

文章编号:1671-6833(2004)04-0052-04

求解三角形Packing 问题的占角算法

王瑞民¹, 陈卓²

(1. 郑州大学信息工程学院, 河南 郑州 450052; 2. 华中科技大学计算机科学与技术学院, 湖北 武汉 430074)

摘 要: 三角形Packing 问题是NP 难的, 其完整算法的时间复杂度是指数型的. 建立了求解三角形Packing 问题的数学模型; 沿着拟人的途径, 通过把待解决的问题与人类的社会经验作对比, 得到了求解三角形Packing 问题的若干拟人策略, 并以此策略为基础发展出一种高效的近似求解算法. 实验结果表明, 沿此途径得出的求解三角形Packing 问题的算法, 针对三角形数目较多的情况, 具有较好的完整性和较低的时间复杂度.

关键词: 三角形; Packing 问题; 价值度; 拟人策略

中图分类号: TP 301

文献标识码: A

0 引言

Packing 问题^[1]是一个 NP 难问题, 该问题吸引了大量数学家和计算机科学理论学家的注意. 然而, 在 20 世纪 60 年代以前, 人们对这一问题还仅仅停留在理论的研究上. 在计算技术高度发达的今天, 这方面的工作才在理论上和实践上取得了显著效果.

Packing 问题分为三维和二维 Packing 问题. 三维 Packing 问题的研究主要集中在立方体和集装箱的填装上^[2], 对一般的三维 Packing 问题的研究还处于起步阶段; 二维 Packing 问题的研究主要集中在矩形 Packing 问题、方格 Packing 问题、多边形 Packing 问题和圆形 Packing 问题上^[3~5].

多边形 Packing 问题是一类比较复杂的问题, 已经证明了平移多边形的 Packing 问题是 NP 完全问题. 对于任意多边形的 Packing 问题(多边形既可平移又可旋转)是否为 NP 完全问题还没有得以证明^[6]. 如何快速求解任意多边形的 Packing 问题, 目前国际学术界还没有给出令人满意的算法.

三角形 Packing 问题是多边形 Packing 问题的简化情况, 但其完整算法的时间复杂度仍然是指数型的, 因此必须从新的途径寻找快速高效的算

法. 文献[7]对单位等边三角形 Packing 问题进行了初步的研究, 得到了求解单位等边三角形 Packing 问题的拟人策略. 作者以此为基础, 对一般的三角形 Packing 问题进行深入细致的研究, 沿着拟人的途径, 根据人们生产实践中处理这类问题的经验, 提出了几种拟人策略, 发展出两种高效率的求解算法. 理论分析和实验结果表明, 该算法具有较低的时间复杂度和较高的完整度.

1 三角形Packing 问题与拟人思想

三角形 Packing 问题是指: 已知长和宽一定的矩形空容器 R 和 N 个大小形状确定的三角形(每个三角形的 3 个边长已知), 问: 这 N 个三角形能否互不重叠地放入矩形容器 R 里? 若能, 则给出每个三角形在矩形容器里的位置和方向; 若不能, 则作出该算法放不下的回答.

如图 1 所示, 矩形容器由 4 个两两互不重叠的三角形块 1、2、3、4 的边围成. 这样就把三角形块放入矩形容器内的三角形 Packing 问题转化为: 判断在构成矩形容器的 4 个三角形和已经放入矩形容器的三角形中找到合适的位置, 以便把待放的三角形放入矩形容器内, 从而转化为判断三角形与三角形的位置关系问题.

收稿日期: 2004-09-11; 修订日期: 2004-10-12

基金项目: 郑州大学引进人才科研基金资助项目

作者简介: 王瑞民(1974-), 男, 河南省滑县人, 郑州大学讲师, 博士, 主要从事密码学、网络安全和计算复杂性理论方面的研究.

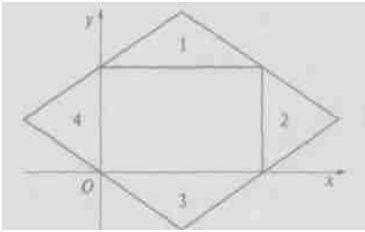


图 1 矩形容器的示意图

Fig. 1 Rectangular container

此问题可形式化地表示为:对于任意给定的正整数 N 和任意给定的 $3N+2$ 个正实数组成的有序数组 $\{Length, Width, a_1, b_1, c_1, a_2, b_2, c_2, \dots, a_N, b_N, c_N\}$ (其中 $a_i < b_i + c_i, b_i < c_i + a_i, c_i < a_i + b_i, i = 1, 2, \dots, N$)。记 $f(x, y, x_1, y_1, x_2, y_2) = (y - y_2)(x_1 - x_2)(x - x_2)(y_1 - y_2)$, 问是否存在 $6N$ 个实数组成的有序数组 $\{x_{11}, y_{11}, x_{12}, y_{12}, x_{13}, y_{13}, x_{21}, y_{21}, x_{22}, y_{22}, x_{23}, y_{23}, \dots, x_{N1}, y_{N1}, x_{N2}, y_{N2}, x_{N3}, y_{N3}\}$, 使得

- (1) 对于 $\{1, 2, \dots, N\}$ 中任意的正整数 i 有 $0 \leq x_{i1}, x_{i2}, x_{i3} \leq Length; 0 \leq y_{i1}, y_{i2}, y_{i3} \leq Width$;
$$a_i^2 = (x_{i2} - x_{i3})^2 + (y_{i2} - y_{i3})^2$$
$$b_i^2 = (x_{i3} - x_{i1})^2 + (y_{i3} - y_{i1})^2$$
$$c_i^2 = (x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2$$

(2) 对于 $\{1, 2, \dots, N\}$ 中任意两个不同的正整数 i, j , 下列 6 个式中至少有一个式子成立:

$$\begin{cases} \max \{f(x_{i1}, y_{i1}, x_{j2}, y_{j2}, x_{j3}, y_{j3}), f(x_{j1}, y_{j1}, x_{i2}, y_{i2}, x_{i3}, y_{i3}), \\ f(x_{i1}, y_{i1}, x_{j2}, y_{j2}, x_{i3}, y_{i3}), f(x_{j1}, y_{j1}, x_{i2}, y_{i2}, x_{j3}, y_{j3})\} \leq 0; \\ \max \{f(x_{i2}, y_{i2}, x_{j3}, y_{j3}, x_{i1}, y_{i1}), f(x_{j1}, y_{j1}, x_{i3}, y_{i3}, x_{j1}, y_{j1}), \\ f(x_{i2}, y_{i2}, x_{j3}, y_{j3}, x_{i1}, y_{i1}), f(x_{j2}, y_{j2}, x_{i3}, y_{i3}, x_{j1}, y_{j1}), \\ f(x_{i2}, y_{i2}, x_{j3}, y_{j3}, x_{i1}, y_{i1}), f(x_{j3}, y_{j3}, x_{i3}, y_{i3}, x_{i1}, y_{i1})\} \leq 0; \\ \max \{f(x_{i3}, y_{i3}, x_{j1}, y_{j1}, x_{i2}, y_{i2}), f(x_{j1}, y_{j1}, x_{i1}, y_{i1}, x_{i2}, y_{i2}), \\ f(x_{i3}, y_{i3}, x_{j1}, y_{j1}, x_{i2}, y_{i2}), f(x_{j2}, y_{j2}, x_{i1}, y_{i1}, x_{i2}, y_{i2}), \\ f(x_{i3}, y_{i3}, x_{j1}, y_{j1}, x_{i2}, y_{i2}), f(x_{j3}, y_{j3}, x_{i1}, y_{i1}, x_{i2}, y_{i2})\} \leq 0; \\ \max \{f(x_{j1}, y_{j1}, x_{i2}, y_{i2}, x_{j3}, y_{j3}), f(x_{i1}, y_{i1}, x_{j2}, y_{j2}, x_{j3}, y_{j3}), \\ f(x_{j1}, y_{j1}, x_{i2}, y_{i2}, x_{j3}, y_{j3}), f(x_{i2}, y_{i2}, x_{j2}, y_{j2}, x_{j3}, y_{j3}), \\ f(x_{j1}, y_{j1}, x_{i2}, y_{i2}, x_{j3}, y_{j3}), f(x_{i3}, y_{i3}, x_{j2}, y_{j2}, x_{j3}, y_{j3})\} \leq 0; \\ \max \{f(x_{j2}, y_{j2}, x_{j3}, y_{j3}, x_{j1}, y_{j1}), f(x_{i1}, y_{i1}, x_{j3}, y_{j3}, x_{j1}, y_{j1}), \\ f(x_{j2}, y_{j2}, x_{j3}, y_{j3}, x_{j1}, y_{j1}), f(x_{i2}, y_{i2}, x_{j3}, y_{j3}, x_{j1}, y_{j1}), \\ f(x_{j2}, y_{j2}, x_{j3}, y_{j3}, x_{j1}, y_{j1}), f(x_{i3}, y_{i3}, x_{j3}, y_{j3}, x_{j1}, y_{j1})\} \leq 0; \\ \max \{f(x_{j3}, y_{j3}, x_{j1}, y_{j1}, x_{j2}, y_{j2}), f(x_{i1}, y_{i1}, x_{j1}, y_{j1}, x_{j2}, y_{j2}), \\ f(x_{j3}, y_{j3}, x_{j1}, y_{j1}, x_{j2}, y_{j2}), f(x_{i2}, y_{i2}, x_{j1}, y_{j1}, x_{j2}, y_{j2}), \\ f(x_{j3}, y_{j3}, x_{j1}, y_{j1}, x_{j2}, y_{j2}), f(x_{i3}, y_{i3}, x_{j1}, y_{j1}, x_{j2}, y_{j2})\} \leq 0. \end{cases}$$

若存在, 则给出一组符合上述条件的实数。
拟人方法是将原始的数学问题与人类生产实践中遇到的实际问题相比拟, 模拟人们的思维活

动, 把人类在生产实践中处理各种复杂矛盾问题的经验、思想和方法加以精确化、完整化并形式化为算法, 用以求解相应数学问题的思路和方法。这种方法的精神在于, 在某些困难的算法问题面前, 当纯粹数学的思维显得似乎枯竭的时候, 可以到人类社会中去寻求启发, 而在最后设计算法时又不必拘泥于社会现象的真实性。

通常情况下, 拟人算法的效率之所以比生物遗传算法、神经网络方法、淬火法要高, 其原因就是拟人方法是在向人类学习, 而人类比起遗传、神经网络、淬火世界里的那些蛋白质单个的神经元及晶体显然有高得多的智慧^[9]。但是, 如何正确利用人类处理社会问题的经验并形式化为恰当的数学模型, 则需要长期细心的工作。这种得出算法的过程的艰苦性, 可以说是拟人途径的一个缺点。

把三角形块放入矩形容器内, 主要考虑以下两个方面的因素: 一、找到可利用的空间; 二、在可利用的空间中寻找位置, 尽可能减少空间的浪费。

这两个标准, 充分地考虑了问题的主要矛盾, 也是人们手工处理实际问题所凭借的客观依据。本文从这两个要点出发, 制定了解决实际问题的策略和方案。理论分析和实践证明, 虽然在十分困难的场合下, 该方法还表现不出熟练的设计人员那样的机智和灵巧, 但在许多场合它确实能如人做的同样准确并且比人要快的多。这也为解决其它问题提供了一种思路和参考。

2 三角形Packing 问题的相关概念

定义 1 两个三角形的相切。若两个三角形有公共点, 且相交部分的面积为零。则称这两个三角形相切。

定义 2 角区。已知两个相切的三角形, 在这两个三角形中各取一边, 若:

- (1) 这两边有且仅有一个交点。记该点为 O 。
- (2) 这两边所在的射线 (以 O 为端点) 形成的角为小于 π 的正角。
- (3) 条件 (2) 中的角内不含有这两个三角形中除条件 (1) 外的其它边。

则 (2) 中的角称为这两个相切三角形的一个角区, 该角的大小记为该角区的角度。两个三角形中和 (2) 中的角的始边相重合的部分称为该角区的始边; 两个三角形中和 (2) 中的角的终边相重合的部分称为该角区的终边; 称 (1) 中的交点为该角区的顶点。

由定义可知, 两个相切三角形的两边是否构

成角区,与将要放入该角区的三角形的大小和形状无关.这些角区,好像人们加工材料过程中裁下的“边角余料”.如能够优先考虑利用这些“边角余料”,便能充分提高整个材料的利用率.因此,能否充分高效的利用这些角区,直接决定着算法的最终效果.

定义 3 占角动作:如果一个三角形的一边和一个角区的始边或终边相贴(即重合部分的长度大于零),该三角形的另一边和角区的终边或始边相切,把该三角形按该位置放入该角区的过程称为一个占角动作.该三角形的第三条边的对角称为这个占角动作的插入角.

定义 4 占角动作的合角度:设占角动作所在的角区中与三角形相贴的边和与三角形相切的边的夹角为 α ,占角动作的插入角的角度为 β .则该占角动作的合角度为 $\min(\alpha, \beta) / \max(\alpha, \beta)$.

3 拟人策略

定义 5 格局:一个格局是指某个时刻平面上的一种有序对 $\langle P, Q, R \rangle$,其中 P 是组成矩形容器的 4 个三角形和已经放入矩形容器内的三角形的集合(集合中的每个元素用三角形的 3 个顶点的坐标来表示), Q 是尚未放入矩形容器内的三角形的集合(集合中的每个元素用三角形的 3 个边长来表示), R 是 P 中的三角形构成的角区的集合.

在开始时刻, P 中只含有组成矩形容器的 4 个三角形, Q 是所有待放入矩形容器内的三角形的集合, R 是 P 中组成矩形容器的 4 个三角形构成的四个角区的集合.称这时的格局为初始格局,记为 $\langle P_0, Q_0, R_0 \rangle$.从初始格局开始,从 Q 中取出第 k 个三角形放入矩形容器内后的格局称为第 k 格局,记为 $\langle P_k, Q_k, R_k \rangle$.如果在第 k 格局下, $Q_k = \emptyset$,则称该格局为终止格局.

定义 6 合理的占角动作:在到达终止格局前的第 k 格局,做一个占角动作,如果做该占角动作的三角形在矩形容器内且和 P_k 中的三角形不相交,则该占角动作称为一个合理的占角动作.

一般来说,如果盲目地做一个合理的占角动作,可能会形成新的角区,并把整个可用空间分成一些“碎块”.这样以来,不利于整体材料的充分利用,同时,在以后的计算中,增加了搜索空间,使得计算速度降低.依据人们处理实际问题的经验,有以下两种拟人策略.其主要依据原则是:充分利用“边角余料”,尽量地减小整体材料的破坏.

(1) 优先度策略.对于两个合理的占角动作 a_1, a_2 , 设两者的合角度分别为 p_1, p_2 , 如果 $p_1 < p_2$, 则称动作 a_1 比动作 a_2 具有较高的优先度.如果 $p_1 = p_2$, 则称动作 a_1 和动作 a_2 具有相同的优先度.采取这一拟人策略,是为了把三角形放入和其大小形状特征尽量“吻合”的地方.每到一个新的格局,按照优先度策略,可以把优先度不同的合理占角动作排序.选择优先度最高的合理占角动作来做,有利于充分利用空间,减少空间的浪费.

在合理的占角动作排序过程中,可能出现两个或更多的合理的占角动作具有相同的优先度.

(2) 原点策略.对于两个合理的占角动作 a_1, a_2 , 设两者所对应三角形的重心坐标分别为 $\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle$, 如果 $x_1^2 + y_1^2 < x_2^2 + y_2^2$, 则动作 a_1 比动作 a_2 优先.

如果 $x_1^2 + y_1^2 = x_2^2 + y_2^2$, 则先遇到的合理占角动作优先.

采用原点策略,是为了使得矩形容器内三角形的重心都尽可能的靠近原点.这样,有利于把三角形更紧凑地放入矩形容器内.

4 占角算法

基于以上两种拟人策略,求解三角形 Packing 问题,有下面的占角算法:

Step 1: 给定初始格局 $\langle P_0, Q_0, R_0 \rangle$, 其中 P_0 只包含组成矩形容器的 4 个三角形, Q_0 是所有待放入矩形容器内的三角形的集合, R_0 是组成矩形容器的 4 个三角形构成的四个角区的集合.格局中的角区和合理的占角动作分别按先后顺序进行排列.

(1) 若格局中没有合理的占角动作可做,报告该算法求不出问题的解.算法结束.

(2) 否则,选择优先度最高的合理的占角动作来做;若具有最高优先度的合理的占角动作不止一个,按原点策略选择优先度最高的动作来做.在 P_0 中加入做该动作的三角形的 3 个顶点的坐标,从 Q_0 中去掉做该动作的三角形的 3 个边长.将 P_0 换成 P_1 , 将 Q_0 换成 Q_1 , 将 R_0 换成由 P_1 中的三角形构成的角区的集合 R_1 .进入新的格局 $\langle P_1, Q_1, R_1 \rangle$, 转 Step 2.

Step 2: 一般地,假定在格局 $\langle P_k, Q_k, R_k \rangle$ 下, P_k 中已有 $k+4$ 个三角形, $k \geq 1$.

①若 $Q_k = \emptyset$, 则报告得到了问题的一个解.算法结束.②若 $Q_k \neq \emptyset$, 且格局中没有合理的占

角动作可做,报告该算法求不出问题的解.算法结束.③若 $Q_k \neq \emptyset$,且格局中存在合理的占角动作可做,选择优先度最高的合理的占角动作来做;若具有最高优先度的合理的占角动作不止一个,按原点策略选择优先度最高的动作来做.在 P_k 中加入做该动作的三角形的3个顶点的坐标,从 Q_k 中去除掉该动作的三角形的3个边长.将 P_k 换成 P_{k+1} ,将 Q_k 换成 Q_{k+1} ,将 R_k 换成由 P_{k+1} 中的三角形构成的角区的集合 R_{k+1} .进入新的格局 $\langle P_{k+1}, Q_{k+1}, R_{k+1} \rangle$.转Step 2.

5 实验结果分析

本算法在内存 32 M,主频为 Pentium 166Hz 的 PC 机上进行了程序实现,并对大量的实例进行了实验.结果(见图 2)表明,本算法具有可观的效率.特别是当三角形数目较多时,与手工设计及目前已经公开的算法相比,本算法在计算速度方面具有明显的优越性.另外,如果以放入三角形面积的总和与矩形面积的比值作为利用率,由计算结果还可以看出,本算法求解三角形 Packing 问题的利用率一般在 80% 以上.

图 2 是从众多实例中随机挑选的几个实例,

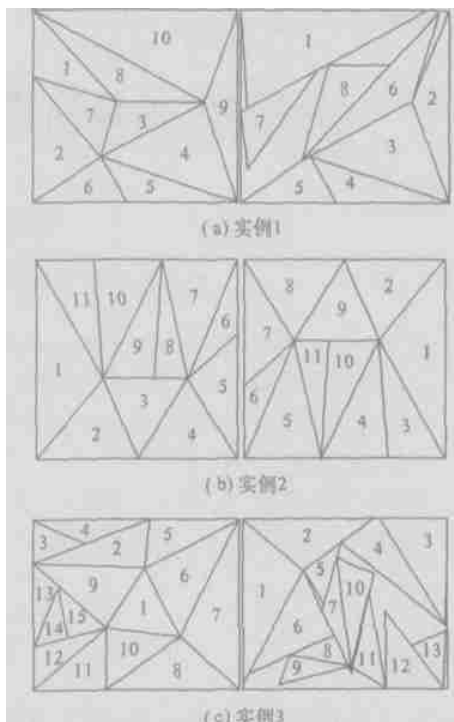


图 2 实例运算结果

Fig. 2 Results of instances

每个实例图中的左半部分,表示已知的一个手工解,右半部分表示根据本算法得到的解.图中三角形上的标号,表明三角形放入矩形容器时的顺序.

参考文献:

- [1] DOWSLAND K A, DOWSLAND W B. Packing problems [J]. European Journal of operational Research, 1992, 56 (1): 2~14.
- [2] LI Kengqin, CHENG Kam Hsi. Heuristic algorithms for on-line packing in three dimensions [J]. Journal of Algorithms, 1992, 13 (4): 589~605.
- [3] 黄文奇,李庆华,余向东.求解 Packing 问题的拟物方法 [J].应用数学学报,1986,9 (4): 443~453.
- [4] 黄文奇,许如初.支持求解圆形 Packing 问题的两个拟人策略 [J].中国科学(E 辑),1999,29 (4): 347~353.
- [5] 黄文奇,朱虹,许向阳,等.求解方格 Packing 问题的启发式算法 [J].计算机学报,1993,16 (11): 829~836.
- [6] MLENKOMIC V J. Multiple Translational Containment, Part II: Exact Algorithms [J]. Algorithmic, special issue on Computational Geometry in Manufacturing, 1997, 19 (2): 183~218.
- [7] 黄文奇,王瑞民.求解单位等边三角形 Packing 问题的占角算法 [J].鄂州大学学报,2000,7 (2): 1~3.
- [8] 黄文奇,金人超.求解 SAT 问题的拟物拟人算法—Solar [J].中国科学(E 辑),1997,27 (2): 179~186.

A Full-angle Algorithm for Solving the Triangles Packing Problem

WANG Rui-min¹, CHEN Zhuo²

(1. College of Information Engineering, Zhengzhou University, Zhengzhou 450052, China; 2. School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: The triangles packing problem is NP-hard. The complexity of their accurate algorithm is exponential. Personification approach is an which draws on analogy between the resolving problem and some social phenomena. According to personification approach, some strategies for tackling the triangles packing problem are presented and two highly efficient approximate algorithms have been developed based on these strategies. The result of our tests explains that the algorithm to solve the triangles packing problem by using personification approach is of high efficiency and low complexity.

Key words: triangles; packing problem; action value; quasi-person strategy