参考文献：

[1] 王晓东. 计算机算法设计与分析[M]. 北京：电子工业出版社，2001. 179－181.

[2] 刑文训，谢金星. 现代优化计算方法[M]. 北京：清华大学出版社，1999：40－45.

[3] 高国华，沈林成，常文森. 求解 TSP 问题的空间锐化模拟退火算法[J]. 自动化学报，1999，25(3)：425－428.

[4] KIRKPATRICK S，GELATT JR，VECCHI JR. Optimization by simulated annealing[J]. Science，1983，220：671－680.

[5] 康立山，谢云，尤矢勇，等. 模拟退火算法[M]. 北京：科学出版社，1994：150－151.

[6] 谢胜利，唐敏，董金祥. 求解 TSP 问题的一种改进的遗传算法[J]. 计算机工程与应用，2002，38(8)：58－60.

[7] 张立明. 人工神经网络的模型及其应用[M]. 上海：复旦大学出版社，1994：97－98.

[8] 张纪会，徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展，1999，36(10)：1240－1245.

[9] 马良，项培军. 蚂蚁算法在组合优化中的应用[J]. 管理科学学报，2001，4(2)：32－37.

[10] Colorni A，Dorigo M，Maniezzo V. An investigation of some properties of an ant algorithm[A]. Proc. of the Parallel Problem Solving from Nature Conference (PPSN'92)[C]. Brussels，Belgium：Elsevier Publishing，1992. 509－520