软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(5):1037−1044 [doi: 10.3724/SP.J.1001.2012.03986]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于动作空间求解二维矩形 Packing 问题的高效算法*

何 琨, 黄文奇+, 金 燕

(华中科技大学 计算机科学与技术学院,湖北 武汉 430074)

Efficient Algorithm Based on Action Space for Solving the 2D Rectangular Packing Problem

HE Kun, HUANG Wen-Qi+, JIN Yan

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

+ Corresponding author: E-mail: wqhuangwh@gmail.com

He K, Huang WQ, Jin Y. Efficient algorithm based on action space for solving the 2D rectangular packing problem. *Journal of Software*, 2012,23(5):1037−1044. http://www.jos.org.cn/1000-9825/3986.htm

Abstract: This paper addresses a typical NP-hard problem, the two-dimensional (2D) rectangular packing problem. The study makes improvements on a quasi-human approach, a caving degree algorithm proposed by Huang Wen-Qi, *et al*., by defining the conception of action space such that the calculation of the caving degree is simplified. Therefore, the evaluation on different placements is reduced considerably, and good layouts could be obtained quickly. The experiments tested 21 famous instances of the 2D rectangular packing problem provided by Hopper and Turton. The improved algorithm achieved optimal layout with a space utilization of 100% for each instance, and the average computing time on a personal computer was within seven minutes. Computational results show that the improvement strategies on the quasi-human caving degree approach are evident and effective.
Key words: NP hard; rectangular packing; quasi-human; action space; caving degree

摘 要: 对于二维矩形 Packing 这一典型的 NP 难度问题,在黄文奇等人提出的拟人型穴度算法的基础上,通过定义动作空间来简化对不同放入动作的评价,使穴度的计算时间明显缩短,从而使算法能够快速地得到空间利用率较高的布局图案.实验测试了 Hopper 和 Turton 提出的 21 个著名的二维矩形 Packing 问题的实例.改进的算法对其中的每一个实例都得到了空间利用率为 100%的最优布局,且在普通 PC 机上的平均计算时间未超过 7 分钟.实验结果表明,基于动作空间对拟人型穴度算法所进行的改进是明显而有效的.
关键词: NP 难度;矩形 Packing;拟人;动作空间;穴度
中图法分类号: TP301 **文献标识码**: A

二维矩形 Packing 问题是一类典型的 NP 难度问题[1],有着重要的理论研究价值.自 20 世纪 70 年代以来,NP 难度问题的求解一直是计算机科学领域的瓶颈任务,至今人们尚无法给出经典数学所期望的那种既完整、精确又快速、高效的求解方法.因此,寻找矩形 Packing 这一典型 NP 难度问题的最优解或近似解,对于 NP 难度问题的研究有着重要的意义,同时对于其他 NP 难度问题的求解也有借鉴和参考的价值.

* 基金项目: 国家自然科学基金(60773194)
收稿时间: 2010-09-25; 定稿时间: 2011-01-20
CNKI 网络优先出版: 2011-05-26 11:46, http://www.cnki.net/kcms/detail/11.2560.tp.20110526.1146.001.html

另外,二维矩形 Packing 问题在工业生产和现实生活中有着广泛的应用.例如,在集成电路的设计过程中,问如何布置才能使电子元件的放置达到最多;在钢板、木板、服装和玻璃等的加工过程中,问如何剪裁才能使原材料的利用率达到最大;在网页、报纸等版面的设计过程中,问如何排版才能使版面的面积利用率达到最大.

由于二维矩形Packing问题有着重要的理论意义和广泛的应用价值,近几十年来,国内外有较多的学者致力于其高效求解算法的研究.目前,代表性的算法可分为随机型算法和确定型算法两大类.其中,随机型算法着眼于设计能表示各矩形块间位置关系的编码,求解方法主要包括遗传算法[2−4]、模拟退火算法[5]、粒子群算法[6]、蚁群算法[7]等等;而确定型算法着眼于设计矩形块紧密放置的策略,求解方法主要包括 BL(bottom-left)算法[8]、BLF(bottom-left-fill)算法[9]、BLD(bottom-left-decreasing)算法[10]、分枝限界法[11]和拟物拟人法[12,13]等等.另外,不少学者采用上述两类方法相结合的方式,以进一步提高求解的优度[14,15].

在黄文奇等人提出的拟人型穴度算法[16−19]的基础上,本文改进穴度的定义,使其既反映了原有穴度概念的深刻本质又明显地简化了其计算的过程,从而使新算法能够在较短的时间内得到满意的布局图案.本文实验测试了 Hopper 和 Turton 提出的 21 个著名的算例(简称 HT 算例)[20].实验结果表明,新算法对其中的每一个算例都得到了面积利用率为 100%的最优布局,且平均计算时间仅为 409s.

我们首先给出二维矩形Packing问题的确切定义,然后介绍和分析本文的基础即拟人型穴度算法,接着给出改进的方案和详细的算法,最后进行实验分析和讨论.

1 问题描述

二维矩形 Packing 问题是指:在二维欧氏空间中,已知一个长、宽任意给定的大矩形框和有限个长、宽分别任意给定的小矩形块,要求将这些矩形块尽可能多地放入到矩形框内,使框的面积利用率达到最大.放入的块须完全在框内、边与框的边平行且任意两块间无嵌入.

图 1(a)给出了一个二维矩形 Packing 问题的实例.图 1(b)即该实例的最优布局图案,其面积利用率达到了100%.

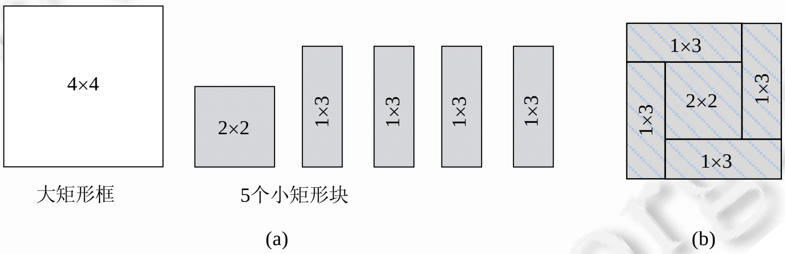


Fig.1 An instance and its optimal layout

图 1 一个问题实例及其最优布局

2 穴度算法的求解策略

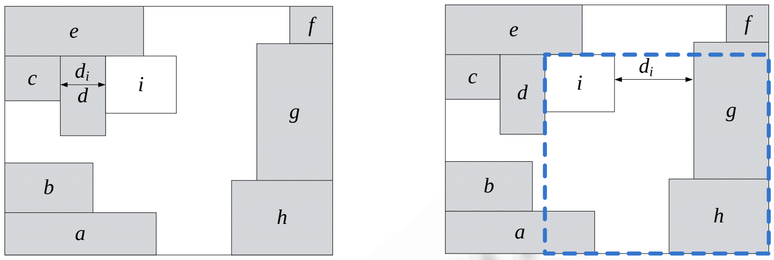
受中国古代围棋谚语“金角银边草肚皮”的启发,并发展提高到“价值最高钻石穴”,黄文奇等人提出了求解二维矩形 Packing 问题的拟人型穴度算法——HRP 算法[16]和 HA 算法[17].其主要思想是,挑选一个矩形块放入到框内的某个角区,使之与其他已放入的块或矩形框的边尽量靠拢,并定义穴度等概念来评价不同放入动作的优劣.其穴度的定义如下:

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

公式(1)中,w_i,h_i表示当前放入块的长和宽.我们可将矩形框的 4 条边看作 4 个特殊的扁平块,则 d_i 表示放入块与其他未构成动作角区的已放入块的最小距离(x 方向与 y 方向的距离之和).如图 2(a)中,d_i 为块 i 与块 a,b,c, f,g,h

及框的 4 条边的最小距离.

随后,黄文奇等人在求解三维矩形 Packing 问题的穴度算法[18]中提出了动作空间的概念.对应到二维矩形 Packing 问题,动作空间是指通过延长放入块的被贴边而在框内围成的包含放入块的那个空间.如图 2(b)中,虚线框所示即为当前占角动作对应的动作空间.在计算 d_i 时,只考虑与动作空间相嵌或相贴的矩形块和放入块的距离.如图 2(b)中,改进后的 d_i 为块 i 与块 a,g,h 及框右方与下方的边的最小距离.



(a) (b)

Fig.2 Caving degree of a corner-occupying action

图 2 占角动作的穴度

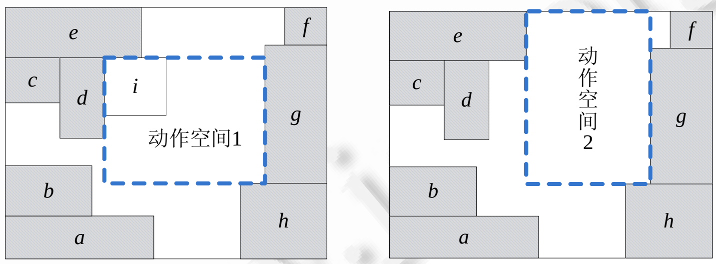
3 基于动作空间的改进策略

本文进一步改进动作空间的定义,缩小其空间的范围,以进一步简化穴度的计算.改进后的动作空间、穴度及相关概念的具体定义如下:

定义 1(格局). 设某一时刻,框内已放入若干个矩形块,还有若干个块待放,这称为一个格局.初始格局时,框内未放入任何块.当所有块都已放入框中,或框外的剩余块均无法再放入时,称为终止格局.

定义 2(动作空间). 在当前格局下,若往框内合法地放入一个虚拟的矩形块,该块的上、下、左、右 4 条边均与其他已放入的块或框的边相贴(即重合的长度大于 0),则该虚拟块所占的空间称为当前格局下的一个动作空间.

一个动作空间实际上对应着当前格局下可能放入的一个极大矩形块.图 2 中的块 i 处可能放入的最大虚拟块如图 3(a)中的动作空间 1 所示.不同的动作空间之间可以有重叠,如图 3(a)、图 3(b)中动作空间 1 与动作空间 2 所示.因此,当前的剩余空间可看作是多个动作空间的并集.



(a) (b)

Fig.3 An example of the action space

图 3 动作空间示例

改进动作空间后,在计算 d_i 时,只需考虑与动作空间相贴的块或框的边和放入块的距离.比如,图 3(a)中 d_i 为块 i 与块 g,h 的最小距离.

定义 3(占角动作). 每个动作空间的每个角均称为一个角区.在当前的格局下,若将一个矩形块放入到动作空间的一个角区,使块的顶点和角区的顶点重合,顶点对应的边分别与角区的两条边相贴,而且该放入块不超出动作空间,则称此动作为一个占角动作.

在初始格局下,框内的矩形空间为当前格局下唯一的一个动作空间.若在该空间的左下角放入一个矩形块,则产生了两个新的动作空间,而原来的动作空间将被取代,这称为该动作空间的更新.我们把当前格局下所有的动作空间放到一个动作空间链表中,每做一个占角动作,则需要相应地更新该动作对应的动作空间.在链表中去掉此动作空间,增加 0~2 个新生成的空间.

定义 4(穴度). 穴度表示占角动作与当前动作空间的紧密程度,其具体定义为一个三元组,即〈贴边数,平整度,他贴边数〉:

(1) 贴边数 k_i:表示放入块与当前动作空间相贴的边的数目,$k_i \in \{2,3,4\}$.

(2) 平整度 s_i:表示块放入后剩余空间的平整程度,其定义如公式(2)所示.其中,n_i 表示块放入后新格局下动作空间的数目.若矩形框被完全填满,则 n_i 为 0,s_i 为 1;新格局下的剩余空间越不平整,则动作空间的数目越多,n_i 的值越大,相应的 s_i 的值越小.$s_i \in (0,1)$:

$$s_i = \mathrm{e}^{-n_i} \tag{2}$$

(3) 他贴边数 p_i:表示其他已放入块及矩形框有多少条边被该动作块贴住.

可依字典序比较不同穴度的大小.穴度越大,说明放入块与当前动作空间相贴的边越多,对剩余空间的损害越小,与其他已放入块的关系越紧密,相应的占角动作越好.

4 基于动作空间的穴度算法

基于以上定义,我们设计了改进的穴度算法,包括基本的穴度算法和增强的回溯算法两部分.

4.1 基本算法

基本算法 A_0 用于快速地构造初始解和在增强算法中构造邻域.它每一步选择一个穴度最大的占角动作来做,将选中的块按照指定的方向放入到对应的角区,从而不断生成新的格局.具体描述如下:

Step 1. 在当前格局下,对每个动作空间的每个角区、框外剩余的每个块、块的不同放置方向,将块放入到角区,若构成一个占角动作,则计算其穴度.

Step 2. 依字典序按如下规则挑选一个穴度最大的占角动作来做:

(1) 穴度:大优先;

(2) 放入块的面积:大优先;

(3) 块的长边长:大优先;

(4) 块左下顶点的 x 坐标:小优先;

(5) 块左下顶点的 y 坐标:小优先;

(6) 块的方向:躺优先.

Step 3. 依如下规则更新动作空间链表,得到一个新的格局:

(1) 更新做占角动作的动作空间;

(2) 检查放入块是否与其他动作空间相嵌,并更新有嵌入的动作空间;

(3) 去掉有包含关系的多余的动作空间.

Step 4. 重复 Step 1~Step 3,直至终止格局.

在 Step 2 中,规则(2)~规则(6)用于唯一地确定一个占角动作.若有多个最大穴度的动作,则规则(2)和规则(3)指明了块的形状大小,规则(4)和规则(5)明确了块的放入位置,规则(6)确定了块的放置方向.

4.2 增强算法

从初始格局出发,增强算法 A_1 每一步考查穴度最大的前 N 个占角动作,然后通过向前看并回溯的方法最终

选择一个占角动作.其具体过程如下:

Step 1. 在当前格局下,用 A_0 的排序规则对所有的占角动作排序.

Step 2. 设置 N 的值:

$$N=\lfloor 当前格局下所有占角动作的数目\times k\%\rfloor.$$

- 若 $N<lowerbound$,则 $N=lowerbound$.
- 否则,若 $N>upperbound$,则 $N=upperbound$.

Step 3. 对前 N 个占角动作的每一个动作,执行 A_0 并将终止格局时矩形框的面积利用率记为该动作的价值度.选择价值度最大的占角动作来做,若价值度最大的动作有多个,则选择排在最前面的一个动作.

Step 4. 更新动作空间链表,得到一个新的格局.

Step 5. 重复 Step 1~Step 4,直至终止格局.

在增强算法中,我们并未考察所有的占角动作,只是按一定比例选取穴度较大的动作来进行回溯.这是因为一般而言,当前较优的动作得到较好的最终格局的可能性也较大.这种“优中选优”的策略使我们能在较短的时间内得到满意的布局图案.另外,当候选的占角动作较少时,为避免搜索的范围过小,我们对 N 的取值设定了一个下界;当候选的占角动作过多时,考虑到计算的时间,则只选择穴度较大的固定数量的动作进行回溯.

5 计算与讨论

我们将算法 A_1 用 C 语言编程,并在 1.73GHz 的微型计算机上进行了计算.在实验中,与 N 相关的 3 个参数 k,$lowerbound$,$upperbound$ 的取值分别为 25%,55,150.测试的数据为 Hopper 和 Turton 在文献[20]中提出的 21 个著名的算例,这组算例的特点是,每一个算例在理论上都有最优的布局,待放的矩形块能够全部放入到矩形框内,且框的面积利用率为 100%.

表 1 给出了 A_1 与代表性算法 HRP[16]和 HA[17]的计算结果比较.

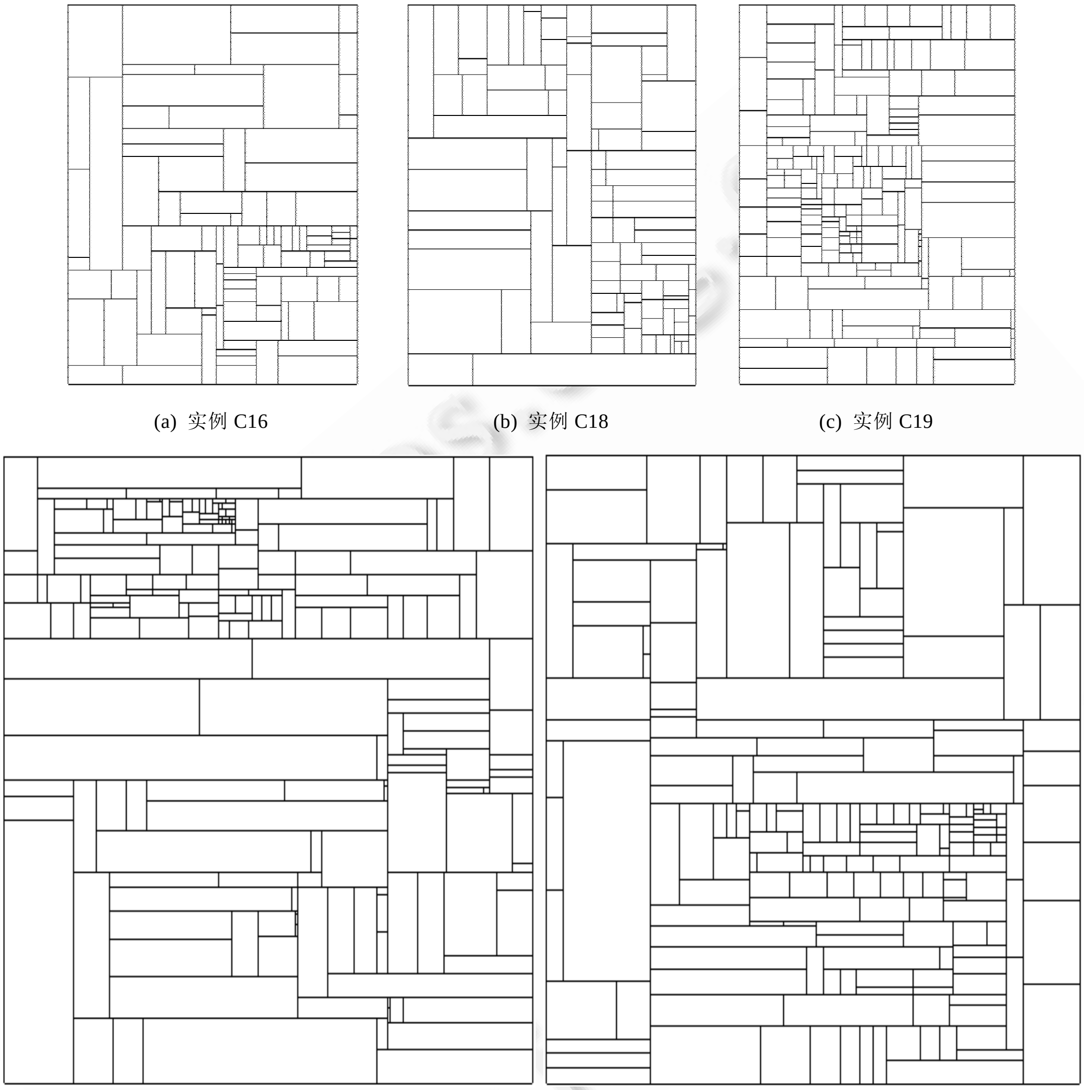
Table 1 Computing results comparison of different algorithms

表 1 相关算法的计算结果比较

实例	小矩形块的块数	大矩形框的长和宽(w×h)	HRP[16]			HA[17]			A_1		
			填充的总面积	面积利用率(%)	运行时间(s)	填充的总面积	面积利用率(%)	运行时间(s)	填充的总面积	面积利用率(%)	运行时间(s)
C1	16	20×20	400	100.00	0.05	400	100.00	0.02	400	100.00	0.02
C2	17	20×20	400	100.00	0.23	400	100.00	0.22	400	100.00	0.19
C3	16	20×20	400	100.00	1.12	400	100.00	0.04	400	100.00	0.00
C4	25	40×15	600	100.00	0.08	600	100.00	0.30	600	100.00	0.28
C5	25	40×15	600	100.00	0.10	600	100.00	0.09	600	100.00	0.05
C6	25	40×15	600	100.00	0.28	600	100.00	0.05	600	100.00	0.02
C7	28	60×30	1 800	100.00	2.58	1 800	100.00	1.16	1 800	100.00	0.50
C8	29	60×30	1 800	100.00	4.19	1 800	100.00	7.77	1 800	100.00	1.27
C9	28	60×30	1 800	100.00	2.50	1 800	100.00	2.51	1 800	100.00	2.61
C10	49	60×60	3 600	100.00	327.12	3 600	100.00	265.58	3 600	100.00	23.81
C11	49	60×60	3 600	100.00	36.59	3 600	100.00	20.13	3 600	100.00	0.17
C12	49	60×60	3 600	100.00	135.60	3 600	100.00	20.78	3 600	100.00	10.06
C13	73	60×90	5 400	100.00	55.44	5 400	100.00	72.09	5 400	100.00	4.13
C14	73	60×90	5 400	100.00	29.17	5 400	100.00	5.25	5 400	100.00	0.17
C15	73	60×90	5 400	100.00	51.13	5 400	100.00	38.34	5 400	100.00	11.66
C16	97	80×120	9 586	99.85	873.38	9 600	100.00	1 610.00	9 600	100.00	74.06
C17	97	80×120	9 600	100.00	327.61	9 600	100.00	86.29	9 600	100.00	2.92
C18	97	80×120	9 594	99.94	577.59	9 600	100.00	490.81	9 600	100.00	15.28
C19	196	160×240	38 308	99.76	4 276.82	38 385	99.96	8 303.13	38 400	100.00	2 055.00
C20	197	160×240	38 355	99.88	3 038.60	38 400	100.00	1 520.27	38 400	100.00	1.56
C21	196	160×240	38 337	99.84	3 980.65	38 350	99.87	9 288.21	38 400	100.00	6 381.91
平均			8 532.38	99.97	653.37	8 539.76	99.99	1 034.91	8 542.86	100.00	408.84

对这 21 个算例,HRP 算法得到了 16 个最优解,HA 算法得到了 19 个最优解,而 A_1 算法得到了 21 个最优解.另外,HRP 和 HA 算法是在 2GHz 的微型机上进行计算.可见,A_1 的平均计算时间比 HRP 和 HA 的都要短.因此,改进的穴度算法无论是在计算的优度还是计算的速度上都有明显的优势.

图 4 给出了 A_1 对其中的 5 个困难算例 C16,C18~C21 的布局图案,其面积利用率均为 100%,达到了最优.对这 5 个算例,HRP 算法均未得到最优布局,而 HA 算法则对其中的 C19 和 C21 未达到最优.



(a) 实例 C16 (b) 实例 C18 (c) 实例 C19

(d) 实例 C20 (e) 实例 C21

Fig.4 Packing layouts for several difficult instances

图 4 若干困难实例的布局图案

6 结束语

本文基于动作空间的定义,提出了求解二维矩形 Packing 问题的高性能穴度算法.新算法一方面继承了原有穴度算法在求解 Packing 问题上的优势,另一方面又基于动作空间明显地简化了穴度的计算.对 21 个著名算例

的计算结果表明,新算法不但提高了求解的精度,而且加快了计算的速度.对每一个算例均在较短的时间内得到了面积利用率为 100%的最优布局.在今后的工作中,我们拟进一步推广所得的成果,深入研究和改进此基于动作空间的穴度方法来求解相关的问题如二维 strip packing 问题.

References:

[1] Huang WQ, Xu RC. Introduction to the Modern Theory of Computation—Background, Foreground and Solving Method for the NP-Hard Problem. Beijing: Science Press, 2004. 47−70 (in Chinese).

[2] Hopper E, Turton B. A genetic algorithm for a 2D industrial packing problem. Computers and Industrial Engineering, 1999,37(1-2): 375−378. [doi: 10.1016/S0360-8352(99)00097-2]

[3] Kröger B. Guillotineable bin packing: A genetic approach. European Journal of Operational Research, 1995,84(3):645−661. [doi: 10.1016/0377-2217(95)00029-P]

[4] Bortfeldt A. A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. European Journal of Operational Research, 2006,172(3):814−837. [doi: 10.1016/j.ejor.2004.11.016]

[5] Leung TW, Chan CK, Troutt MD. Application of a mixed simulated annealing-genetic algorithm heuristic for the two-dimensional orthogonal packing problem. European Journal of Operational Research, 2003,145(3):530−542. [doi: 10.1016/S0377-2217(02) 00218-7]

[6] Jiang JQ, Liang YC, Shi XH, Lee HP. A hybrid algorithm based on PSO and SA and its application for two-dimensional non-guillotine cutting stock problem. Lecture Notes in Computer Science, 2004,3037:666−669. [doi: 10.1007/978-3-540-24687-9_98]

[7] Dorigo M, Blum C. Ant colony optimization theory: A survey. Theoretical Computer Science, 2005,344(2-3):243−278. [doi: 10.1016/j.tcs.2005.05.020]

[8] Baker BS, Jr Coffman EG, Rivest RL. Orthogonal packings in two dimensions. SIAM Journal on Computing, 1980,9(4):846−855. [doi: 10.1137/0209064]

[9] Chazelle B. The bottom-left bin-packing heuristic: An efficient implementation. IEEE Trans. on Computers, 1983,32(8):697−707. [doi: 10.1109/TC.1983.1676307]

[10] Hopper E. Two-Dimensional packing utilising evolutionary algorithms and other meta-heuristic methods [Ph.D. Thesis]. Cardiff: Cardiff University, 2000.

[11] Cui YD, Yang YL, Cheng X, Song PH. A recursive branch-and-bound algorithm for the rectangular guillotine strip packing problem. Computers and Operations Research, 2008,35(4):1281−1291. [doi: 10.1016/j.cor.2006.08.011]

[12] Huang WQ, Liu JF. A deterministic heuristic algorithm based on euclidian distance for solving the rectangles packing problem. Chinese Journal of Computers, 2006,29(5):734−739 (in Chinese with English abstract).

[13] Wu YL, Huang WQ, Lau SC, Wong CK, Young GH. An effective quasi-human based heuristic for solving the rectangle packing problem. European Journal of Operational Research, 2002,141(2):341−358. [doi: 10.1016/S0377-2217(02)00129-7]

[14] Zhang DF, Liu YJ, Chen SD, Xie XG. A meta-heuristic algorithm for the strip rectangular packing problem. Lecture Notes in Computer Science, 2005,3612:1235−1241. [doi: 10.1007/11539902_157]

[15] Jiang XB, Lü XQ, Liu CC. Lowest-Level left align best-fit algorithm for the 2D rectangular strip packing problem. Journal of Software, 2009,20(6):1528−1538 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3395.htm [doi: 10.3724/SP. J.1001.2009.03395]

[16] Huang WQ, Chen DB, Xu RC. A new heuristic algorithm for rectangle packing. Computers and Operations Research, 2007,34(11): 3270−3280. [doi: 10.1016/j.cor.2005.12.005]

[17] Huang WQ, Chen DB. An efficient heuristic algorithm for rectangle-packing problem. Simulation Modelling Practice and Theory, 2007,15(10):1356−1365. [doi: 10.1016/j.simpat.2007.09.004]

[18] Huang WQ, He K. A pure quasi-human algorithm for solving the cuboid packing problem. Science in China (F: Information Science), 2009,39(6):617−622 (in Chinese with English abstract).

[19] He K, Huang WQ. An efficient placement heuristic for three-dimensional rectangular packing. Computers and Operations Research, 2011,38(1):227−233. [doi: 10.1016/j.cor.2010.04.015]

[20] Hopper E, Turton BCH. An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. European Journal of Operational Research, 2001,128(1):34–57. [doi: 10.1016/S0377-2217(99)00357-4]

附中文参考文献:

[1] 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.北京:科学出版社,2004.47–70.
[12] 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.计算机学报,2006,29(5):734–739.
[15] 蒋兴波,吕肖庆,刘成城.二维矩形条带装箱问题的底部左齐择优匹配算法.软件学报,2009,20(6):1528–1538. http://www.jos.org.cn/1000-9825/3395.htm [doi: 10.3724/SP.J.1001.2009.03395]
[18] 黄文奇,何琨.求解长方体 Packing 问题的纯粹拟人算法.中国科学(F 辑:信息科学),2009,39(6):617–622.

何琨(1972－),女,北京人,博士,副教授,CCF 会员,主要研究领域为 NP 难度问题现实求解,算法优化.

金燕(1986－),女,硕士生,主要研究领域为 NP 难度问题现实求解,算法优化.

黄文奇(1938－),男,教授,博士生导师,主要研究领域为 NP 难度问题现实求解,算法优化.