

文章编号: 1001-0920(2008)01-0025-05

用蚁群算法求解带平衡约束的圆形布局问题

徐义春, 肖人彬

(华中科技大学 CAD 中心, 武汉 430074)

摘要: 采用启发式方法结合演化算法的思路求解带平衡约束的圆形布局问题. 首先对传统优化模型进行调整, 并探讨了调整的合理性; 然后设计一种分步定位的布局方法, 在此基础上利用蚁群算法寻优; 最后利用局部搜索技术, 在传统模型意义下对布局进行了改进. 数值实验表明, 算法的性能比目前已有的结果有较大的提高.

关键词: 约束圆形布局问题; 蚁群算法; 局部搜索

中图分类号: TP18 **文献标识码:** A

Ant colony algorithm for layout optimization with equilibrium constraints

XU Yi-chun, XIAO Ren-bin

(CAD Center, Huazhong University of Science and Technology, Wuhan 430074, China. Correspondent: XU Yi-chun, E-mail: xuyichun@tom.com)

Abstract: The idea of combining the heuristics with meta-heuristics is applied to solve the problem of layout optimization with equilibrium constraints. The existing model of the problem is adjusted and the rationality is discussed. Then a step-by-step positioning technique is provided, based on which an ant colony optimization (ACO) algorithm is designed. Finally, the results of the ACO are improved by using the technique of local search. The numerical experiments show that the proposed method has much better performance.

Key words: Constrained layout optimization; Ant colony optimization; Local search

1 引言

在一个圆柱形航天器的圆形底面上, 需要布置一些物体 (如仪器、设备等), 这些物体不能互相挤压, 其位置应尽量靠近底面圆心, 同时要求系统的静不平衡量应尽量小. 文献 [1] 以返回式卫星舱布局为背景, 提出了布局优化问题的模型并初步进行了求解.

当小圆的数目增多时, 可供选择的布局形式迅速增多, 使得选择一个最优布局非常困难. 目前的研究大都采用演化算法 [2-7]. 演化算法在一些不可微的、多峰值的目标函数优化问题上有很多成功的应用. 然而对于布局优化问题, 文献 [2-7] 直接在 R^n 空间中进行搜索, 问题本身的启发信息运用较少, 不仅搜索时间较长, 结果的精度也达不到要求.

分步定位法是求解布局问题的启发式方法, 比较著名的有装箱问题的 BL 算法 [8], “求穴和占角”

策略 [9, 10]. 最近, 这类方法也被用于无平衡量要求的圆形布局问题中 [11, 12]. 由于布局位置是直接计算的, 其速度和精度都非常好.

本文将分步定位法引入有平衡量要求的圆形布局问题中. 首先将传统优化模型作了调整, 并指出这种调整对解的精度影响不大; 然后设计了分步定位的策略, 并将该策略结合进蚁群算法中, 得到性能较好的布局; 最后, 利用局部搜索技术, 对蚁群算法的布局解进行了改进. 实验结果表明了算法的有效性.

2 问题的数学模型

2.1 传统的优化模型 [1]

假定有 n 个不等圆, 半径为 (r_1, r_2, \dots, r_n) , 质量为 (m_1, m_2, \dots, m_n) , 圆形容器的半径为 r_0 , 容器的圆心为坐标原点 $o = (0, 0)$, 用向量 $X = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)$ 记录 n 个不等圆的圆心坐标, 则该向量表示一种布局方案. 约束圆形布局要求如下:

收稿日期: 2006-10-08; 修回日期: 2007-03-26.

基金项目: 国家自然科学基金项目 (60474077).

作者简介: 徐义春 (1970—), 男, 湖北红安人, 博士生, 从事优化设计、计算智能等研究; 肖人彬 (1965—), 男, 武汉人, 教授, 博士生导师, 从事人工智能、系统复杂性理论等研究.

1) 最小包络半径为

$$\min f_1(X) = \max(r_i + \sqrt{x_i^2 + y_i^2}); \quad (1)$$

2) 最小静不平衡量为

$$\min f_2(X) = \sqrt{\sum_{i=1}^n (m_i x_i)^2 + \sum_{i=1}^n (m_i y_i)^2}; \quad (2)$$

3) 约束条件为

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < r_i + r_j, \quad (3)$$

$$f_1(X) < R, \quad (4)$$

$$f_2(X) < \dots \quad (5)$$

其中:式(3) 要求小圆互相之间没有覆盖,式(4) 给出容器的最大半径,式(5) 给出系统能承受的最大不平衡量.

2.2 调整后的优化模型

传统的数学模型表明,圆形布局优化是一个多目标优化问题.一般情况下,问题没有全局最优解,只有pareto 最优解集.在pareto 最优解集中,一个点在某些目标上占优,而在另外的目标上则会处于劣势.

为了应用分步定位方案,本文将优化目标定位于要求不平衡量达到最优值 0 的情况下,搜索容器的最小半径.对于任何一个布局 X , 如果要求不平衡量为 0,只需将系统的重心平移到容器的中心即可,因而调整过的优化模型为

$$\min R(X) = \max\left(r_i + \left[\left(x_i - \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}\right)^2 + \left(y_i - \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}\right)^2\right]^{1/2}\right), \quad (6)$$

$$\text{s.t. } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < r_i + r_j, \quad (7)$$

$$f_1(X) < R. \quad (8)$$

定理 1 调整后模型的最优容器半径比传统模型pareto 最优解集中的任意一个解的容器半径最多

$$\text{大 } \frac{\sum_{i=1}^n m_i}{\sum_{i=1}^n m_i}.$$

证明略.

在实践中,为便于航天器的控制,系统能承受的不平衡量的界 是一个比较小的量,而系统总质量

$\sum_{i=1}^n m_i$ 却相对较大.根据定理 1,调整后的模型精度损失并不大.

3 蚁群算法求解布局优化问题

3.1 蚁群算法简介

真实的蚂蚁群体能通过信息素进行通信.蚂蚁

在觅食过程中,在其走过的路径上留下信息素.比较短的路径蚂蚁能很快往返,遗留的信息素会比其他路径要多,从而导致蚁群最终发现最短的路径.蚁群算法(ACO) 就是模拟真实世界中蚂蚁觅食行为的一种优化方法^[13,14].

文献[14] 对 ACO 算法进行了全面介绍.ACO 算法主要有 3 类:1) AS(Ant System) 算法,这是最基本的蚂蚁算法,特点是每只蚂蚁寻找到路径上的信息素都能得到加强;2) ACS(Ant Colony System) 算法,特点是只有最佳路径上的信息素才能得到加强,这样算法的收敛性更快;3) ANTS 算法,类似于树搜索算法.本文的蚁群算法以 ACS 算法为基础.

应用蚁群算法的一般步骤是,首先将优化问题模型化为一个多步决策过程,从而成为一个图论中的最短路径问题.算法为图的每一条边上配备信息素,蚁群将根据边上的信息素浓度采用基于概率的方法选择路径,信息素浓度大的边有更高的机会被选中.蚁群得到一个路径后,根据路径的长度,对信息素进行更新,最短路径上的边的信息素浓度得到加强.所有边上的信息素浓度在一次迭代后将因为“挥发”而有所减少.

ACO 算法因为既使用问题的特点(先验信息),又使用迭代过程中得到的解的信息(后验信息),其优化效果良好,在许多组合优化问题中得到应用.如推销员问题(TSP),车辆调度问题(VRP),以及二次分配问题(QAP) 等^[14].

3.2 布局优化问题的分步决策思路

本文将 n 个小圆的布局问题视为一个 n 步决策问题.决策的每一步,蚂蚁将选择一个小圆,并决定其布局位置.如果以 S_i 记录一个状态,表示此时有 i 个小圆已布局好,还有 $n - i$ 个小圆等待布局,则布局优化问题可用有向图 1 表示.图 1 中,左边表示已完成布局的 3 个小圆{1,2,3},右边是等待布局的小圆{4,5,6}.

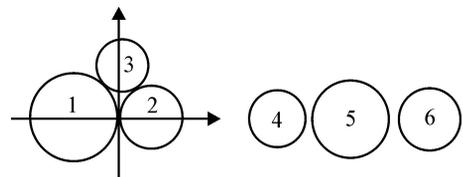


图 1 布局时的一个状态 S_3

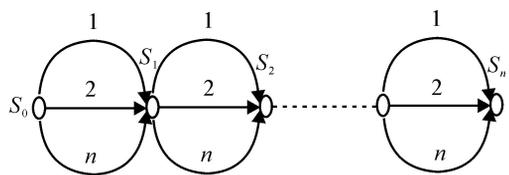


图 2 分步布局的图论表示

图 2 给出了分步布局的图论表示. 在图 2 中, S_0, S_1, \dots, S_n 构成有向图的顶点, 在每个顶点 S_i 后, 有 n 条有标号的边, 不妨以 (i, j) 来标记 S_i 后的第 j 条边, 表示在已经布局 i 个小圆的情况下, 选择编号为 j 的小圆作为第 $i + 1$ 个小圆参与布局. 用一个 2 维数组 $B = \{b_{i,j}\}_{n \times n}$ 来记录每条边的信息素, 其中边 (i, j) 的信息素为 $b_{i,j}$.

蚁群的任务是选择一条从 S_0 到 S_n 的最短路径. 蚂蚁在 S_i 状态下需做两件事情: 1) 从余下的 $n - i$ 个等待布局的小圆中选择一个圆 j ; 2) 确定小圆 j 的位置. 图 1 表明在布局 6 个小圆时, 蚂蚁处于 S_3 时的局面. 左边小圆 $\{1, 2, 3\}$ 已布局好了, 蚂蚁要在右边的小圆 $\{4, 5, 6\}$ 中选择一个作为第 4 个小圆参与布局. 如果蚂蚁决定选择 6, 它还要决定 6 将放在左边的什么位置.

3.3 选择一个新的小圆 j 参加布局

采用蚁群算法的标准方法, 在 S_i 处根据等待布局的小圆所在边 (i, j) 的信息素, 决定选择哪个小圆作为第 $i + 1$ 个布局小圆. 注意, 已经布局过的小圆不参与选择. 用 W_i 记录等待布局的小圆集合, 以 $p_{i,j}$ 记录在 S_i 处选择第 j 个小圆的概率, 则

$$p_{i,j} = \frac{b_{i,j} \times \alpha_j}{\sum_{k \in W_i} b_{i,k} \times \alpha_k}, j \in W_i. \quad (9)$$

其中: α_j 是给予的启发信息, 根据作者的计算经验, 优先布局半径大的小圆可能会有更好的结果, 因此令 $\alpha_j = r_j$; 本文中 α 设为 1.

计算出概率后, 用轮盘赌的方式在 W_i 中选出一个布局小圆.

3.4 确定选中小圆的位置

如同文献[11, 12], 在确定新选出小圆 j 的位置时, 要求小圆 j 至少与两个已经布局的小圆相切. 该要求是合理的, 因为小圆与已经布局的小圆紧挨时, 布局一般会更紧凑, 否则要求更大的容器. 根据两个已布局小圆 p 和 q 的位置, 可直接计算出小圆 j 的位置, 即

$$\begin{cases} (x_j - x_p)^2 + (y_j - y_p)^2 = (r_j + r_p)^2, \\ (x_j - x_q)^2 + (y_j - y_q)^2 = (r_j + r_q)^2. \end{cases} \quad (10)$$

一般情况下, 方程组(10) 可得到 2 组解, 即 2 个已经布局的小圆可以决定 2 个位置, 故小圆 j 最多可有 $2 \times C_2^n$ 个位置. 例如在图 1 的局面中, 如果选择 6 作为第 4 个布局小圆, 则它至少有 3 个合理的位置可供选择, 如图 3 所示. 那么蚂蚁在 $2 \times C_2^n$ 个位置中选择哪一个位置作为小圆 j 的位置呢?

本文采取一种贪心策略. 将小圆 j 放在 $2 \times C_2^n$ 个位置上, 可导致 $2 \times C_2^n$ 个临时布局方案. 根据式

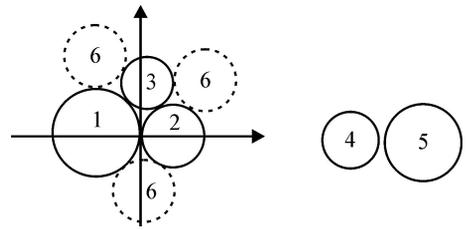


图 3 一个小圆可有多个位置可供选择

(6) 可计算出每个临时布局方案的包络半径, 选择包络半径最小的位置作为小圆 j 的位置.

在布局第 1 和第 2 个小圆时, 需直接给出其位置. 假定前 2 个小圆的编号为 p, q , 则令其位置为 $(-r_p, 0)$ 和 $(r_q, 0)$.

3.5 信息素的更新

每只蚂蚁经过 n 次选择和定位, 最后获得一个布局. 当所有蚂蚁都完成一次布局过程后, 需要对图上各条边的信息素进行更新. 首先, 图中所有边的信息素进行一次“蒸发”, 即

$$b_{i,j} = \rho \times b_{i,j}, \quad \rho < 1 \text{ (本文取 } 0.9 \text{)}. \quad (11)$$

然后, 根据式(7) 计算出每个布局的容器半径, 获得最优布局的蚂蚁可强化其路径上各边的信息素,

$$b_{i,j} = (1 + \frac{C}{R(X)}) \times b_{i,j}. \quad (12)$$

其中: $R(X)$ 为本次迭代获得的最好布局半径; C 为一个常数, 可设定为容器的最大允许半径.

另外, 为避免算法过早收敛, 定义信息素的上下界. 如果 $b_{i,j} < 0.1/n$, 则令 $b_{i,j} = 0.1/n$; 如果 $b_{i,j} > 10/n$, 则 $b_{i,j} = 10/n$.

3.6 蚂蚁算法框架

设定迭代总次数为 K , 蚂蚁总数量为 L , 初始化信息素数组 B , 令所有的 $b_{i,j} = 1/n$.

```

for i = 1 to K { // 迭代 K 次
  for j = 1 to L { // 处理 L 只蚂蚁
    for k = 0 to n - 1 { // 分步布局 n 个小圆
      利用式(9) 计算出待选小圆的概率
      利用轮盘赌方法选出第 k + 1 小圆 c
      如果 k = 0 或 1, 直接给出小圆 c 的位置
      否则, 根据 3.4 节的方法确定小圆 c 的位置
    }
  }
  选出 L 只蚂蚁得到的布局中最好的一个, 并保存.
  根据 3.5 节的规定, 完成信息素的更新
}
输出 K 次迭代中的最佳布局.

```

4 局部搜索

假定蚁群算法得到的解为 x , 为进一步得到最优布局, 使得到的解在传统模型下达到最优, 可将 x 作为种子, 采用局部搜索策略对 x 进行改进. 由于并不要求局部搜索完成全局优化的功能, 只需要很少的迭代次数即可. 局部搜索方法可直接采用现有文献中介绍的各种演化算法.

本文使用粒子群算法 (PSO) 完成局部搜索, 因是标准的 PSO 方法, 故在此不作详述, 可参见文献 [6, 7]. 唯一的不同点是, 在群体初始化时, 令其中一个个体为 x , 其他个体如原来一样随机取值. 随着迭代的进行, x 将不断得到改进.

表 1 各种算法的性能比较

算例	文献[3]	文献[4]	文献[5]	文献[6]	文献[7]	本文算法	
1	计算时间	1 002	—	211	1 002	57.5	0.95
	包络半径	32.662	—	31.967	31.985	32.23	31.90
	不平衡量	0.029	—	0.000 69	0.018 2	7.04E-05	0
2	计算时间	670	—	—	—	—	2
	包络半径	100.6	—	—	—	—	100.004
	不平衡量	0.001 0	—	—	—	—	0
3	计算时间	1 358	1 116	573	2 523	1 128	121
	包络半径	870.331	—	768	843.94	811.806	754
	不平衡量	0.006	8.8E-5	0.000 7	0.003 895	0.002	0

从 3 个算例的结果可以看出, 本文算法在 3 个指标上都超过了目前文献中的结果. 其中: 对于算例 1, 本文算法的精度最好, 速度快了 50 ~ 1 000 倍; 算例 2 是一个相对较“难”的布局, 它很容易落入局部最优, 各种自动优化算法都没选取该例, 只有文献 [3] 做了求解, 但其算法需要人的参与, 而本文算法的精度更好, 速度比文献 [3] 快了约 300 倍; 算例 3 是一个规模较大的、初始数据随机的问題, 其测试结果具有较强的说服力, 本文算法比文献中的结果好了很多, 而速度则快了 5 ~ 10 倍.

6 结 语

对于布局优化问题, 本文首先将优化模型进行调整, 并分析了调整后的模型对结果的影响; 然后根据调整后的目标, 运用分步定位的思想设计了蚂蚁算法; 最后, 对蚂蚁算法的输出结果应用局部搜索, 使得结果更接近传统模型. 实验结果表明, 本文的计算结果在包络半径、不平衡量、计算时间等多个评价指标上均对现有算法有了较大的提高.

应用分步定位方法解决布局问题时, 需要做 2 个工作: (1) 直接计算出布局物体的可行位置; (2) 若有多个位置, 则给出合适的位置评价. 这也符合文献 [9, 10] 给出的“求穴和占角”策略解决布局问题的

5 数值实验

为便于比较, 本文采用文献 [3-7] 中采用的算例来测试本文算法. 共有 3 个算例: 算例 1 有 7 个小圆, 规模稍小; 算例 2 有 9 个小圆, 有理论上的最佳布局; 算例 3 有 40 个小圆, 是一个规模比较大的问题. 3 个算例的原始数据参见文献 [3].

本文的蚂蚁群体选择 10 只蚂蚁, 迭代次数为 200. PSO 局部搜索的粒子数目为 20, 迭代次数为 100. 以计算时间、包络半径和不平衡量为衡量算法性能的标准. 为便于性能比较, 采用文献 [3-7] 的惯例, 将计算时间统一换算为 CPU 主频为 166 M 下, 计算时间以 s 为单位. 各种算法的结果列于表 1.

一般思路.

从另外一个角度看, 传统的迭代优化算法效果都依赖于初始值的选择, 而恰当的初始值又很难找到, 所以一般采用随机值. 本文最后的局部搜索与现有文献基本相同, 只是之前的蚂蚁算法能迅速提供一个良好的初始值.

参考文献 (References)

- [1] Teng Hong-fei, Sun Shou-lin, Ge Wei-hai, et al. Layout optimization for the dishes installed on a rotating table — The packing problem with equilibrium behavioral constraints[J]. Science in China (Series A), 1994, 37(10): 1272-1279.
- [2] 唐飞, 滕弘飞. 一种改进的遗传算法及其在布局优化中的应用[J]. 软件学报, 1999, 10(10): 1096-1102.
(Tang Fei, Teng Hong-fei. A modified genetic algorithm and its application to layout optimization[J]. J of Software, 1999, 10(10): 1096-1102.)
- [3] 钱志勤, 滕弘飞, 孙治国. 人机交互的遗传算法及其在约束布局优化中的应用[J]. 计算机学报, 2001, 24(5): 553-559.
(Qian Zhi-qin, Teng Hong-fei, Sun Zhi-guo. Human-computer interactive genetic algorithm and its application to constrained layout optimization [J]. Chinese J of

- Computers, 2001, 24(5): 553-559.)
- [4] 于洋, 查建中, 唐晓君. 基于学习的遗传算法及其在布局中的应用[J]. 计算机学报, 2001, 24(12): 1242-1249.
(Yu Yang, Cha Jian-zhong, Tang Xiao-jun. Learning based GA and application in packing[J]. Chinese J of Computers, 2001, 24(12): 1242-1249.)
- [5] 刘建, 黄文奇. 利用改进的微分进化算法求解带平衡约束的圆形 packing 问题[J]. 信息与控制, 2006, 35(1): 103-107.
(Liu Jian, Huang Wen-qi. A modified differential evolution algorithm for solving circles packing problem with constraints of equilibrium [J]. Information and Control, 2006, 35(1): 103-107.)
- [6] 李宁, 刘飞, 孙德宝. 基于带变异算子粒子群优化算法的约束布局优化研究[J]. 计算机学报, 2004, 27(7): 897-903.
(Li Ning, Liu Fei, Sun De-bao. A study on the particle swarm optimization with mutation operator constrained layout optimization[J]. Chinese J of Computers, 2004, 27(7): 897-903.)
- [7] 周驰, 高亮, 高海兵. 基于粒子群优化算法的约束布局优化[J]. 控制与决策, 2005, 20(1): 36-40.
(Zhou Chi, Gao Liang, Gao Hai-bing. Particle swarm optimization based algorithm for constrained layout optimization[J]. Control and Decision, 2005, 20(1): 36-40.)
- [8] Liu D Q, Teng H F. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles [J]. European J of Operational Research, 1999, 112(2): 413-420.
- [9] 黄文奇, 许如初. 近世计算理论导引 ——NP 难度问题的背景, 前景及其求解算法研究[M]. 北京: 科学出版社, 2004.
(Huang Wen-qi, Xu Ru-chu. Introduction of computational theory — The study of NP hard problems and the algorithms[M]. Beijing: Science Press, 2004.)
- [10] 黄文奇, 刘景发. 基于欧氏距离的矩形 Packing 问题的确定性启发式求解算法[J]. 计算机学报, 2006, 29(5): 734-739.
(Huang Wen-qi, Liu Jing-fa. A deterministic heuristic algorithm based on euclidian distance for solving the rectangles packing problem [J]. Chinese J of Computers, 2006, 29(5): 734-739.)
- [11] George J A, George J M, Lamar B W. Packing different-sized circles into a rectangular container[J]. European J of Operational Research, 1995, 84(3): 693-712.
- [12] Huang W Q, Li Y, Li C M, et al. New heuristics for packing unequal circles into a circular container [J]. Computers and Operations Research, 2006, 33(8): 2125-2142.
- [13] Dorigo M, Dicaro G. The ant colony optimization metaheuristic [C]. New Ideas in Optimization. McGraw Hill, 1999.
- [14] Maniezzo V, Gambardella L M, De Luigi F. Ant colony optimization[C]. New Optimization Techniques in Engineering. Berlin Heidelberg: Springer-Verlag, 2004.

(上接第 24 页)

- [7] 贺有智, 李君龙. 神经网络在质量矩导弹控制系统上的应用[J]. 系统工程与电子技术, 2005, 27(1): 93-96.
(He Youzhi, Li Jun-long. Using of neural networks for the missile control system with moving masses [J]. Systems Engineering and Electronics, 2005, 27(1): 93-96.)
- [8] 维滕伯格. 多刚体系统动力学[M]. 北京: 北京航空学院出版社, 1986: 30-42.
(Wittenburg Jens. Dynamics of systems of rigid bodies [M]. Beijing: Publishing Beijing Aeronautics Institute, 1986: 30-42.)
- [9] 廖国宾, 于本水, 杨宇光. 质量矩控制技术的机理分析及方程简化研究[J]. 系统工程与电子技术, 2004, 26(1): 1635-1639.
(Liao Guo-bin, Yu Ben-shui, Yang Yu-guang. Study on the theory and simplified equations of mass moment control missile [J]. Systems Engineering and Electronics, 2004, 26(1): 1635-1639.)
- [10] 李士勇. 模糊控制、神经控制和智能控制论[M]. 哈尔滨: 哈尔滨工业大学出版社, 1998: 257-286.
(Li Shi-yong. Fuzzy control, neurocontrol and intelligent cybernetics [M]. Harbin: Harbin Institute of Technology in Press, 1998: 257-286.)
- [11] 徐丽娜. 神经网络控制[M]. 北京: 电子工业出版社, 2003: 144-150.
(Xu Li-na. Neural networks control [M]. Beijing: Publishing House of Electronics Industry, 2003: 144-150.)