利用粒子群优化算法求解圆排列问题①

徐小平 [1], 朱秋秋 [1], 邰会强 [2]

[1](西安理工大学 理学院, 西安 710054)
[2](西安理工大学 机械与精密仪器工程学院, 西安 710048)

摘　要: 针对已有算法搜索时间较长, 且易于过早地收敛于非最优解的缺陷, 利用粒子群优化算法给出了圆排列问题的求解方法. 首先, 在分析了圆排列问题与旅行商问题关系的基础上, 将圆排列问题转化为旅行商问题, 从而得到一个相应的组合优化问题. 然后, 利用粒子群优化算法进行了求解. 接着, 为了进一步提高算法的精度, 文中给出了一种利用混合粒子群优化算法的方案. 最后, 在仿真实验中, 与已有算法进行了比较, 实验结果表明, 文中所给方法是有效的.

关键词: 圆排列问题; 组合优化; 粒子群优化算法; 进化算法

Solving Circle Permutation Problem by Particle Swarm Optimization Algorithm

XU Xiao-Ping[1], ZHU Qiu-Qiu[1], TAI Hui-Qiang[2]

[1](School of Sciences, Xi'an University of Technology, Xi'an 710054, China)
[2](School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an 710048, China)

Abstract: In view of the existing algorithms are of long searching time and easily to prematurely converge to the optimal solution, this paper proposes a method for solving circle permutation problem using particle swarm optimization algorithm. First of all, based on the analyzing of relationship between circular permutation problem and the traveling salesman problem, circular permutation problem is translated into traveling salesman problem and accordingly, a corresponding combinatorial optimization problem is obtained. Then, the problem is solved by particle swarm optimization algorithm. Thirdly, in order to further improve the precision of the algorithm, this paper proposes a scheme based on a hybrid particle swarm optimization algorithm. Finally, in the simulation experiments, compared with the existing algorithm, the simulation results show that the proposed method is effective.

Key words: circle permutation problem; combinatorial optimization; particle swarm optimization algorithm; evolutionary algorithm

组合优化问题的目标是从可行解集中求出目标函数的最优解. 而圆排列问题是组合优化中一种最典型的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]等等问题, 都可以被转化为圆排列问题. 因此, 对圆排列问题的研究具有重要的理论和现实意义.

粒子群优化算法[7]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. 它自诞生以来, 由于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等优点[8,9], 从而在短期内得到了很大发展, 目前该算法已经在数字图像处理[10,11]、神经网络的训练[12]、无线通信技术[13]、观测站的部署[14]、控制器优化设计[15]等多个领域得到了成功应用.

本文利用粒子群优化算法对圆排列问题进行求解.

① 基金项目:国家自然科学基金(61273127);陕西省自然科学基础研究计划(2014JM8325);陕西省教育厅科研计划(14JK1538)
收稿时间:2015-05-21;收到修改稿时间:2015-06-23

© 中国科学院软件研究所 http://www.c-s-a.org.cn

首先, 叙述了圆排列问题和旅行商问题之间的关系. 接着, 将圆排列问题转化为旅行商问题. 然后, 利用粒子群优化算法对其进行求解, 并且为了提高算法的精度, 给出了一种混合粒子群优化算法的求解方案. 最后, 通过仿真实验结果说明了该方法是有效的.

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. 所有的排列有 $n!$ 个, 去掉对称的排列(如, $1,2,3,\cdots,n-1,n$ 与 $n,n-1,\cdots,3,2,1,$ 对称)后, 共有 $\frac{1}{2}n!$ 个排列.

本文首先将圆排列问题转化为旅行商问题, 然后用混合粒子群优化算法对其进行求解.

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 ($i=1,2,\cdots,n-1,n,$), 假如排列方式为 $i_1,i_2,\cdots,i_n$, 则长度为

$$D=r_{i1}+2\sqrt{r_{i1}r_{i2}}+2\sqrt{r_{i2}r_{i3}}+\cdots2\sqrt{r_{i-1}r_{in}}+r_{in} \quad (1)$$

假设把 $1\sim n$ 个圆分别放置在 $1\sim n$ 个城市中, 城市 i 和城市 j 之间的距离 d_{ij} ($i=1,2,\cdots,n-1,n$; $j=1,2,\cdots,n-1,n$)为 $d_{ij}=2\sqrt{r_ir_j}$. 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[16]. 也就是说, 再增加一个城市 0, 它与城市 j 的距离 d_{0j} ($j=1,2,\cdots,n-1,n$)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 $1\sim n$ 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变为求解旅行商问题.

2 标准粒子群优化算法

粒子群优化算法是在 1995 年由心理学研究者 Kennedy 博士和从事计算智能研究的 Elberhart 博士受到人工生命研究结果的启发, 基于对鸟群捕食行为的研究而提出的一种群智能的进化计算技术[7].

对于标准粒子群优化算法而言, 即假设有若干个粒子组成的群体, 对于 n 维的空间进行搜索, 对于每一个粒子来说都要考虑自己的历史适应度最好的点和群体内适应度最好的点, 在其基础上考虑自己下一步位置取值. 第 i 个粒子的速度 X_i 和位置 V_i 表示如下:

$$\begin{cases} X_i=(X_{i1},X_{i2},\cdots,X_{iD}) \\ V_i=(V_{i1},V_{i2},\cdots V_{iD}) \end{cases} (1\le i\le n) \quad (2)$$

其中, n 为种群的大小, D 为解空间维数. 第 i 个粒子位置的局部最优值为:

$$P_i=(X_{i1},X_{i2},\cdots,X_{in}) \quad (3)$$

其中, $X_{i1},X_{i2},\cdots X_{in}$ 为当前粒子的历史最优值点的坐标. 粒子位置的全局最优值为:

$$P_{gi}=(X_{gi1},X_{gi2},\cdots,X_{gin}) \quad (4)$$

其中, $X_{gi1},X_{gi2},\cdots,X_{gin}$ 是所有粒子历史最优值点的坐标.

根据以上的定义, 第 i 个粒子的速度和位置更新公式如下:

$$V_{id}^{k+1}=\omega V_{id}^k+C_1\xi(P_{id}^k-X_{id}^k)+C_2\eta(P_{gd}^k-X_{id}^k) \quad (5)$$

$$X_{id}^{k+1}=X_{id}^k+V_{id}^{k+1} \quad (6)$$

其中, ω 为惯性系数; C_1 和 C_2 为学习因子, 它们取值通常是 2; ξ 和 η 为两个在[0,1]均匀分布的伪随机数, 即 ξ, $\eta\in U(0,1)$ [17].

3 混合粒子群优化算法求解圆排列问题

在实际应用中, 标准粒子群优化算法存在着粒子多样性较差, 全局搜索能力较差, 过早陷入局部最优解等缺陷和不足. 从而, 本文针对标准粒子群优化算法存在的问题, 提出了一种混合粒子群优化算法. 它摒弃了标准粒子群优化算法中的通过跟踪极值来更新粒子位置的方法, 而是引入了遗传算法中的交叉和变异操作, 通过粒子同个体极值和群体极值的交叉及粒子自身变异的方式来搜索最优解. 直接以目标函数作为搜索信息, 以一种概率的方式来进行, 这样就能够有效的增强粒子群优化算法的全局寻优能力, 加快算法的进化速度, 提高收敛精度. 以下为利用混合粒子群优化算法求解圆排列问题的实现步骤.

步骤 1. 随机初始化种群, 进行个体编码: 这里, 粒子个体编码采用整数编码的方式, 每个粒子表示历经的所有圆. 比如, 当历经的圆的个数为 10, 个体编码为[5 6 7 4 3 2 8 1 10 9], 表示圆的遍历从 5 开始, 经过, 最终返回 5, 从而完成对圆排列的遍历.

步骤2. 计算适应度值: 粒子适应度值的计算按如

© 中国科学院软件研究所 http://www.c-s-a.org.cn

下(7)式进行.

$$fitness(i)=\sum_{i,j=1}^{n} path_{i,j} \tag{7}$$

其中, n 为圆的数量; $path_{ij}$ 为圆 i,j 之间的距离.

步骤3. 更新粒子: 按如下所述交叉操作和变异操作进行.

(a) 交叉操作: 个体通过和个体极值和群体极值交叉来更新, 交叉方法采用整数交叉法. 首先, 选择两个交叉位置. 然后, 把个体和个体极值或个体与群体极值进行交叉后, 将产生新的个体. 如果产生的新个体存在重复位置则进行调整, 调整方法为用个体中未包括的圆代替重复包括的圆.

假定随机选取的交叉位置为3和5, 那么操作的方法如下: 如果个体为[9 4 2 1 3 7 6 1 0 8 5], 极值为[9 2 1 6 3 7 4 1 0 8 5], 那么对它们进行交叉后, 得到的新个体为[9 4 1 6 3 7 6 1 0 8 5]. 这时, 产生的新个体存在重复位置则进行调整, 将该新的个体调整为[9 4 2 1 3 7 6 1 0 8 5].

进行如此操作之后, 对得到的新个体采用了保留优秀个体策略, 只有当新粒子适应度值好于旧粒子时才更新粒子.

(b) 变异操作: 这里, 变异方法采用个体内部的两位互换方法. 首先, 随机选择变异位置 pos1 和 pos2. 然后, 把两个变异位置互换.

假设选择的变异位置为 2 和 4, 那么将[9 4 2 1 3 7 6 1 0 8 5]变异为[9 1 2 4 3 7 6 1 0 8 5].

进行上述操作之后, 对得到的新个体采用了保留优秀个体策略, 只有当新粒子适应度值好于旧粒子时才更新粒子[15].

步骤 4. 判断是否满足设定的最大代数, 如果满足, 终止程序, 求得最短的结果, 然后, 再加上最前面和最后面的两个圆的半径, 从而求得最终的最短长度, 即最优值. 否则, 转到步骤 2.

4 仿真实验

为了说明本文所给方法的合理性和可行性, 这里考虑了当圆排列问题 n 取 30, 50, 100, $r_i=i,(i=1,2,\cdots n)$ 的情况. 利用文中所给混合粒子群优化算法对上述问题分别进行求解. 这里, 选择的进化次数为 200 次, 个体数目为 100 个.

当 n 取 30 时, 首先, 随机选择各圆的排列半径为: 15 17 13 19 12 11 21 9 23 7 25 5 30 27 3 29 1 2 28 22 4 26 6 24 8 10 20 18 14 16. 求得它的总距离为 796.0705.

利用文中所给混合粒子群优化方法进行一次求解时, 相应的随机轨迹, 优化过程和最优轨迹分别如图 1, 图 2 和图 3 所示. 其中算法训练过程是指从最初路线距离到优化后的路线距离, 不包括最初和最终两个圆的半径.

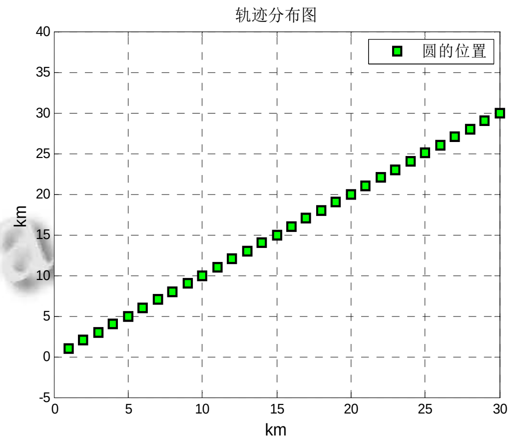

图 1 随机轨迹图

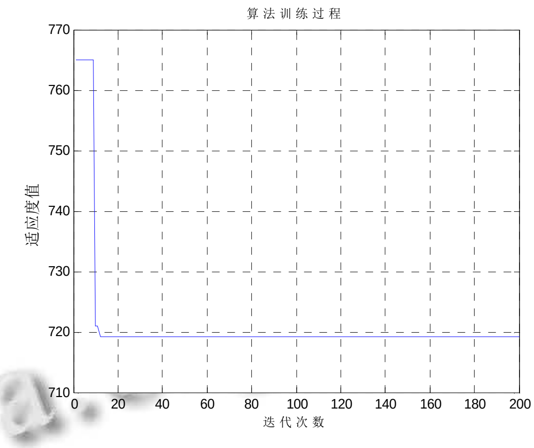

图 2 优化过程图

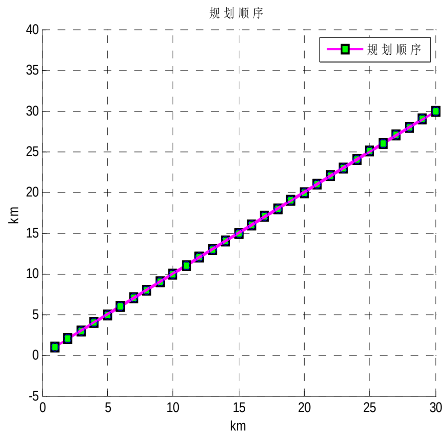

图 3 最优轨迹图

© 中国科学院软件研究所 http://www.c-s-a.org.cn

求得最终各圆的排列半径为: 14 18 15 16 27 3 29 1 28 2 22 4 26 6 24 10 20 8 17 13 19 12 11 21 9 23 7 25 5 30. 此时, 求得最优的总距离为 763.1348.

为了进一步说明本文方法(即利用改进 GA)的有效性, 对上述 n 取 50 和 100 进行分别进行求解一次, 将优化过程图罗列出来.

当 n 取 50 时, 首先, 随机选择各圆的排列半径为: 26 24 28 22 30 20 32 18 34 16 36 14 38 12 40 10 42 8 44 6 46 4 48 2 50 49 1 3 47 5 45 7 43 9 41 11 39 13 37 15 35 17 33 19 31 21 29 23 27 25. 求得它的总距离为 2101.6.

利用文中所给混合粒子群优化方法进行一次求解时, 求得的最优过程如图 4 所示.

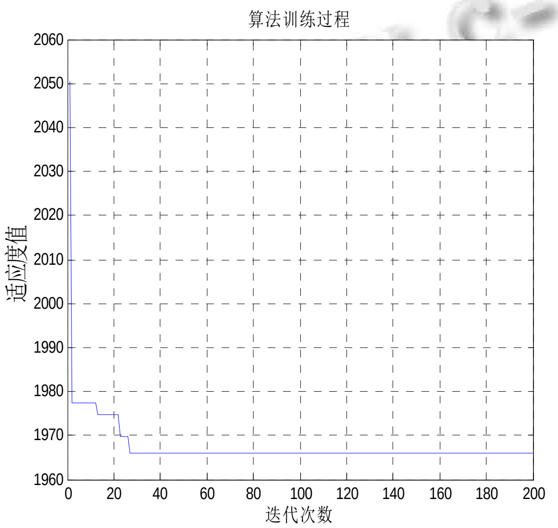

图 4 优化过程图

求得最终各圆的排列半径为: 41 11 39 9 37 13 26 15 35 17 33 19 31 21 29 23 27 25 24 28 22 30 20 32 18 34 16 36 14 38 12 40 10 42 8 44 6 46 4 48 2 491 50 3 47 5 45 7 43. 此时, 求得最优的总距离为 2049.9.

当 n 取 100 时, 首先, 随机选择各圆的排列半径为: 50 51 49 57 53 47 55 45 43 61 37 63 39 65 62 38 35 67 33 31 71 29 73 27 75 25 17 77 23 69 79 21 81 19 83 85 15 87 13 89 11 91 9 93 7 95 98 5 97 3 99 1 100 2 4 96 6 94 8 92 10 90 12 88 14 86 16 84 18 82 20 80 22 78 24 76 26 74 32 28 72 30 70 68 34 66 58 36 64 41 60 40 59 42 44 56 46 54 48 52. 求得它的总距离为 8225.8.

利用文中所给混合粒子群优化方法进行一次求解时, 求得的最优过程如图 5 所示.

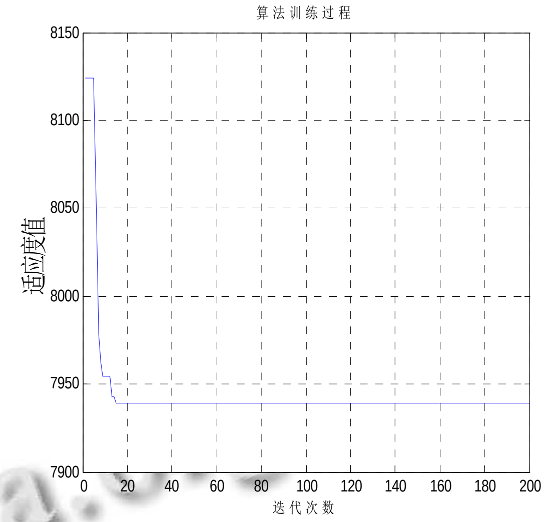

图 5 优化过程图

此时, 求得最终各圆的排列半径为: 79 21 50 81 12 88 14 86 16 84 19 83 18 82 20 80 22 78 24 76 26 74 32 51 85 1 87 13 89 11 91 9 93 7 95 5 97 3 90 15 100 2 98 4 96 6 94 8 92 42 44 56 46 54 48 52 49 57 28 72 30 70 34 66 58 36 64 41 25 40 59 68 53 47 55 45 43 61 37 63 39 65 62 38 35 67 33 31 71 29 73 27 75 60 17 77 23 69 10 99. 此时, 求得最优的总距离为 8029.07.

为了进一步说明本文所给方法(混合粒子群优化算法)的有效性, 对上述 n 取 30, 50 和 100 时, 利用本文方法分别进行 100 次求解后, 将其结果的平均值、最好解、最差解和平均所用时间均罗列在表 1 中. 并且分别利用标准粒子群优化算法和文献[16]中的蚁群模拟退火算法分别对 n 取 30, 50 和 100 进行 100 次求解, 其相应结果也罗列在表 1 中.

表 1 实验结果

方法	n	平均值	最好解	最差解	平均时间/s
混合粒子群优化算法	30	764.48	750.75	765.342	0.35
	50	2041.0	2037.5	2050.6	0.62
	100	8015.7	8004.18	8034.83	0.94
标准粒子群优化算法	30	772.35	759.42	771.348	0.48
	50	2044.8	2039.6	2046.51	0.75
	100	8031.9	8026.7	8050.3	1.01
文献[16]	30	765.78	750.75	770.459	0.67
	50	2041.5	2037.5	2045.5	0.98
	100	8023.1	8019.8	8049.7	1.05

从以上结果可以看出, 利用文中所提方法即混合粒子群算法求得的结果, 相对于标准粒子群算法和文

© 中国科学院软件研究所 http://www.c-s-a.org.cn

献[16]中的蚁群模拟退火算法而言，结果更优，用时更短. 从而可以看出文中所给方法是合理有效的.

5 结语

本文给出了基于旅行商问题和一种混合粒子群优化算法求解圆排列问题的新方法. 首先，将圆排列问题与旅行商问题进行了对比分析. 然后，将圆排列问题转化为旅行商问题，从而得到一个相应的组合优化问题. 接着，利用具有良好收敛性的混合粒子群优化算法给出了对该问题的求解. 最后，利用仿真结果说明了所给方法是可行的.

参考文献

1 Applegate DL, Bixby RE, Chvatal V, et al. The Traveling Salesman Problem: A Computational Study. Princeton: 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.计算机应用,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.预测,2014,33(1):65–69.

4 杨金勇,宋海洲.圆排列包装问题最优解解析.华侨大学学报,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.农机化研究,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.系统管理学报,2014,23(2):247–254 .

7 Kennedy JF, Eberhart RC. Particle swarm optimization. Proc. of IEEE International Conference on Neural Networks. Perth, Australia. 1995. 1942 –1948

8 纪震,廖惠连,吴青华.粒子群优化算法及应用.北京:科学出版社,2009.

9 Yan ZP, Deng C, Zhou JJ, Chi DN. A novel two-subpopulation particle swam optimization. Proc. of 10th Intelligent Control and Automation. Beijing, China. 2012. 4113–4117.

10 Muruganandham A, Wahida Banu Dr RSD. Adaptive fractal image compression using PSO. Procedia Computer Science, 2010, 2: 338–344.

11 Ding W. A new method for image noise removal using chaos PSO and nonlinear ICA. Procedia Engineering, 2011, 24: 111–115.

12 Yu ZF, Li JW, Liu K. Radar emitter recognition based on PSO-BP network. AASRI Procedia, 2012, 1: 213–219.

13 Hu Y, Wang XH, PSO-based energy-balanced double cluster-heads clustering routing for wireless sensor networks. Procedia Engineering, 2011, 15: 3073–3077.

14 刘俊杰,巴海涛.基于粒子群优化的观测站部署算法.指挥控制与仿真,2014,36(3):40–43.

15 史峰,王辉,郁磊,胡斐.智能算法 30 个案例分析.北京:北京航空航天大学出版社,2011.

16 高尚,杨静宇,吴小俊,等.圆排列问题的蚁群模拟退火算法.系统工程理论与实践,2004,8(8):102–106.

17 沈佳杰,江红,王肃.基于多点速度向量的多目标粒子群算法改进.计算机工程与应用,2015,51(2):46–56.

www.c-s-a.org.cn

© 中国科学院软件研究所 http://www.c-s-a.org.cn