Journal of System Simulation

Volume 29 | Issue 2 Article 3

6-1-2020

Particle Swarm Ant Colony Optimization Algorithm for Solving Circle Permutation Problem

Xiaoping Xu
1. School of sciences, Xi'an University of Technology, Xi'an 710054, China; ;

Qiuqiu Zhu
1. School of sciences, Xi'an University of Technology, Xi'an 710054, China; ;

Wang Feng
2. School of Mathematics and Statistics Sciences, Xi'an Jiaotong University, Xi'an 710049, China;

Particle Swarm Ant Colony Optimization Algorithm for Solving Circle Permutation Problem

Abstract

Abstract: A particle swarm ant colony optimization algorithm of solving circle permutation problem was proposed. *Based on the basis of analyzing relationship between circular permutation problem and traveling salesman problem, circular permutation problem was translated into traveling salesman problem, and it was further transformed into an optimization problem. In order to further improve the performance of the algorithm, here, four kinds of new scheme were proposed to solve circular permutation problem based on particle swarm ant colony optimization algorithm.* In the simulation experiments, compared with the existing algorithm, the simulation results show that the proposed method is reasonable and feasible. Namely, the proposed four kinds of new algorithm have fast convergence speed and optimal performance.

Keywords

circle permutation problem, traveling salesman problem, combinatorial optimization, particle swarm optimization algorithm, ant colony algorithm

第 29 卷第 2 期 系统仿真学报© Vol. 29 No. 2
2017 年 2 月 Journal of System Simulation Feb., 2017

求解圆排列问题的粒子群蚁群优化算法

徐小平 [1]，朱秋秋 [1]，王峰 [2]
（1. 西安理工大学理学院，西安 710054；2. 西安交通大学数学与统计学院，西安 710049）

摘要：提出一种求解圆排列问题的粒子群蚁群优化算法。*分析了圆排列问题与旅行商问题的关系后，将圆排列问题转化为旅行商问题，并将其转化为一个优化问题。为了改善算法的性能，这里给出了利用粒子群蚁群优化算法来求解圆排列问题的四种新方案。*在数值仿真中，与已有算法进行了比较，实验结果验证了所给方法是合理的和可行的。也就是说，所提四种新算法收敛速度快，寻优性能优越。

关键词：圆排列问题；旅行商问题；组合优化；粒子群优化算法；蚁群算法
中图分类号：TP301 文献标识码：A 文章编号：1004-731X (2017) 02-0248-08
DOI: 10.16182/j.issn1004731x.joss.201702003

Particle Swarm Ant Colony Optimization Algorithm for Solving Circle Permutation Problem

Xu Xiaoping[1], *Zhu Qiuqiu*[1], *Wang Feng*[2]
(1. School of sciences, Xi'an University of Technology, Xi'an 710054, China;
2. School of Mathematics and Statistics Sciences, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: A particle swarm ant colony optimization algorithm of solving circle permutation problem was proposed. *Based on the basis of analyzing relationship between circular permutation problem and traveling salesman problem, circular permutation problem was translated into traveling salesman problem, and it was further transformed into an optimization problem. In order to further improve the performance of the algorithm, here, four kinds of new scheme were proposed to solve circular permutation problem based on particle swarm ant colony optimization algorithm.* In the simulation experiments, compared with the existing algorithm, the simulation results show that the proposed method is reasonable and feasible. Namely, the proposed four kinds of new algorithm have fast convergence speed and optimal performance.

Keywords: circle permutation problem; traveling salesman problem; combinatorial optimization; particle swarm optimization algorithm; ant colony algorithm

引言

在组合优化问题中，圆排列问题是一种最典型的 NP-hard 问题[1]。它不仅具有组合优化问题的典型特征，并且描述简单。因此，许多学者将圆排列问题作为优化算法研究的公共实例[2-3]。而且圆排列问题有很强的实际应用背景，比如，包装问题[4]、农机作业优化[5]、应急物资配送[6]等问题，都可以被转化为圆排列问题。可惜的是，目前还没有有效的算法来求解它。因此，对圆排列问题的研究具有重要的理论和现实意义。

蚁群算法是由意大利学者 Marco Dorigo[7]等人提出的。它是一种结合分布式计算，正反馈机制和

收稿日期：2015-03-25 修回日期：2015-10-24；
基金项目：国家自然科学基金(61273127)；陕西省教育厅专项科研计划(14JK1538)；西安理工大学科技创新计划项目(2016CX013)。
作者简介：徐小平(1973-)，男，陕西蓝田，博士，副教授，研究方向为进化算法，系统建模理论等。

贪婪式搜索的优化算法。目前，蚁群算法已经在图着色问题[8]，大规模集成电路设计[9]，通讯网络中路由问题以及负载平衡问题[10]和车辆调度问题[11-12]等领域被广泛地应用。

粒子群优化算法[13]是人类受到自然界生物活动规律的启发而进一步研究发展起来的一种群智能全局随机优化算法。该算法自诞生以来，由于其具有容易被人们理解和操作，收敛速度较快，设置和调整的参数较少等优点[14-15]，因此，在数字图像处理[16-17]，神经网络的训练[18]，无线通信技术[19]，观测站的部署[20]和控制器优化设计[21]等多个领域被成功应用。

针对蚁群算法在求解复杂问题时容易陷入局部最优以及出现早熟现象的缺点。本文提出了粒子群蚁群优化算法，并用来求解圆排列问题。文中在叙述了圆排列问题和旅行商问题的关系之后，将圆排列问题转化为旅行商问题。接着，利用粒子群蚁群优化算法对其进行求解。最后，通过仿真实验结果说明了所给方法是有效的。

1 问题描述

圆排列问题是说：给定 n 个大小不等的圆 $c_1, c_2, \cdots, c_n$，现将这 n 个圆排进到一个矩形框中，要求与矩形的底边相切，并且要求在所有排列中找出具有最小长度的路线。显然，所有的排列有 $n!$ 个，去掉对称的排列(如 $1,2,\cdots,n-1,n$ 与 $n,n-1,\cdots,2,1$ 对称)后，即为一半。这里首先将圆排列问题转化为旅行商问题，然后用粒子群蚁群优化算法对其进行有效求解。

旅行商问题具体是指单一旅行者由起点城市出发，不重复地走完其余城市并回到原出发点，在所有可能的路径中，求出路径长度最短的一条。

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

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

假设把 1~n 个圆分别放置在 1~n 个城市中，城市 i 和城市 j 之间的距离 $d_{ij}(i=1,2,\cdots,n; j=1,2,\cdots,n)$ 为 $d_{ij} = 2\sqrt{r_i r_j}$。求得最短的路径长度后，再加上最初和最终的城市的半径，即求得最终要求的最短路径长度。因此，求解圆排列问题与求解旅行商问题是等价的[22]。也就是说，再增加一个城市 0，它与城市 j 的距离 $d_{0j}(j=1,2,\cdots,n)$ 为 $d_{0j} = r_j$。即一个旅行商从城市 0 出发到其它每个城市去一次且只去一次，最后回到城市 0。而旅行商问题要求从 1~n 个城市的所有排列中找出总路线最短的路线。因此，求解圆排列问题就可以转化为求解旅行商问题。

2 基本蚁群算法求解圆排列问题

利用基本蚁群算法求解圆排列问题时，算法的相关参数如下：

m：蚁群中蚂蚁的数量；

$b_i(t)$：t 时刻位于第 i 个圆的蚂蚁个数，这样就有 $m = \sum_{i=1}^{n} b_i(t)$；

d_{ij}：两个圆 i 和 j 之间的距离；

η_{ij}：边 i 和边 j 能见度，反映圆 i 转移到圆 j 启发程度；

τ_{ij}：边(i, j)上的信息素轨迹强度；

$\Delta\tau_{ij}=k$：蚂蚁 k 在边(i, j)留下单位长度轨迹信息素；

p_{ij}^k：位于圆 i 的蚂蚁 k 选择移动到圆 j 的概率；

ρ：信息素残留系数；

α：信息启发式因子；

β：期望启发式因子。

利用基本蚁群算法求解圆排列时，算法步骤如下：

Step 1. 初始化，设置时间 t=0 和循环次数 N_c=0，将 m 个蚂蚁置于 n 个圆上，每条路径 $v_{id} \in [-v_{\max}, v_{\max}]$$(i, j)$的初始化信息素 $\tau_{ij}(t) = c$，初始时刻 $\Delta\tau_{ij}(0) = 0$；

Step 2. 设置蚂蚁的禁忌表索引号 s=1，对 $k = 1,2,\cdots,m$，将 k 个蚂蚁的起始圆的编号放入禁忌表中。

DOI: 10.16182/j.issn1004731x.joss.201702003

第 29 卷第 2 期 系统仿真学报 Vol. 29 No. 2
2017 年 2 月 Journal of System Simulation Feb., 2017

Step 3. 循环执行以下步骤，直到禁忌表满足：

① $s=s+1$，

② 对 $k=1,2,\cdots,m$，以概率 $p_{ij}^k(t)$ 选择下一个圆 j，其公式为

$$P_{ij}^k(t)=\frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{s\in allowed_k}\tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)}\ ,j\in allowed_k \quad (2)$$

从而将蚂蚁 k 移到圆 j，将其编号放到禁忌表中；

Step 4. 对 $k=1,2,\cdots,m$，计算蚂蚁 k 所走的长度，记录当前找到的最短的路径的圆，按公式 $\Delta\tau_{ij}(t,t+1)=\sum\limits_{k=1}^{m}\Delta\tau_{ij}^k(t,t+1)$ 计算每只蚂蚁信息素增量；

Step 5. 对每一条路径$(i,\ j)$根据公式 $\tau_{ij}(t+1)=\rho.\tau_{ij}(t)+\Delta\tau_{ij}(t,t+1)$ 更新路径上的信息素。设置 $t=t+n$，$N_c=N_c+1$，对于每条路径(i,j)，设置 $\tau_{ij}=0$；

Step 6. 若循环次数 $N_c\geqslant N_{c_{\max}}$，则循环结束，并输出结果，否则，清空禁忌表，并且转到 Step 2。

3 粒子群蚁群优化算法求解圆排列问题

由 Eberhart 博士和 Kennedy 博士于 1995 年提出的粒子群优化算法是一种群体智能优化算法[13]，其基本思想源于对鸟群的行为模拟，通过群体中粒子之间的合作与竞争 而产生的群体智能来指导优化搜索。粒子群优化算法用随机解初始化一群随机粒子，通过迭代找到最优解，在每一次迭代中，每个粒子通过跟踪个体极值和全局极值来更新自己。同时，每个粒子不断改变其在解空间中的速度，以决定个体的走向及飞行距离，并尽可能朝个体极值和全局极值所指向的区域“飞”去来得到最优解。

设 D 维搜索空间中，有 M 个粒子，其中第 i 个粒子的位置是 $X_i=(x_{i1},x_{i2},\cdots,x_{iD})$，其速度为 $V_i=(v_{i1},v_{i2},\cdots,v_{iD})$，$i=1,2,\cdots,M$。记第 i 个粒子搜索到的最优位置为 $P_i=(p_{i1},p_{i2},\cdots,p_{iD})$，整个粒子群搜索到的最优位置为 $P_g=(p_{g1},p_{g2},\cdots,p_{gD})$。粒子状态更新公式如下：

$$V_{id}(t+1)=$$
$$V_{id}(t)+c_1r_1(p_{id}-x_{id}(t))+c_2r_2(p_{gd}-x_{id}(t)) \quad (3)$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \quad (4)$$

其中，$i=1,2,\cdots,M$；$d=1,2,\cdots,D$；学习因子 c_1 和 c_2 是非负常数；r_1 和 r_2 是相互独立的随机数，服从(0, 1)上的均匀分布；$v_{id}\in[-v_{\max},v_{\max}]$，$v_{\max}$ 是在之前设定的最大速率，t 为当前迭代次数。

蚁群算法利用正反馈原理和某种启发式算法的有机结合来进行寻优，但在某些复杂问题中，该算法容易出现过早收敛，陷入局部最优解的缺点。而基于粒子群优化算法的寻优能力主要依赖于粒子之间的相互作用和相互影响，但粒子自身没有变异能力。在这里，如果对粒子进行交叉操作，那么将会产生新的个体。如果对粒子进一步进行变异操作，那么将会产生更多的新个体。这样对粒子进行交叉和变异操作后，二者相互配合，能够更好的完成对搜索空间的全局搜索和局部搜索。从而当粒子陷入局部极值时，它可以借助其它粒子来跳出局部极值点。因此，将蚁群算法和粒子群算法混合后，蚂蚁就具有了“粒子”的特性，按照蚁群算法，完成一次遍历后，再让蚂蚁根据局部最优解和全局最优解来进行调整，从而会得到更有优的解。

下面给出利用粒子群蚁群优化算法来求解圆排列问题的步骤：

Step 1. 初始化，产生大量排列顺序，从中选择出较优的顺序，使这些顺序所在线路留下信息素，将蚂蚁置于 n 个圆上；

Step 2. 根据当前位置计算适应度值，即各线路半径的长度，设置当前适应度值为个体极值，当前位置为个体极值位置，根据各个粒子的个体极值，找出全局极值和全局极值位置；

Step 3. 将各个蚂蚁初始出发点置于当前解集中，对每只蚂蚁，按照概率 p_{ij}^k 移至下一圆 j，将圆 j 置于当前解集；

Step 4. 对每只蚂蚁进行交叉和变异操作，找出全局极值和全局极值位置。

在对蚂蚁进行交叉和变异操作，找出全局极值和全局极值位置的过程中，本文提出如下四种交叉和变异的方法：

算法Ⅰ：首先，随机选择出路线 C1 和 C2，然后，将 C1 和 C2 进行相互交叉和变异。例如：当 n=10 时，随机选择出 C1=[10 1 9 4 2 7 5 8 3 6]，C2=[7 3 9 2 1 10 6 8 4 5]后，首先，对其进行交叉后，结果为 C01=[10 4 5 1 9 2 7 8 3 6]，然后，在交叉的基础上再进行变异，结果为 C11=[10 4 5 1 9 2 6 8 3 7]。

算法Ⅱ：首先，随机选择出路线 C1 和 C2，接着，从 C2 中随机选出其局部作为路线 C3，然后，将 C3 和 C1 进行相互交叉和变异。例如：C1 和 C2 如上算法Ⅰ所示，从 C2 中选出 C3=[7 3 9]后，对 C3 和 C1 交叉后的结果为 C02=[7 3 9 10 1 4 2 5 8 6]，对其进行再变异后的结果为 C22=[7 3 910 1 4 5 2 8 6]。

算法Ⅲ：首先，随机选择出路线 C1 和 C2，接着，从 C2 中随机选出其局部作为路线 C4，然后，随机产生一个序号，最后，对 C4 和 C1 进行相互交叉和变异。例如：C1 和 C2 如上算法Ⅰ所示，从 C2 选出 C4=[1 10 6 8 4]后，对 C4 和 C1 交叉后的结果为 C03= [1 10 6 8 4 9 2 7 5 3]，进一步再变异后的结果为 C33=[5 7 2 9 4 8 6 10 1 3]。

算法Ⅳ：首先，随机选择出路线 C1 和 C2，接着，从 C2 中随机选出其局部作为路线 C5，然后，采用 C1 中再次出现的重复的个体作为序号后，再从开始到重复个体的位置，最后，对 C5 和 C1 进行相互交叉和变异。例如：C1 和 C2 如上算法Ⅰ所示，从 C2 选出 C5=[8 4 5]后，对 C5 和 C1 相互交叉后的结果为 C04=[10 1 9 2 7 8 4 5 3 6]，再对其进行变异后的结果为 C44=[10 1 9 2 7 8 5 3 6 4]。

Step 5. 计算各蚂蚁的路径的长度，记录当前的最好解；

Step 6. 按更新方程修改轨迹强度；

Step 7. 对各边弧，置路径上信息素增量为零，迭代次数增加 1；

Step 8. 若迭代次数小于预定的迭代次数且无退化行为(即找到的都是相同的解)，则转至 Step 2；否则，终止程序，输出目前的最优解。

图 1 给出所给改进算法求解圆排列问题过程的结构框图。

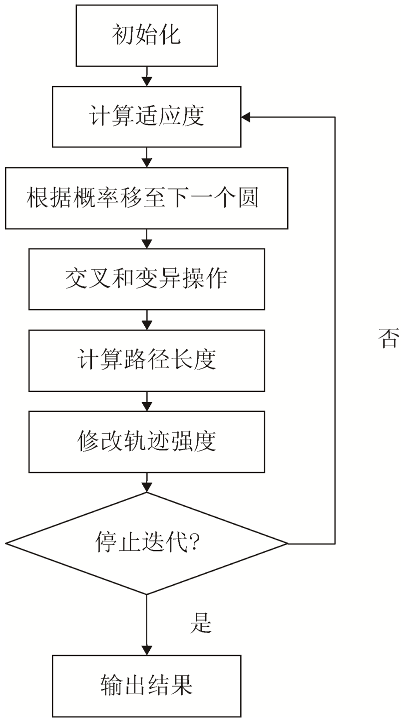

图 1 结构框图
Fig. 1 Structured flowchart

4 仿真实验

在仿真实验中，文中所给四种算法的参数设置如下：蚂蚁的数量和粒子的数量均为 m=30，信息素残留系数ρ=0.9，信息启发式因子α=1.5，期望启发式因子β=2，学习因子 c_1=c_2=0.7，最大速度 v_{max}=2 和迭代次数为 100。

首先，给出利用文中所给的四种算法、基本蚁群算法(参数设置：蚂蚁数量 m=30，信息素残留系数ρ=0.9，信息启发式因子α=1.5，期望启发式因子β=2 和迭代次数为 100)、遗传算法[6](参数设置：种群数量为 30，交叉概率为 0.9，变异概率为 0.05 和迭代次数为 100)，以及粒子群算法[13](参数设置：粒子数量 m=30，学习因子 c_1=c_2=0.7，最大速度 v_{max}=2 和迭代次数为 100)分别对旅行商问题

https://dc-china-simulation.researchcommons.org/journal/vol29/iss2/3
DOI: 10.16182/j.issn1004731x.joss.201702003

第 29 卷第 2 期　　　　系统仿真学报　　　　Vol. 29 No. 2
2017 年 2 月　　　　Journal of System Simulation　　　　Feb., 2017

Oliver30 进行求解，结果罗列在表 1 中。

表 1　求解 Oliver30 的结果
Tab. 1　Results of solving Oliver30

算法	Oliver30	
	最优路径	最差路径
算法 I	423.821 5	430.556 7
算法 II	423.675 8	429.854 2
算法 III	425.613 9	430.717 0
算法 IV	426.554 3	430.812 5
基本蚁群算法	427.649 1	431.820 1
遗传算法	439.359 5	472.474 6
粒子群算法	427.820 1	455.986 2

从表 1 可以看出，本文所提 4 种算法在求解旅行商问题的公共数据集上是有效的。

接着，考虑当圆排列问题 n 取 30，50 和 100，$r_i = i\ (i = 1, 2, \cdots, n)$ 的情况。利用文中所给四种方案对上述问题分别进行求解。

当 n 为 30 时，随机给定一个初始的位置，利用文中所给算法 I 进行一次求解，得到的圆的半径为 27 3 29 1 30 2 28 4 26 6 15 17 13 19 11 21 9 23 7 25 5 24 8 22 10 20 12 18 14 16，如图 2 所示，此时，求得最优的总距离为 756.396 5；利用文中所给算法 II 进行一次求解，得到的圆的半径为 24 8 22 10 20 18 12 14 16 15 17 13 19 11 9 23 7 25 5 27 3 29 1 30 2 28 4 26 6 21，如图 3 所示，此时，求得最优的总距离为 753.330 0；利用文中所给算法III进行一次求解，得到的圆的半径为 16 14 18 12 15 17 13 19 11 21 9 23 4 26 3 29 7 25 5 24 8 22 10 20 6 27 1 30 2 28，如图 4 所示，此时，求得最优的总距离为 757.077 8；利用文中所给算法Ⅳ进行一次求解后，得到的圆的半径为 17 13 19 11 21 9 23 27 3 29 1 30 2 28 4 26 6 15 7 25 5 24 8 22 10 20 12 18 14 16，如图 5 所示此时，求得最优的总距离为 759.415 6。并且图 6 分别给出了四种算法的相应求解过程图。

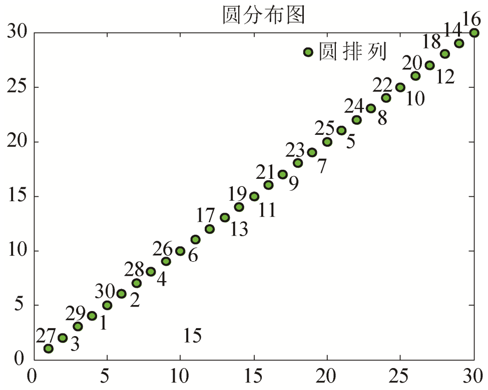

图 2　算法 I 的求解结果图
Fig. 2　Solving result diagram of algorithm Ⅰ

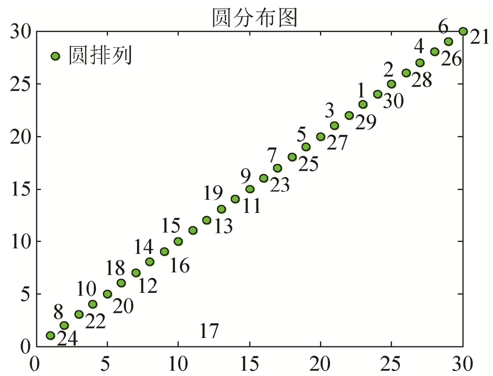

图 3　算法 II 的求解结果图
Fig. 3　Solving result diagram of algorithm Ⅱ

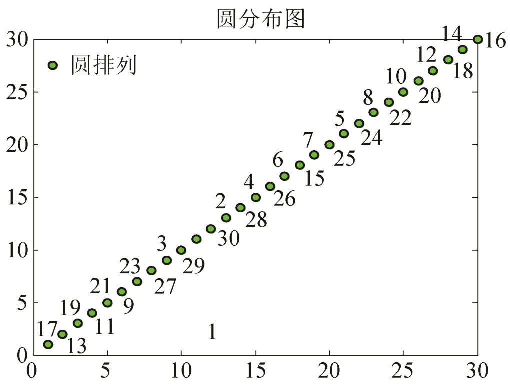

图 4　算法III的求解结果图
Fig. 4　Solving result diagram of algorithm III

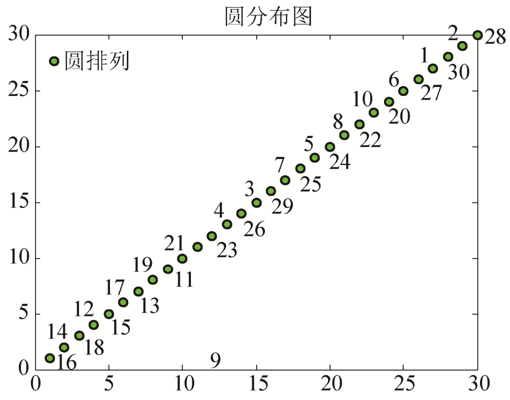

图 5　算法Ⅳ的求解结果图
Fig. 5　Solving result diagram of algorithm Ⅳ

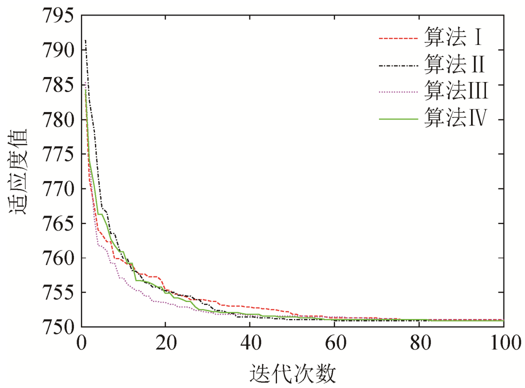

图 6　n 为 30 时的求解过程图
Fig. 6　Solving process diagram for n=30

当 n 为 50 和 100 时，同样分别随机给定一个初始的位置，利用文中所给的 4 种算法分别进行类似于 n 为 30 的求解过程分别进行一次求解，为了避免文字叙述过于繁琐，具体求解结果不在这里一一列出，这里只利用图 7 和图 8 分别给出了利用所给的 4 种算法对它们进行一次求解的相应过程图。

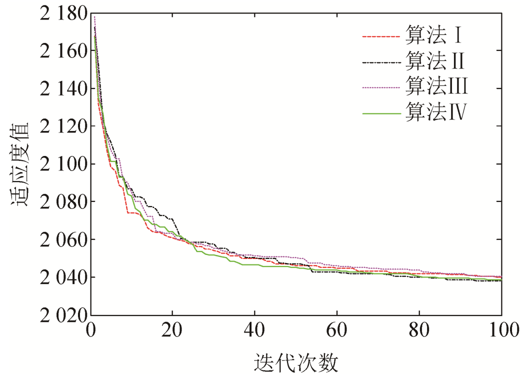

图 7　n 为 50 时的求解过程图
Fig. 7　Solving process diagram for n=50

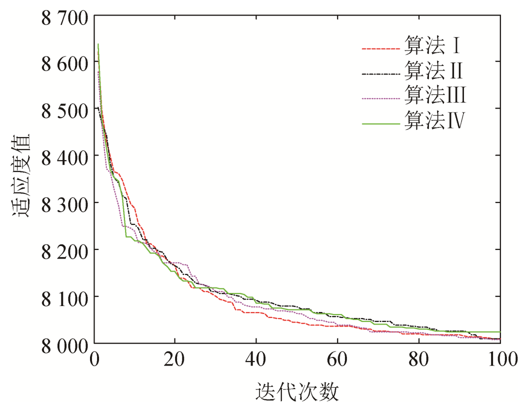

图 8　n 为 100 时的求解过程图
Fig. 8　Solving process diagram for n=100

对 n 取 30，50 和 100 时，利用本文所给的 4 种方法分别进行 100 次求解后，将其求解结果的平均值，最好解，最差解和所用平均时间分别罗列在表 2~4 中。为了进一步说明本文所给方法的可行性，再分别利用基本蚁群算法(参数设置：蚂蚁数量 m=30，信息素残留系数 ρ=0.9，信息启发式因子 α=1.5，期望启发式因子 β=2 和迭代次数为 100)、文献[22]中的蚁群模拟退火算法(参数设置：蚂蚁数量为 30，信息素残留系数为 0.9，信息启发式因子为 1.5，期望启发式因子为 2，起始温度为 100 000，终止温度为 1，退火速度为 0.99 和迭代次数为 100)，以及文献[23]中的改进蚁群算法(参数设置：蚂蚁数量为 30，信息素残留系数为 0.9，信息启发式因子为 1.5，期望启发式因子为 2，信息素强度为 1 和迭代次数为 100)分别对 n 取 30，50 和 100 进行 100 次求解，其相应结果的平均值，最好解，最差解和平均所用时间也分别罗列在表 2~4 中。

表 2　n=30 的相应求解结果
Tab. 2　Corresponding solving results of n=30

算法	结果比较			
	平均时间/s	平均值	最好解	最差解
算法 I	0.47	760.82	750.75	765.34
算法 II	0.43	757.32	750.75	762.5
算法 III	0.49	759.24	750.75	763.48
算法 IV	0.46	758.36	750.75	764.91
基本算法	0.75	787.42	784.25	790.48
文献[22]	0.68	765.77	750.75	770.459
文献[23]	0.546	787.469	784.417	792.951

表 3　n=50 的相应求解结果
Tab. 3　Corresponding solving results of n=50

算法	结果比较			
	平均时间/s	平均值	最好解	最差解
算法 I	0.77	2 047.8	2 037.5	2 055.2
算法 II	0.72	2 039.7	2 037.5	2 043.2
算法 III	0.80	2 041.8	2 037.5	2 045.7
算法 IV	0.86	2 042.5	2 037.5	2 050.6
基本算法	1.03	2 220.57	2 210.96	2 239.4
文献[22]	0.98	2 041.5	2 037.5	2 045.5
文献[23]	0.981	2 224.82	2 218.76	2 238.61

表 4 n=100 的相应求解结果

Tab. 4 Corresponding solving results of n=100

算法	结果比较			
	平均时间/s	平均值	最好解	最差解
算法 I	0.92	8 024.6	8 004.18	8 042.6
算法 II	0.87	8 020.5	8 004.18	8 040.9
算法 III	0.94	8 023.2	8 004.18	8 043.7
算法 IV	0.96	8 022.1	8 004.18	8 045.3
基本算法	2.478	8 843.6	8 824.7	8 856.3
文献[22]	1.05	8 023.1	8 019.8	8 049.7
文献[23]	2.273	8 843.46	8 821.88	8 867.70

从以上仿真结果可以看出，文中所给的利用粒子群蚁群优化算法求解圆排列问题的四种方法是有效的。

5 结论

针对圆排列的求解问题，文中给出了基于粒子群蚁群优化算法的 4 个新方法。最后，利用仿真实验结果说明了所给方法是合理的和有效的。

参考文献：

[1] Applegate D L, Bixby R E, Chvatal V, et al. The Traveling Salesman Problem: A Computational Study [M]. Princeton, USA: Princeton University Press, 2011.

[2] 麻存瑞，马昌喜. 不确定旅行商问题的鲁棒模型与算法 [J]. 计算机应用，2014, 34(7): 2090-2092. (Ma Cunrui, Ma Changxi. Robust Model and Algorithms for Uncertain Traveling Salesman Problem [J]. Journal of Computer Applications, 2014, 34(7): 2090-2092.)

[3] 王庆，刘学鹏. 基于流水算法的旅行商问题求解 [J]. 预测，2014, 33(1): 65-69. (Wang Qing, Liu Xuepeng. Solution for Travelling Salesman Problem Based on Flowing Water Algorithm [J]. Forecasting, 2014, 33(1): 65-69.)

[4] 杨金勇，宋海洲. 圆排列包装问题最优解解析 [J]. 华侨大学学报，2013, 34(2): 221-224. (Yang Jinyong, Song Haipeng. Solving Circle Permutation Packing Problem [J]. Journal of Huaqiao University, 2013, 34(2): 221-224.)

[5] 杨巍，刘占良. 农机作业路径优化的研究-基于旅行商问题新算法 [J]. 农机化研究，2014, 36(6): 54-57. (Yang Wei, Liu Zhanliang. Research on the Optimization of Agricultural Machinery Operation Path-Based on Traveling Salesman Problem New Algorithm [J], Journal of Agricultural Mechanization Research, 2014, 36(6): 54-57.)

[6] 刘明，张培勇. 求解多旅行商问题的新混合遗传算法-以应急物资为例 [J]. 系统管理学报，2014, 23(2): 247-254. (Liu Ming, Zhang Peiyong. A New Hybrid Genetic Algorithm for Multiple Traveling Salesman Problem-Taking Emergency Supplies as an Example [J]. Journal of Systems Management, 2014, 23(2): 247-254.)

[7] Costa D, Hertz A. Ants can colour graphs [J]. Journal of the Operational Research Society (S0160-5682), 1997, 48(3): 295-305.

[8] Kuntz P, Layzell P, Snyers D. A Colony of Ant-like Agents for Partitioning in VLSI Technology [C]// Proceedings of the 4th European Conference on Artificial Life. Boston, USA, 1997. USA: IEEE, 1997: 417-424.

[9] Schoonderwoerd R, Holland O, Bruten J. Ant like Agents for Loadbalancing in Telecommunications Networks [C]// Proceedings of Agemts'97 Marinadel del Rey, Washington, USA, USA: ACM, 1997: 209-216.

[10] Di Caro G, Dorigo M. Mobile Agents for Adaptive Routing [C]// Proceedings of the 31st Hawaii International Conference on System. Los Alamitos, USA, 1998. USA: IEEE, 1998: 74-83.

[11] Bullnheimer B, Hartl R F, Strauss C. Applying the Ant System to the Vehicle Routing Problem [J]. Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization (S0092-2102), 1998, 89(8): 109-120.

[12] Colorni A, Dorigo M, Maniezzo V, et al. Ant System for Job Shop Scheduling [J]. Belgian of Operations Research Statistics and Computer Science (S1098-7371), 1994, 34(1): 39- 53.

[13] Kennedy J F, Eberhart R C. Particle Swarm Optimization [C]// Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, 1995. USA: IEEE, 1995: 1942-1948.

[14] 纪震，廖惠连，吴青华. 粒子群优化算法及应用 [M]. 北京：科学出版社, 2009. (Ji Zheng, Liao Huilian, Wu Qinghua. Particle Swarm Optimization Algorithm and Its Application [M]. Beijing, China: Science Press, 2009.)

[15] Yan Z, Deng C, Zhou J, et al. A Novel Two-subpopulation Particle Swam Optimization [C]// Proceedings of 10th Intelligent Control and Automation, Beijing, China, 2012. USA: IEEE, 2012: 4113-4117.

[16] Muruganandham A, Wahida Banu Dr S D. Adaptive Fractal Image Compression Using PSO [J]. Procedia Computer Science (S1877-0509), 2010, 2(2): 338-344.

（下转第 263 页）