计算机科学 2006Vol. 33№. 5

一种求解矩形块装填问题的拟人算法*)

陈端兵　黄文奇

(华中科技大学计算机科学与技术学院　武汉 430074)

摘　要　在货物装载、木材下料、超大规模集成电路（VLSI）设计等工作中提出了矩形块装填与切割问题，对这一问题，国内外学者提出了诸如模拟退火算法、遗传算法及其它一些启发式算法等求解算法。本文利用人类的智慧和他们上万年以来形成的经验，提出了一种求解矩形块装填问题的拟人算法。该算法使用了两个主要的思想策略，即矩形块选择策略和矩形块放置策略。用本文提出的算法，对 21 个测试算例进行了实算测试，测试结果表明：算法所得装填结果的优度高，计算时间短。对这 21 个测试算例，用本文算法计算，得到了其中 16 个算例的最优解，而计算时间都在 2 秒以内。进一步的测试表明，本文提出的算法对求解矩形块装填问题十分有效。

关键词　矩形块装填与切割，拟人，占角动作

A Quasi-human Heuristic Algorithm for Rectangle Packing Problem

CHEN Duan-Bing　HUANG Wen-Qi

(College of Computer Science, Huazhong University of Science and Technology, Wuhan 430074)

Abstract　Rectangle packing/cutting problem is often raised in loading, timber cutting, Very Large Scale Integration (VLSI) design, and so on. For solving this problem, many algorithms such as simulated annealing, genetic algorithm, and other heuristic algorithms have been proposed. In this paper, an efficient quasi-human heuristic rectangle packing algorithm is proposed according to ten-thousand-year experience and wisdom of human being. The algorithm utilizes two important strategies, i. e., how to select the rectangle based on its value and how to pack it according to the corner occupying place. 21 rectangle packing test instances are tested by the produced algorithm, 16 instances of them achieve optimum solutions, and the runtime is less than 2 seconds for each instance. The integrated performance of the produced algorithm is rather satisfied. Experimental results demonstrate that the algorithm proposed in this paper is rather efficient for solving rectangle packing problem.

Keywords　Rectangle packing and cutting problem, Quasi-human heuristic, Corner-ocupying action

1　引言

对矩形块装填问题（或者说矩形块切割问题），有许多方面的应用，如火车站、码头的货物装载，超大规模集成电路的布局与布图规划，木材加工厂的木材下料等等。对这一问题，许多学者利用不同的思想提出了各种各样的算法，如：Beasley J E[1,2]，Hadjiconstantinou E 和 Christofides N[3,4]，Fekete S P 和 Schepers J[5]，Lai K K 和 Chan W M[6]，Leung T W[7]，Wang P Y[8]，Wu Y L[9]等，这些算法基本上可分为两类，即随机优化算法和确定性构造算法。对于随机优化算法，如模拟退火算法、遗传算法，其核心是设计一种能表示装填结果的编码；对于确定性的构造算法，关键在于确定一种装填规则，使得在一个可以接受的时间内求得问题的近似优化解。

1985 年，Beasley J E 用搜索树方法导出了矩形块切割问题的最优解的上界，并给出了 12 个测试算例的最优解[1]。1997 年，Lai 和 Chan 利用进化理论的方法提出了一个启发式算法，算法的目标是对面积进行优化，即最大化面积利用率，并给出了一些随机生成算例的优化结果[6]。2001 年，Leung T W 等学者用遗传算法和模拟退火算法对矩形块切割问题进行了研究[7]。2002 年，Wu Y L 等提出了基于最小自由度优先的启发式算法[9]，2005 年，Zhang Defu 等提出了一种基于分治策略的启发式算法[10]，这两种算法也是对面积进行优化。2004 年，Beasley J E 提出了一种新的基于人口进化理论的启发式矩形块切割算法，该算法对 21 个测试算例进行了实算，得到了其中 13 个算例的最优解[2]。受以上这些研究的启发，一些人把人类的经验和智慧形式化为算法而得到拟物、拟人算法[11~13]。

本文利用人类的智慧和他们上万年以来形成的经验，以最大化装填价值（即已经放入容器中所有矩形块的价值的总和）为优化目标，提出了一种矩形块装填问题的拟人算法。该算法的一个主要思想就是如何选择矩形块的问题，我们的原则是先选价值高的矩形块（或单位面积价值高的矩形块，具体的选择原则第 3 节有详细的描述），后选价值低的矩形块。算法的另一个主要思想就是如何放置矩形块的问题，我们采取占角放置的原则，即放进矩形容器的矩形块始终占据一个角，在占角的前提下，还尽量占穴，这样就使矩形块之间挨得很紧，减小了对空间的浪费，从而提高装填价值。利用本文提出的算法，对 21 个测试算例进行了实算测试，得到了其中 16 个算例的最优解，而计算时间都在 2 秒以内。测试结果表明，

*）基金项目：国家自然科学基金资助项目 10471051 和 973 项目 2004CB318000。陈端兵　博士生，主要研究方向为 NP 难问题高效求解。黄文奇　教授，博士生导师，主要研究方向为 NP 难问题高效求解，人工智能，博奕等。

本文提出的算法对求解矩形块装填问题十分有效。

2 问题描述

矩形块装填问题可描述为：已知一个矩形容器 $R_0(L_0, W_0)$，其长度为 L_0，宽度为 W_0，以及 m 类矩形块组成的集合 $\{R_i | i=1, 2, \cdots, m\}$，每一类矩形块 $R_i(l_i, w_i)$ 的长度为 l_i，宽度为 w_i，价值为 $v_i(i=1, 2, \cdots, m)$。现在的问题是，要从这 m 类矩形块中选出若干矩形块放入到矩形容器 $R_0(L_0, W_0)$ 中，使得所有放入矩形容器的矩形块的价值总和最大，并满足如下四条限制条件：

① 任一放入矩形容器的矩形块的边必须和矩形容器的边平行；

② 任意两块放入矩形容器的矩形块没有嵌入（即两矩形块重叠的面积不能大于0）；

③ 矩形块不能旋转，即，若 $l \neq w$，那么矩形块 $R(l, w)$ 和矩形块 $R'(w, l)$ 表示不同的矩形块；

④ 对每一类矩形块 R_i，放入的个数必须介于 P_i 和 Q_i 之间（P_i 和 Q_i 均为整数，且 $0 \leqslant P_i \leqslant Q_i$）。

为了便于描述，本文用 M 表示所有 Q_i 的和，即 $M=\sum_{i=1}^{m} Q_i$。

3 算法基本思想与基本策略

3.1 算法思想

在某一时刻，已经按放置规则在矩形容器中放置了若干矩形块，那么对还未放入的矩形块，选择哪一块，放置到哪一个位置好呢？我们先来看这样一个例子，假如你要进行一次环球旅行，你身边有一个可装1公斤的袋子，现在有0.8公斤黄金和0.7公斤白银，你如何用这个袋子来装这些黄金白银呢？很明显，你会用它先装0.8公斤的黄金，再装0.2公斤的白银，因为黄金比白银价值高，这种装法就使得袋子装的东西“更值钱”。在本文的矩形块装填问题中，若将矩形块单位面积的价值作为一种“价值度量标准”，那么，很自然地，我们要尽量先装单位面积价值大的矩形块。

选择好了准备放入的矩形块后，为很好地利用容器的空间，放在容器中哪一个位置也同样重要。中国很早以前就有这样一句警语：“金角银边草肚皮”，从此警语中可悟出一个道理：放进去的矩形块最好是占据某个角，其次是贴边，最差的是悬空。若放进去的矩形块不仅占据了某个角，还和形成这个角以外的某些矩形块贴边，那么这个动作就做得相当优美，与刚放入的矩形块贴边的那些矩形块就构成了一个穴，相应的放置动作可称为占穴动作。我们可将前面提到的警语发展为“金角银边草肚皮，价值最高钻石穴”。

本文提出的算法就利用了这两种思想，即先选出单位面积价值较大的矩形块，然后再向矩形容器中放，在放置矩形块时，总是占据某个角，而且动作的穴度还要尽量大，这样放置就可使矩形块之间挨得很紧，减小对空间的浪费。这样一来，就会尽可能地提高矩形容器的装填价值（即已经放入矩形容器中所有矩形块的价值总和）。

在放置过程中，有一个问题值得我们注意：在某一格局之下，对剩下的矩形块，有些类型的矩形块是必须要放入的，例如第 j 类矩形块的 $P_j=2$，而此时矩形容器中第 j 类矩形块的数量小于2，那么此时，我们就要优先挑选第 j 类矩形块放入矩形容器。

3.2 基本概念

(1)格局

在某一时刻，矩形容器 $R_0(L_0, W_0)$ 中已经互不嵌入地放了若干矩形块，还有若干矩形块在容器外面等待放入，这种状态，称为一个格局。若矩形容器中还没有放任何矩形块，此时的格局称为初始格局，若全部矩形块都已互不嵌入地放入了容器或容器中已放不下任何矩形块，此时的格局称为终止格局。

(2)占角动作

在某一格局下，若放进去的矩形块 R 与容器中已有矩形块（包括构成容器的四块矩形块）中的某两块矩形块的不同方向的边有重叠（并且重叠的长度大于0），则称放此矩形块的这种作法为一个占角动作，相应地，矩形块 R 所在位置为该矩形块的一个占角位置。若放进去的矩形块 R 与已有矩形块没有嵌入（即与其它矩形块重叠的面积等于0），并且不超出容器的边界，则称这样的占角动作为合法占角动作。在图1中，阴影矩形块是已经放置好了的矩形块，若将矩形块1放在位置A（或位置B，或位置C，或位置D），则放置矩形块1的动作就是占角动作，若放在位置E或位置F，就不是占角动作，因为若放在位置E，矩形块1只与已放进的某一块矩形块的边有重叠，而在位置F，矩形块1处于悬空状态。

(3)两矩形块间的距离

在两矩形块中各取一点 P_i, P_j，将这两点间的曼哈顿距离（即街道距离）记为 D_{ij}，所有这些距离 D_{ij} 中的最短者定义为这两块矩形块间的距离 d。例如，在图2a中，两矩形块间的距离 $d=0$，图2b中，两矩形块间的距离 $d=l$，图2c中，两矩形块间的距离 $d=l_1+l_2$。

(4)占角动作的穴度

若将某一矩形块 R_i 按占角动作放进矩形容器之后，矩形块 R_i 与形成这个角以外的其它矩形块（包括构成容器的四块矩形块）的距离的最小值为 d_i，则此占角动作的穴度 C_i 定义为：

$$C_i = 1 - \frac{d_i}{\sqrt{l_i \cdot w_i}}$$

式中，l_i 和 w_i 为矩形块 R_i 的长和宽。

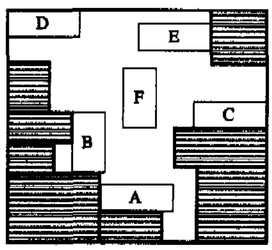

图1 占角动作

(5)占角动作的价值度

在具体选择矩形块时，只考虑矩形块的单位面积的价值是不够的，有时，矩形块的单位面积的价值不是太大，但是它的面积较大，选择这样的矩形块也许更合理。例如，在图3中，矩形容器的大小为(8,8)，阴影矩形块为已经放入容器的矩形块，剩下的空间大小为(2,8)，矩形块a，b，c的大小分别为(2,7)，(2，3)，(3,3)，单位面积的价值分别为1.5，2，2。那么此时就应该选择矩形块a放入容器，这样容器的装填价值增加了21，而选择矩形块b的话，容器的装填价值只增加了12（放了矩形块b之后，矩形块a或矩形块c就再也放不下了）。因此，在选择矩形块时，我们一方面要考虑矩形块

的单位面积的价值，同时也要考虑矩形块本身的价值大小，我们可用占角动作的价值度 V_{degree} 来刻画这一选择策略：

$$V_{degree}=\alpha.\ v+\beta.\ unitv$$

式中，v 和 $unitv$ 分别为占角动作对应矩形块的价值和其单位面积的价值；α 和 β 为对应的权重。

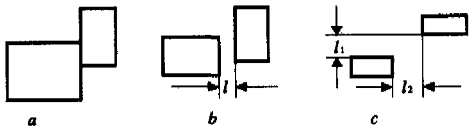

图 2　两矩形块间的距离

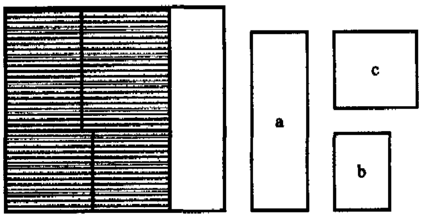

图 3　占角动作的价值度

3.3　算法描述

在当前格局之下，首先从剩下的矩形块中挑选出必须要放入容器的所有类型的矩形块，若已没有这种类型的矩形块，则选择所有类型的矩形块。对选出的矩形块，枚举所有的合法占角动作，然后作价值度最大的占角动作，若这样的占角动作有多个，就作穴度最大的占角动作，直到所有矩形块已全部放入容器或容器中已放不下任何矩形块为止。

上面这种放置办法是一个贪心放置的过程，它只顾及了“眼前利益”，为了进一步提高容器的装填价值，我们引入回溯的过程。具体的办法为，首先还是按前面贪心放置过程的办法选择矩形块，对选出的矩形块，枚举所有的合法占角动作，然后对每个占角动作进行打分，这个分数表征了这个占角动作的好坏，若分数高，说明这个动作做得好，反之，若分数低，说明这个动作做得不好，在选取具体的放置动作时，我们就选取分数最高的占角动作。

4　算法的程序实现

4.1　引算法(贪心放置过程)

步骤 1　在某一格局之下，矩形容器中已经按放置规则放入了若干矩形块，对还没有放入容器的矩形块，若没有合法占角动作可做，输出装填结果，停机；否则，从剩下的矩形块中选择必须要放入矩形容器的所有类型的矩形块，若已没有这种类型的矩形块，则选择所有类型的矩形块，对选出的矩形块，枚举所有的合法占角动作。

步骤 2　从所有的合法占角动作中，选取价值度最大的占角动作。若这样的占角动作唯一，就真实地做这个占角动作，得到新的格局，转到步骤 5；否则，转到步骤 3。

步骤 3　选取穴度最大的占角动作。若这样的占角动作唯一，就真实地做这个占角动作，得到新的格局，转到步骤 5；否则，转到步骤 4。

步骤 4　从这些占角动作中随机选择一个占角动作，并真实地做这个占角动作，得到新的格局。

步骤 5　若所有矩形块都已放入容器，输出装填结果，停机；否则，转到步骤 1。

4.2　主算法(回溯过程)

步骤 1　在某一格局之下，矩形容器中已经按放置规则放入了若干矩形块，对还没有放入容器的矩形块，若没有合法占角动作可做，输出装填结果，停机；否则，从剩下的矩形块中选择必须要放入矩形容器的所有类型的矩形块，若已没有这种类型的矩形块，则选择所有类型的矩形块，对选出的矩形块，枚举所有的合法占角动作。

步骤 2　对每一个合法占角动作，做完后即可得到一个新的格局，从此格局出发，引用算法的放置办法放置剩下的矩形块。如果能将所有矩形块全部放入，则输出装填结果，停机；否则，记录已经放入的矩形块价值之和，作为占角动作的分数。

步骤 3　选取分数最高的占角动作，若这样的占角动作唯一，就真实地做这个占角动作，得到新的格局，转到步骤 1；否则，若分数最高的占角动作有多个，则转到步骤 4。

步骤 4　选取价值度最大的占角动作。若这样的占角动作唯一，就真实地做这个占角动作，得到新的格局，转到步骤 1；否则，转到步骤 5。

步骤 5　选取穴度最大的占角动作。若这样的占角动作唯一，就真实地做这个占角动作，得到新的格局，转到步骤 1；否则，转到步骤 6。

步骤 6　从这些占角动作中随机选择一个占角动作，并真实地做这个占角动作，得到新的格局，转到步骤 1。

5　实算结果

对本文提出的算法，用 C#. net 语言编程实现，并对 21 个测试算例在 pentium 4(256M RAM/ 2.0GHz CPU)微型机上进行了计算测试。这 21 个测试算例中，有 12 个算例取自文[1]，2 个算例取自文[3]，1 个算例取自文[8]，1 个算例取自文[4]，5 个算例取自文[5]。这 21 个测试算例数据也可以从 http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/ngcutap.txt 上下载。

从实算结果来看，用本文算法求解矩形块装填问题，得到装填结果的装填价值高，计算时间短，每个测试算例的计算时间都在 2 秒以内。我们将本文算法得到的结果和文[2]中给出的 PH 算法(Population Heuristic)的计算结果进行了对比，如表 1 所示，PH 算法所用的计算机为 Silicon Graphics O2 workstation (R10000 chip, 225MHz, 128MB main memory)，表中的计算时间为相应算法所用计算机下的计算时间。从对比结果来看，本文算法计算所得结果的综合性能好于 PH 算法的结果，用本文算法计算得到了 21 个测试算例中的 16 个算例的最优解，用 PH 算法可得到 21 个算例中的 13 个算例的最优解。用本文算法计算得到的这 21 个测试算例的结果中，有 6 个算例的装填价值比 PH 算法所得结果要高，14 个算例的装填价值和 PH 算法一样，有一个算例的装填价值不如 PH 算法 (如表 1 的第 7 列和第 9 列所示)，而本文算法的计算时间比 PH 算法的计算时间要短，尤其是规模较大的算例 (如表 1 第 8 列和第 10 列所示)。

结论　在货物装载、木材下料以及超大规模集成电路(VLSI)设计等工作中，提出了矩形块装填与切割问题。本文利用人类的智慧和他们上万年以来形成的经验，以最大化装填价值为优化目标，提出了一种求解矩形块装填问题的拟人算法。该算法使用了两个主要的思想策略，即依据矩形块价值高低选择矩形块的策略和按占角位置放置矩形块的策略。

用本文提出的算法，对 21 个测试算例进行了实算测试，得到了其中 16 个算例的最优解，而计算时间都在 2 秒以内。实算测试表明，本文提出的拟人算法对求解矩形块装填问题十分有效。

表 1 本文算法和 PH 算法[2]的对比

算例来源	算例编号	问题规模				本文算法		PH[2]	
		(L_0, W_0)	m	M	最优解	最好解	计算时间(秒)	最好解	计算时间(秒)
Beasley[1]	1	(10, 10)	5	10	164	Optimal	0.07	Optimal	0.02
	2		7	17	230	Optimal	0.03	Optimal	0.16
	3		10	21	247	Optimal	0.12	Optimal	0.53
	4	(15, 10)	5	7	268	Optimal	0.02	Optimal	0.01
	5		7	14	358	Optimal	0.04	Optimal	0.11
	6		10	15	289	Optimal	0.10	Optimal	0.43
	7	(20, 20)	5	8	430	Optimal	0.08	Optimal	0.01
	8		7	13	834	Optimal	0.15	Optimal	3.25
	9		10	18	924	912	0.15	Optimal	2.18
	10	(30, 30)	5	13	1452	Optimal	0.02	Optimal	0.03
	11		7	15	1688	Optimal	0.08	Optimal	0.60
	12		10	22	1865	Optimal	0.23	1801	3.48
Hadjiconstantinou and Christofides[3]	3	(30, 30)	7	7	1178	Optimal	0.03	Optimal	0.03
	11		15	15	1270	Optimal	0.05	Optimal	0.04
Wang[8]		(70, 40)	20	42	2726	Optimal	0.18	2721	6.86
Christofides and Whitlock[4]	3	(40, 70)	20	62	1860	1800	0.52	1720	8.63
Fekete and Schepers[5]	1	(100, 100)	15	50	27718	27486	0.47	27486	19.71
	2		30	30	22502	Optimal	0.98	21976	13.19
	3		30	30	24019	23743	0.85	23743	11.46
	4		33	61	32893	Optimal	1.57	31269	32.08
	5		29	97	27923	26525	1.77	26332	83.44

参考文献

1 Beasley J E. An exact two-dimensional non-guillotine cutting tree search procedure. Operations Research, 1985, 33: 49～64

2 Beasley J E. A population heuristic for constrained two-dimensional non-guillotine cutting. European Journal of Operational Research, 2004, 156: 601～627

3 Hadjiconstantinou E, Christofides N. An exact algorithm for general, orthogonal, two-dimensional knapsack problems. European Journal of Operational Research, 1995, 83: 39～56

4 Christofides N, Whitlock C. An algorithm for two-dimensional cutting problems. Computers and Operations Research, 1977, 25: 30～44

5 Fekete S P, Schepers J. A new exact algorithm for general orthogonal d-dimensional knapsack problems. Springer Lecture Notes in Computer Science, 1997, 1284: 144～156

6 Lai K K, Chan W M. An evolutionary algorithm for the rectangular cutting stock problem. International Journal of Industrial Engineering, 1997, 4: 130～139

7 Leung T W, Yung C H, Troutt M D. Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem. Computers and Industrial Engineering, 2001, 40: 201～214

8 Wang P Y. Two algorithms for constrained two-dimensional cutting stock problems. Operations Research, 1983, 31:573～586

9 Wu Y L, Huang Wenqi, et al. An effective quasi-human based heuristic for solving the rectangle packing problem. European Journal of Operational Research, 2002, 141: 341～358

10 Zhang Defu, Deng Ansheng, Kang Yan. A hybrid heuristic algorithm for the rectangular packing problem. Lecture Notes in Computer Science (ICCS 2005), 2005, 3514: 783～791

11 Huang Wenqi, Jin Rencao. The quasi-physical and quasi-sociological algorithm solar for solving SAT problem. Science in China, Ser. E, 1997, 27(2): 179～186

12 Huang Wenqi, Xu Ruchu. Two personification strategies for solving circles packing problem. Science in China, Ser. E, 1999, 42 (6): 595～602

13 黄文奇，陈端兵. 一种求解矩形块布局问题的拟物拟人算法. 计算机科学，2005(11)