计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
44(12):2092~2097,2007

求解不等圆 Packing 问题的一个启发式算法

陈 矛[1,2] 黄文奇[2]
[1](华中师范大学教育信息技术工程研究中心 武汉 430079)
[2](华中科技大学计算机科学与技术学院 武汉 430074)
(mchen_1@163.com)

A Heuristic Algorithm for the Unequal Circle Packing Problem

Chen Mao[1,2] and Huang Wenqi[2]
[1](*Engineering Center for Educational Information Technology, Huazhong Normal University, Wuhan* 430079)
[2](*School of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan* 430074)

Abstract Circle packing problem, one of the NP hard problems, is of great theoretical and practical value. To solve the circle packing problem that encounters in the field of transportation of freight, a heuristic algorithm is proposed for finding a good arrangement of multiple different-sized circles within a larger rectangular container. In this algorithm, the circles are sorted by non-increasing order of radius and packed into the container one by one according to the order. Each circle should be placed inside the container by a corner placement so that the circle does not overlap any other circle and is tangent with two previously packed circles. By pseudo-placing one or more circles to be packed, two greedy methods are introduced to evaluate the benefit of a corner placement, one of which is the degree of placement, and the other is a bounded enumeration strategy that is based on the first one. At each iteration of packing in the resulting algorithm, each circle is packed into the container by a corner placement with the highest benefit according to the bounded enumeration strategy. The experimental results are presented, showing the effectiveness of this algorithm.

Key words NP-hard problem; circle packing problem; heuristic algorithm; corner placement; bounded enumeration

摘 要 求解具有NP难度的圆形packing问题具有很高的理论与实用价值.现提出一个启发式方法,求解了货运中常遇到的矩形区域内的不等圆packing问题.此算法首先将待布局圆按半径大小降序排列,然后用占角动作来逐个放置.通过试探性地放入一个或多个待布局圆,给出了占角动作的度以及更全局的有限枚举策略来评价占角动作的优度.在放置每一个圆时,以贪心的方式选取当前具有最大优度的占角动作来放置.最后用测试算例验证了算法的高效性.

关键词 NP难问题;圆形packing问题;启发式算法;占角动作;有限枚举策略

中图法分类号 TP301.6

给定一个布局空间和若干待布局物体,packing问题就是如何将这些物体按一定的约束条件合理地摆放在布局空间中,并达到某些最优指标[1-3].圆形packing问题是经典的优化问题,在航空航天、造船、交通运输、板材加工、服装等行业有着广泛的应用背景,packing问题解决得好坏对这些行业生产的合理

收稿日期:2006-07-05;修回日期:2007-06-13
基金项目:国家自然科学基金项目(10471051);国家"九七三"重点基础研究发展规划基金项目(2004CB318000);"十一五"国家科技支撑计划重点基金项目(2006BAK11B01)

性和经济性等指标具有相当大的影响. 圆形 packing 问题已被证明是 NP 难度的,这意味着该问题不存在既完整严格又不是太慢的求解算法[4-5]. 为了满足实际需要,人们于是着手研究非绝对完整但是快速适用的启发式算法.

圆形 packing 问题的布局区域有圆形、矩形和三角形等形状,待布局圆又可分为等圆和不等圆两类[6-12]. 本文将对运输装载中经常碰到的矩形框内不等圆 packing 问题进行求解. 通过调整相邻两行圆间圆心的夹角,Dowsland[9]提出了一个启发式算法来求解矩形区域内等圆 packing 问题. Fraser 和 George[10]利用此算法设计了一个求解造纸厂纸卷运输问题的实用软件. George 等人[11]为了解决轮船货运中常遇到的集装箱内管道装运问题,设计了相应的遗传算法. 在弹性势能模型[6,12]的基础上,文献[7]利用禁忌搜索的思想,对矩形框内圆形 packing 问题给出了一个有效算法.

长期以来,在人们从事铺地、砌墙等布局劳动中,积累了大量的实践经验. 通过借鉴人们的实践经验,本文提出一个新颖的启发式算法来求解矩形框内圆形 packing 问题. 对测试算例的计算表明,该算法效率较高.

1 问题的描述

给定一个长、宽分别为 L, W 的矩形框及 n 个半径为 $R_1,\cdots,R_n$ 的圆 $c_1,\cdots,c_n$,圆形 packing 问题就是问能否将这些圆互不重叠地放到矩形框中. 此问题更形式化的表述如下:

将二维笛卡儿坐标的原点取在矩形框的左下顶点,如图 1 所示:

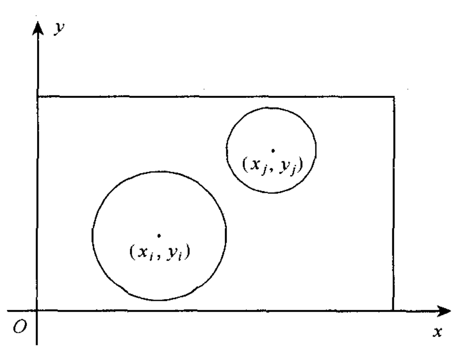

Fig. 1 Circle packing problem.

图 1 圆形 packing 问题示意图

记圆 c_i 的圆心坐标为 (x_i, y_i),问是否存在 $2n$ 个实数 $x_1, y_1, \cdots, x_n, y_n$ 满足以下两个约束条件:

$$R_i \leqslant x_i \leqslant L - R_i,\ R_i \leqslant y_i \leqslant W - R_i,\quad i \in \{1,2,\cdots,n\}, \tag{1}$$

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geqslant R_i + R_j,\quad i, j \in \{1,2,\cdots,n\},\ i \neq j \tag{2}$$

如果存在,则具体给出一组合乎条件的解.

约束条件: 1) 要求任一放到矩形框内的圆 c_i 不与矩形框的边有重叠;2) 要求任意两个已放置到矩形框内的圆 c_i 和 c_j 彼此之间没有重叠.

2 启发式算法

2.1 最大度优先原则

本文启发式算法是一个增长式算法,即所有待布局圆是按一定次序被逐一放置到矩形框中. 在描述算法之前,首先给出如下定义:

定义 1. 格局. 假定 $m(\geqslant 0)$ 个圆已经互不重叠地放到矩形框内,这 m 个圆的位置的总体集合称为一个格局,记为 C. 如果 $m = n$,即所有圆都已放入矩形框中,此格局称为成功格局;反之,若 $m < n$,且矩形框外没有一个圆能够在满足约束条件 1) 和 2) 的前提下放入矩形框中,此格局就称为失败格局. 我们把成功格局和失败格局都称为结果格局.

定义 2. 占角动作. 将已放入矩形框内的圆的集合记为 M,并把矩形框的 4 个边也看做已放"圆". 对于当前待布局圆 c_i,如果做一个动作把 c_i 放到矩形框中某个位置后,使得该圆与 M 中任一圆不重叠且与 M 中至少两个圆相切,就把该动作称为占角动作. 一个占角动作可由一个三元组 $p(c_i, x, y)$ 表示,即把圆 c_i 放到圆心坐标为 (x, y) 的位置.

在任一格局中,一个待布局圆可能有多个候选的占角动作. 如图 2 所示,矩形框中已放入 c_1 和 c_2 两个圆,矩形框外是当前待布局圆 c_3. 如图中虚线圆所示,c_3 共有 6 个候选的占角动作.

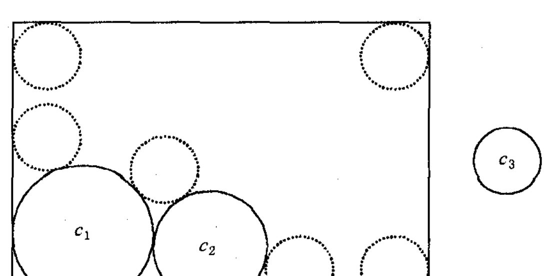

Fig. 2 Corner placement for circle c_3.

图 2 占角动作示意图

在当前已有 m 个圆放入矩形框的格局下,M 中共有 $m+4$ 个圆(包括矩形框的 4 个边),这样第 $(m+1)$ 个圆的占角动作数将不大于 $2C_{m+4}^2$. 为了评价占角动作的优度,我们给出了一个称为占角动作的度的量化指标.

定义 3. 占角动作的度. 本算法中待布局圆是按一定次序 $c_1, c_2, \cdots, c_i, c_{i+1}, \cdots, c_n$ 被逐个放入矩形框,在已放入 $i-1$ 个圆的格局 C 中,假定当前待布局圆 c_i 有 N 个占角动作 $p_1, p_2, \cdots, p_N$. 在占角动作 $p_j(j=1,2,\cdots,N)$ 把圆 c_i 放到矩形框中后,新格局 C^* 下当前待布局圆 c_{i+1} 的占角动作数就称为占角动作 p_j 的度.

直观上看,为了得到更好的布局结果,在选择某个占角动作时,应使它对矩形框内空间的完整性破坏得尽量小,从而使得下一个待布局圆有更多的可放位置(即有更多的占角动作数),这就是最大度优先原则.

定义 4. 最大度优先原则. 从当前待布局圆 c_i 的 N 个占角动作 $p_1, p_2, \cdots, p_N$ 中,选择具有最大度的占角动作 $p_j(c_i, x_j, y_j)$ 来放置圆 c_i. 若占角动作 $p_j(c_i, x_j, y_j)$ 和 $p_k(c_i, x_k, y_k)$ 具有相同的最大度,则按以下优先准则来选择一个,即若有:

1) $x_j < x_k$, 或

2) $x_j = x_k \wedge y_j < y_k$,

则选择占角动作 $p_j(c_i, x_j, y_j)$ 来放置圆 c_i.

2.2 基于最大度优先原则的算法: A_0

如前所述,待布局圆是依序逐个被放入矩形框的. 本算法将所有待布局圆按半径大小进行降序排列成一个表,记为 L,然后依此顺序来逐个放置. 给定一个格局 C 和一个表 L,按照最大度优先原则,就得到基本的布局算法 $A_0(C, L)$:

1) 若 L 为空,转步 5),否则计算当前待布局圆 c_i 的占角动作数 N;

2) 若 $N=0$,没有可选的占角动作,转步 4);

3) 计算这 N 个占角动作的度,选择度最大的占角动作 $p(c_i, x, y)$ 来放置圆 c_i,更新 C 和 L,转步 1);

4) 返回该失败格局;

5) 所有圆都已放入矩形框,返回该成功格局 C.

在每一步迭代中,A_0 以最大度优先原则来放置表 L 中第 1 个圆,然后更新格局 C 和表 L(将该圆从 L 中删除). 在某一步迭代中,若 L 中第 1 个圆没有可选的占角动作,即矩形框中没有位置来摆放它,此时就是失败格局,A_0 返回该失败格局;若所有圆都已互不重叠地放入矩形框中(L 为空),则 A_0 返回该成功格局.

假定 m 次迭代已把 m 个圆放到矩形框中,第 $m+1$ 步迭代将把第 $m+1$ 个圆放入矩形框,第 $m+1$ 个圆最多有 $2C_{m+4}^2$ 个占角动作. 在分别计算每个占角动作的度之后,A_0 选出具有最大度的占角动作来放置第 $m+1$ 个圆,这样放置一个圆的计算复杂度是 $O(n^2)$,所以 A_0 的计算复杂度是 $O(n^3)$.

2.3 有限枚举策略

占角动作的度只是一个局部的评价标准,它采用向前试探一步的策略,通过考虑已放入矩形框中的圆与当前待布局圆 c_i 以及下一个待布局圆 c_{i+1} 之间的关系来评价一个占角动作的优度,而没有考虑其他待布局圆与已放圆之间的关系.

为了更全局地评价占角动作的优度,在最大度优先原则的基础上,我们给出了一个有限枚举策略来评价占角动作的优度,如程序 *Benefit*(c_i, x, y, C, L) 所示. 给定格局 C 和待布局圆构成的表 L,程序 *Benefit*(c_i, x, y, C, L) 利用算法 A_0 来试探性地放置待布局圆,然后以所得结果格局作为占角动作 $p(c_i, x, y)$ 的优度.

Procedure *Benefit*(c_i, x, y, C, L)

Begin

  把 C 和 L 分别复制给 C' 和 L';

  试探性地把 c_i 放到 (x, y),并更新 C' 和 L';

  $C' = A_0(C', L')$;

  If(C' 是个成功的格局)

    返回该格局 C';

  Else

    返回该格局的占空比 *density*(C');

End

程序 *Benefit*(c_i, x, y, C, L) 首先试探性地用占角动作 $p(c_i, x, y)$ 将圆 c_i 放入矩形框,然后利用 A_0 把剩下的待布局圆试探性地放入,直至得到一个结果格局 C'. 若 C' 是一个成功格局,*Benefit*(c_i, x, y, C, L) 就返回该成功格局;否则就返回该失败格局的占空比 *density*(C')(即已放入矩形框中圆的面积之和除以矩形框的面积),作为占角动作 $p(c_i, x, y)$ 的优度.

与最大度优先原则相比,有限枚举策略"看得更长远",在试探性地放置尽可能多的待布局圆之后,以所得结果格局来评价一个占角动作. 需要指出的是,这里所说的试探性的放置,是一种模拟的、非确定性的摆放. 如程序 *Benefit*(c_i, x, y, C, L) 所示,

在考查占角动作 $p(c_i, x, y)$ 的优度之前，须把格局 C 以及表 L 复制给 C' 和 L'，这样试探过程就不会改变真实的格局 C 和表 L.

2.4　启发式算法 A_1

有了新的、更全局的评价标准 *Benefit*()，我们将利用这个标准来评价占角动作的优度，并在每一步迭代中，选择优度最大的占角动作来确定性地摆放一个圆. 给定格局 C 和表 L，启发式算法 A_1 描述如下：

1) 计算当前待布局圆 c_i 的占角动作数 N；

2) 若 $N=0$，转步 6)；

3) 对每一个占角动作 $p(c_i, x, y)$，按有限枚举策略计算其优度：$b = Benefit\ (c_i, x, y, C, L)$；

4) 若 b 是一个成功格局，则返回该格局，成功停机；

5) 从这 N 个占角动作中选择具有最大优度 b' 的占角动作 $p(c_i, x', y')$ 来放入 c_i，更新 C 和 L，转步 1)；

6) 当前待放圆没有可选的占角动作，失败停机.

每一步迭代中，当前待布局圆 c_i 都是表 L 的第 1 个圆. 给定一个格局，假定当前待布局圆 c_i 有 N 个候选的占角动作. 对每一个占角动作，A_1 都要利用有限枚举策略 *Benefit*()来计算其优度，然后选择优度最大的占角动作来摆放 c_i. *Benefit*()的计算复杂度是 $O(n^3)$，则 A_1 确定性地放置一个圆的计算复杂度是 $O(N \times n^3)$. N 的数量在 $O(n^2)$ 以内，这样，A_1 的计算复杂度是 $O(n^6)$.

由 A_1 可知，一旦 *Benefit*()在计算某个占角动作的优度时找到了一个成功格局，就立即返回该成功格局并停机，而不再继续迭代下去. 事实上，A_1 的计算速度是相当快的.

3　算例及评论

我们将本文启发式算法用 C 语言在 2.4GHz PC 上编程进行了大量的实例测试. 对于空间较宽松的情形，计算毫无例外都进行得十分顺利快捷，这里选的都是空间很紧张的几个典型算例[7]，如表 1 所示：

Table 1　The Benchmark Instances and the Run Time Comparison of the Proposed Heuristic Algorithm (HA) with IOP and HTS

表 1　测试算例及算法计算时间

Instance	n	L, W	R_i	Running Time(s)		
				IQP	HTS	HA
1	6	300, 200	$R_1=100$, $R_2=53.589$ $R_3=R_4=30.094$ $R_5=21$, $R_6=20$	2.73	0.31	0.00
2	14	3.4142, 3.4142	$R_1=R_2=1$, $R_3=0.5$ $R_4=R_5=0.42$ $R_6=R_7=0.32$ $R_8=\cdots=R_{12}=0.2$ $R_{13}=R_{14}=0.17$	244.28	5.75	0.08
3	19	160, 80	$R_1=\cdots=R_8=20$ $R_9=\cdots=R_{11}=8.28$ $R_{12}=\cdots=R_{19}=5$	59.56	2.99	0.14
4	21	200, 200	$R_1=100$ $R_2=\cdots=R_5=17.157$ $R_6=\cdots=R_{13}=8.578$ $R_{14}=\cdots=R_{21}=5$	207.75	5.13	0.12
5	32	160, 120	$R_1=\cdots=R_{12}=20$ $R_{13}=\cdots=R_{18}=8.28$ $R_{19}=\cdots=R_{28}=5$ $R_{29}=\cdots=R_{32}=3.431$	534.68	14.38	0.27

表1中给出了本文算法的计算时间,还列出了拟人拟物法(IPQ)[12]和启发式禁忌搜索算法(HTS)[7]的计算时间. IPQ和HTS都是建立在弹性势能模型基础上的启发式算法. 建立在弹性势能模型基础上的算法的初始解都是随机给定的,所以IPQ和HTS对每个算例都进行了5次计算,表中给出的是它们的平均计算时间,所用机器是IBM PC586. 本文算法是一种确定性算法,所以对每个算例只需计算一次. 对所有算例,本文算法在不到1s时间内就找到成功布局结果. 图3和图4分别给出了由本文算法得到的算例2和算例5的最优布局结果.

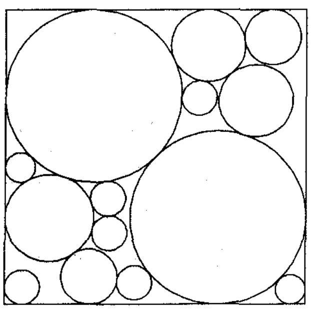

Fig. 3 Packing result for instance 2.
图3 算例2的布局结果

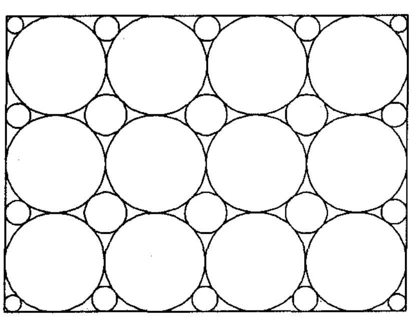

Fig. 4 Packing result for instance 5.
图4 算例5的布局结果

在货运中圆形物体放置、布料的裁剪等实际应用中,本算法具有实际价值,稍做修改,就可以很好地应用在工程领域. 在下一步的工作中,我们将发展本文的思想和技术,为具有更重大意义的各种形状和大小的多边形布局问题找到高效的实用求解算法.

参考文献

[1] K A Dowsland, W B Dowsland. Packing problems [J]. European Journal of Operational Research, 1992, 56(1): 2-14

[2] A Lodi, S Martello, M Monaci. Two-dimensional packing problems: A survey [J]. European Journal of Operational Research, 2002, 141(2): 241-252

[3] D Pisinger. Heuristics for the container loading problem [J]. European Journal of Operational Research, 2002, 141(2): 382-392

[4] D S Hochbaum, M Wolfgang. Approximation schemes for covering and packing problems in image processing and VLSI [J]. Journal of the Association for Computing Machinery, 1985, 32(1): 130-136

[5] J Leung, T Tam, C S Wong, *et al*. Packing squares into square [J]. Journal of Parallel and Distributed Computing, 1990, 10(3): 271-275

[6] W Q Huang, R C Xu. Two personification strategies for solving disks packing problem [J]. Science in China (Series E), 1999, 42(6): 595-602

[7] Kang Yan, Huang Wenqi. A heuristic algorithm based on tabu search for the disks packing problem [J]. Journal of Computer Research and Development, 2004, 41(9): 1554-1558 (in Chinese)
(康雁, 黄文奇. 基于禁忌搜索的启发式算法求解圆形packing问题[J]. 计算机研究与发展, 2004, 41(9): 1554-1558)

[8] J B M Melissen, P C Schuur. Packing 16, 17 or 18 circles in an equilateral triangle [J]. Discrete Mathematics, 1995, 145(1-3): 333-342

[9] K A Dowsland. Palletisation of cylinders in cases [J]. Operation Research Spectrum, 1991, 13(1): 204-212

[10] H J Fraser, J A George. Integrated container loading software for pulp and paper industry [J]. European Journal of Operational Research, 1994, 77(3): 466-474

[11] J A George, J M George, B W Lamar. Packing different-sized disks into a rectangular container [J]. European Journal of Operational Research, 1995, 84(3): 693-712

[12] Kang Yan, Huang Wenqi. A heuristic algorithm for solving the disks packing problem [J]. Journal of Computer Research and Development, 2002, 39(4): 410-414 (in Chinese)
(康雁, 黄文奇. 求解圆形packing问题的一个启发式算法[J]. 计算机研究与发展, 2002, 39(4): 410-414)

Chen Mao, born in 1975. Received his Ph. D. degree in computer software and theory from Huazhong University of Science and Technology, Wuhan, China in 2007. His primary research interests include algorithm for solving the packing problem and other NP-hard problems.
陈 矛,1975年生,博士,主要研究方向为布局问题以及其他NP难问题的高效算法设计.

Huang Wenqi, born in 1938. Professor and Ph.D. supervisor in Huazhong University of Science and Technology. His primary research interests include computational complexity theory, and quasi-physical quasi-human algorithms for solving NP-hard problems.

黄文奇,1938年生,教授,博士生导师,华中科技大学计算机科学理论研究所所长,中国计算机学会理论计算机科学专业委员会常务委员,美国数学会"数学评论"评论员,主要研究方向为可计算性与计算复杂性理论及应用、NP难问题的拟物拟人算法设计.

Research Background

The two dimensional (2D) circle packing problem is a famous cutting and packing problem. It consists of placing a given set of circles in a container without overlap. The usual objective is to maximize the material utilization and hence to minimize the "waste" area. This problem is known to be NP-hard and is encountered in many industries (textile, paper, glass, *etc*). For NP-hard problems, there does not exist an algorithm that is complete, rigorous and efficient. Consequently, various heuristic algorithms have been proposed for circle packing problems. In this paper, two novel placement heuristics are proposed to solve the circle packing problem, the key idea of which is a quantified measure called degree of placement to evaluate the benefit of packing a circle into the container. Computational results show that the performance of the presented algorithm outperforms that of two previous reported algorithms in the literature. Our work is supported by the National Science Foundation of China (10471051) and partially supported by the National Grand Fundamental Research 973 Program of China (2004CB318000).