基于动作空间的求解三维矩形装箱问题的穴度算法

何 琨 黄文奇 胡 骞
(华中科技大学计算机科学与技术学院 武汉 430074)

摘 要 基于拟人途径求解三维矩形装箱问题。在穴度算法的基础之上,通过定义当前格局下的极大空闲矩形空间即动作空间,使得穴度的定义既能反映其本质,同时又大能幅度地缩减计算量,从而使算法能在较短的时间内得出空间利用率较高的布局图案。试算了OR-Library中无方向约束的全部47个算例。实验结果表明,改进后的穴度算法得到的平均空间利用率为95.24%,将目前的最好结果提高了0.32%,且花费了更少的计算时间。

关键词 NP难度,三维装箱,启发式,拟人,穴度
中图法分类号 TP301 **文献标识码** A

Action Space Based Caving Degree Approach for the 3D Rectangular Packing Problem

HE Kun HUANG Wen-qi HU Qian
(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract This paper solved the three-dimensional rectangular packing problem with a quasi-human approach. By defining the maximal rectangular spaces at current iteration, the action space, we improved our caving degree approach such that the computation is largely speeded up at the same time the excellent characteristic of the caving degree is still kept. In this way a good solution could be achieved in a shorter time. In the experiments, we tested the improved algorithm with 47 without-orientation-constraint instances in the OR-Library. Computational results show an average space utilization of 95.24%, which improves current best result reported in the literature by 0.32%. In addition, the results also show less running time compared with other algorithms.

Keywords NP-hard, Three-dimensional packing, Heuristic, Quasi-human, Caving degree

1 引言

Packing问题是一类典型的NP难度问题,但在现实生活中有着许多的应用,如报纸的排版、钢板玻璃的切割、集成电路的规划设计和物流运输业中的集装箱装载等。Packing问题的高性能求解将极大地降低生产的成本,提高资源的利用率,有着重要的理论意义和现实意义。近几十年来,国内外的诸多学者对其进行了广泛而深入的研究。

本文考虑三维空间中长方体的装箱问题,即在三维欧氏空间中,给定一个长方体容器和有限个待放长方体,要求将这些长方体尽可能多地装入容器中,使得容器剩余的空闲空间最小,即容器的空间利用率最大。长方体在放入时要求满足以下3个条件:1) 完全在容器内,不与容器的壁相嵌;2) 不与任一已放入的长方体相嵌;3) 长方体的棱与容器的某条棱平行,即垂直放入长方体。目前代表性的求解方法为启发式算法,包括砌墙法[1,2]、块排列法[3-6]、分层法[7]、极大空间法[10]。部分算法进一步结合遗传[3,5,6]、禁忌[4]、模拟退火[8]、树搜索[2]、随机搜索[10]等,以改进算法的性能。

受中国古代围棋谚语"金角银边草肚皮"的启发,并将它发展提高到"价值最高钻石穴",我们曾提出了一种最大穴度的占角动作优先的拟人型求解算法——穴度算法[9]。穴度算法每一步挑选一个长方体放入到容器内的一个角区,使之与其它已放入的长方体及容器的壁尽量压紧,并通过穴度来统一评价不同占角动作的优劣。本文基于动作空间即当前格局中极大的空闲矩形空间的概念,重新定义了占角动作,并改进了穴度的定义,提出了基于动作空间的穴度算法。实验结果表明,基于动作空间的穴度算法同时具有计算精度高和计算时间短的特点。

2 定义

下面给出基于动作空间的穴度算法所涉及的主要概念。

定义1(格局) 设某一时刻,容器内已放入若干个长方体,还有若干个长方体在容器外待放,这称为一个格局。通常格局可以用已放入长方体及其位置和容器内的剩余空间来描述。容器内尚未放入任何长方体时,称为初始格局。所有长方体已放入容器,或容器外剩余的长方体无法再放入时,称为终止格局。

定义2(动作空间) 在当前格局下,往容器中合法地放

到稿日期:2009-11-20 返修日期:2010-01-22 本文受国家自然科学基金资助项目(No. 60773194)资助。
何 琨(1972—),女,博士,CCF会员,主要研究方向为NP难度问题的高性能求解算法,E-mail:brooklet60@gmail.com;黄文奇(1938—),男,教授,主要研究方向为NP难度问题的高性能求解算法,E-mail:wqhuangwh@gmail.com(通信作者);胡 骞 主要研究方向为最优化问题和启发式算法。

入一个虚拟的长方体。若该长方体的上、下、左、右、前、后6个面均与已放入的长方体或容器的壁相贴(即重合的面积大于0),则该长方体所占的空间称为当前格局下的一个动作空间。

动作空间有以下几个特点:1)剩余空闲空间中可以包含多个动作空间,我们将当前格局下的剩余空闲空间用一个动作空间列表来描述;2) 动作空间的每个面必与已放入长方体的某个面或容器的壁相贴,因此在动作空间内放入长方体,可以保证该长方体不会和其他长方体或者容器的壁相嵌;3)不同的动作空间之间既可以交叠,也可以互不相交,因此在一个动作空间内放入长方体,除了当前动作空间受到影响,其他与该长方体交叠的动作空间也会受到影响;4)如果某长方体能够放入某个动作空间内,除非该长方体和动作空间的大小完全吻合,否则放入后将产生新的动作空间(可能有多个)来取代当前动作空间。因此在算法中,我们需要维护一个动作空间列表。当放入一个长方体后,需要更新所有受到影响的动作空间,用新产生的零至多个动作空间来取代列表中的原动作空间。

可用离坐标原点最近的顶点和最远的顶点来描述一个动作空间。如图1中,$[(x_o, y_o, z_o), (x_a, y_a, z_a)]$表示初始格局下唯一的一个动作空间。若在该空间中放入一个长方体$[(x_o, y_o, z_o), (x_b, y_b, z_b)]$,则产生了3个新的动作空间$[(x_c, y_c, z_c), (x_a, y_a, z_a)]$,$[(x_d, y_d, z_d), (x_a, y_a, z_a)]$和$[(x_e, y_e, z_e), (x_a, y_a, z_a)]$,它们将取代动作空间列表中的$[(x_o, y_o, z_o), (x_a, y_a, z_a)]$。

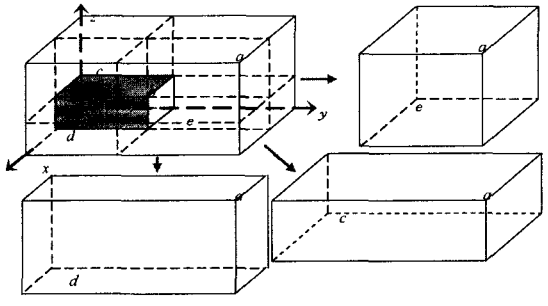

图1 动作空间

定义3(角区) 动作空间的每一个角都称为一个角区。

显然,一个动作空间有8个角区。如图2中,在动作空间$[(x_a, y_a, z_a), (x_g, y_g, z_g)]$中存在着以顶点$a, b, c, d, e, f, g, h$为角顶点的角区。

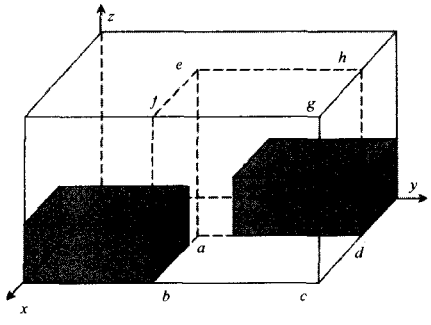

图2 角区

定义4(占角动作) 在当前格局下,若将某个长方体放入某个动作空间的某个角区,该长方体的顶点与角区的顶点重合,长方体的某3个面分别与角区的3个面相贴即重合的面积大于0,则称此放入动作为一个占角动作,称放入的长方体为占角长方体。

原穴度算法中定义的角区为已放入长方体或容器壁的某3个表面两两垂直且相交于一点而围成的空闲空间,如图2中以顶点b, c, d, g为角顶点的角区,本文将之称为狭义角区。新定义的角区包含了狭义角区且范围更广,本文将之称为广义角区。广义角区的定义使改进后的穴度算法可以考虑更多的放入动作,使搜索的空间加大,从而有利于得出更好的布局图案。

定义5(穴度) 穴度定量地描述了一个占角动作的好坏,用一个四元组〈贴面数,它贴面数,邻近度,贴面率〉表示。其中:

1)贴面数表示占角长方体有多少个面被其它已放入的长方体或容器壁贴住。将被贴面的数目记为k_i。k_i值越大,穴度越大,占角动作越好。

2)它贴面数表示占角长方体与多少个已放入长方体及容器的壁相贴,其数目记为p_i。p_i值越大,穴度越大,占角动作越好。

3)邻近度表示占角长方体与动作空间的邻近程度。用ad_i表示邻近度,ad_i的计算如下:

$$ad_i = \exp\left(\frac{-d_i}{\sqrt[3]{l_i w_i h_i}}\right), ad_i \in (0,1]$$

式中,l_i, w_i, h_i分别表示占角长方体的3边边长,d_i表示占角长方体与动作空间未贴面的最短距离。当且仅当贴面数$k_i = 6$时,$d_i = 0$,$ad_i = 1$;若$k_i < 6$,则d_i一定大于0。ad_i的取值在0到1之间。ad_i值越大,穴度越大,占角动作越好。

4)贴面率。占角长方体被贴的面积与该长方体的总表面积之比称为贴面率,记为r_i。r_i值越大,穴度越大,占角动作越好。

因此,第i个占角动作的穴度可表示为$\langle k_i, p_i, ad_i, r_i \rangle$。当需要比较两个占角动作的好坏时,可以依字典序比较两者的穴度四元组。

3 算法描述

基于动作空间的穴度算法包括基本算法和增强算法两部分。其基本思想是一块块地将容器外的长方体放入到当前格局下的一个动作空间的一个角区。从初始格局开始,每次挑选一个占角动作,该动作发生后,容器内增加一个长方体,从而形成新的格局,如此反复直到终止格局。由于动作空间是有限的,容器外长方体的数量亦是有限的,因此当前格局下所有合法的占角动作数量是有限的。

3.1 基本算法

基本算法A_0可描述如下:

Step 1 在当前格局下初始化动作空间,并将其加入动作空间列表;

Step 2 对于动作空间列表中的每一个动作空间和容器外的每一个长方体,找出所有合法的占角动作,并计算每个动作的穴度;

Step 3 选择穴度最大的占角动作,执行该动作,将相应的长方体依指定方向放入相应的角区,得到新格局(若有多个最大穴度的动作,则按动作长方体在x, y, z方向的边长和重心坐标选择一个,长方体的边长越大越好,重心坐标越小越

好),同时更新所有受到该占角动作影响的动作空间,并并用新产生的动作空间来取代原动作空间;

Step 4　重复 Step 2 和 Step 3,直到终止格局。

Step 5　输出终止格局所有已放入长方体的位置、容器的空间利用率和总的计算时间。

3.2　增强算法

对基本算法 A_0 进行回溯处理,得到增强算法 A_1:

Step 1　在当前格局下初始化动作空间,并将其加入动作空间列表;

Step 2　对于动作空间列表中的每一个动作空间和容器外的每一个长方体,找出所有合法的占角动作,并计算每个动作的穴度;

Step 3　依次试做贴面数≥3 的所有占角动作,得到一个新格局,对每一个新格局,执行算法 A_0,得到终止格局时容器的空间利用率,称为该动作的期望值;

Step 4　选择期望值最大的占角动作,执行该动作,将相应的长方体依指定方向放入相应的角区,得到新格局(若不止一个占角动作的期望值为最大期望值,则依据基本算法 A_0 的选择标准来确定要执行的占角动作),同时更新所有受到该占角动作影响的动作空间,用新产生的动作空间取代其在动作空间列表中的位置;

Step 5　重复 Step 2,Step 3 和 Step 4,直到终止格局;

Step 6　输出终止格局所有已放入长方体的位置、容器的体积利用率和总的计算时间。

这里"试做"是指每次用基本算法 A_0 计算得到该占角动作的期望值后,再将格局恢复到该占角动作之前的格局。算法每执行一个占角动作,得到了该动作按照 A_0 算法执行的最终结果,就得到了该动作对应的期望值。

4　计算与讨论

在计算和测试的过程中,我们使用 Java 语言串行编程实现了基于动作空间的穴度算法,并在配置 2.00GHz 双核 CPU 的 PC 上进行了算例计算。

4.1　一个简单算例

首先测试了一个待放长方体数目较少、较难求得最优解的算例。

该算例来自文献[9],它给定了一个尺寸为(4,4,6)的容器和尺寸为(6,3,1),(5,3,1),(2,3,1),(1,1,1)和(2,2,2)的长方体各两个。表 1 给出了基于动作空间的穴度算法的计算结果,其计算时间为 0.79s,容器的空间利用率为 100%,得到了最优解。文献[9]也得到了最优解,在 1.7GHz 的计算机上运行了 4.25s。相比较而言,改进后算法的计算速度更快。

表 1　简单算例的求解结果

序号 i	l_i,w_i,h_i	(x_{i1},y_{i1},z_{i1})	(x_{i2},y_{i2},z_{i2})
1	6,3,1	(0,0,3)	(6,3,4)
2	5,3,1	(1,0,0)	(6,1,3)
3	6,3,1	(0,1,0)	(6,4,1)
4	5,3,1	(0,3,1)	(5,4,4)
5	2,3,1	(0,0,1)	(1,3,3)
6	1,1,1	(0,0,0)	(1,1,1)
7	2,3,1	(5,1,1)	(6,4,3)
8	1,1,1	(5,3,3)	(6,4,4)
9	2,2,2	(1,1,1)	(3,3,3)
10	2,2,2	(3,1,1)	(5,3,3)

4.2　无方向约束型算例

算例库 OR-Library[11] 中为求解三维装箱问题提供了一组无方向约束的算例,共有 47 个实例。对于这组算例,算法 A_1 求解得到的容器平均空间利用率为 95.24%,平均耗时为 282.27s,如图 3 所示。

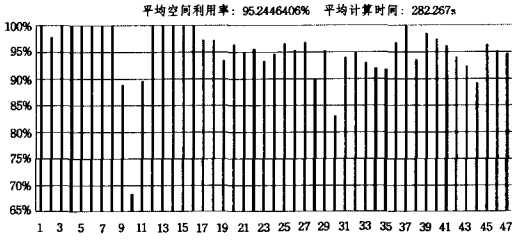

图 3　算法 A_1 对 47 个无方向约束型算例的求解结果

表 2 是算法 A_1 与其他算法之间的性能比较。算法 A_1 得到的结果最好,将目前国际学术界已发表的最好结果提高了 0.32%,并且花费了更少的计算时间。

表 2　算法性能比较

算法	平均空间利用率	计算时间	平台	编程语言
MFB_L[2]	91.00	205s	Unix Apha 600MHz	Java,串行
CDA[9]	94.92	804s	Windows 1.7GHz	Java,串行
A_1	95.24	282s	Windows 2.0GHz * 2	Java,串行

图 4 给出了这组算例中第 46 个实例的具体布局。实例 46 中容器的尺寸为(33,51,68),形状大小各不相同的 4 类待放长方体的尺寸和数量分别为(21,13,11)×25,(11,13,19)×20,(10,14,6)×20 和(8,13,5)×34。穴度算法 CDA[9] 对该算例得出了空间利用率为 94.16%的可行解。改进后的穴度算法对该算例得到了更优的布局图案,空间利用率为 95.22%,容器中可以放入 4 类长方体的数目分别为(21,13,11)×18,(11,13,19)×13,(10,14,6)×19,(8,13,5)×7。

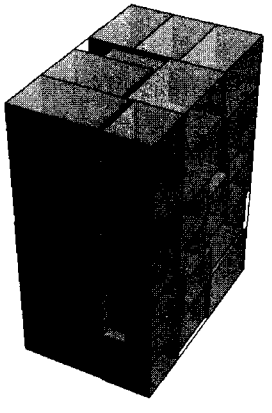

图 4　第 46 个算例的布局图案

结束语　本文提出了一个基于动作空间的求解三维矩形装箱问题的穴度算法。算法具有得出的结果好且计算时间少的特点。分析原因,一方面是由于动作空间巧妙地描述了当前格局下的剩余空闲空间,避免了在不规则的剩余空闲空间中找出所有占角动作时带来的计算量和时间开销,并且动作空间的广义角区加大了算法的搜索空间,有利于得出更好的布局图案;另一方面是由于穴度算法本身所具有的优秀特性,它形式化并且确切化地定义了穴度,每一步选择穴度最大的占角动作使装入容器的长方体尽可能紧凑,从而以尽可能装

(下转第 220 页)

基因数据集的分类能力，并且基于相容关系的基因选择方法由于避免了粗糙集离散化过程的信息丢失，提取的特征基因分类精度优于基于粗糙集的基因选择方法提取的基因。尤其选用 C5.0 作分类器时，基于相容关系的基因选择方法提取的特征基因能得到更高的分类准确率。

结束语 粗糙集不能够处理连续性数据的局限性成为它在基因表达数据研究中的主要障碍。本文给出相容度、综合相容度、相容关系和相容关系下的依赖度的定义，提出了基于相容关系的基因选择方法，并通过一个实例来加以说明。该方法由于避开了离散化过程，因此减少了信息损失，从而相对于基于粗糙集理论的基因选择方法选择的基因有更好的准确率。该方法为基因组的基因选择研究提供了一种新的尝试。

参 考 文 献

[1] Tibshirani R, Hastie T, Narashiman B, et al. Diagnosis of multiple cancer types by shrunken centroids of gene expression[C]// Nat'l Academy of Sciences. USA, 2002: 6567-6572

[2] Kohavi R, John G H. Wrappers for feature subset selection[J]. Artificial Intelligence, 1997: 273-324

[3] Banerjee M, Mitra S, Banka H. Evolutinary-rough feature selection in gene expression Data[J]. IEEE Transaction on Systems, Man, and Cybernetic, Part C: Application and Reviews, 2007, 37: 622-632

[4] Momin B F, Mitra S, Datta Gupta R. Reduct generation and classification of gene expression data[C]// Proceeding of First International Conference on Hybrid Information Technology (ICHICT06). New York, 2006: 699-708

[5] Pawlak Z. Rough sets[J]. International Journal of Information Computer Science, 1982, 11(5): 341-356

[6] Dubois D, Prade H. Putting rough sets and fuzzy sets together [J]. Intelligent Decision Support, 1992: 203-232

[7] Jensen R, Shen Q. Tolerance-based and fuzzy-rough feature selection[C]// Proceedings of the 16th International Conference on Fuzzy Systems(FUZZ-IEEE'07). 2007: 877-882

[8] Yang M, Yang P. A novel condensing tree structure for rough set feature selection[J]. Neurocomputing, 2008, 71: 1092-1100

[9] Parthaláin N M, Shen Q. Exploring the boundary region of tolerance rough sets for feature selection[J]. Pattern Recognition, 2009, 42: 655-667

[10] Yao Y Y, Zhao Y. Discernibility matrix simplification for constructing attribute reducts[J]. Information Sciences, 2009, 179: 867-882

[11] 苗夺谦，胡桂荣. 知识约简的一种启发式算法[J]. 计算机研究与发展，1999，36(6)：681-684

[12] Yang X B, Xie J, Song X N, et al. Credible rules in incomplete decision system based on descriptors[J]. Knowledge-Based Systems, 2009, 22: 8-17

[13] Shen Q, Chouchoulas A. A rough-fuzzy approach for generating classification rules[J]. Pattern Recognition, 2002, 5: 2425-2438

[14] Qian Y H, Dang C Y, Liang J Y, et al. On the evaluation of the decision performance of an incomplete decision table[J]. Data & Knowledge Engineering, 2008, 65: 373-400

[15] 王国胤. Rough 集理论与知识获取[M]. 西安：西安交通大学出版社，2001

[16] 苗夺谦. 粗糙集理论中连续属性的离散化方法[J]. 自动化学报，2001，27(3)：296-302

[17] Grzymala-Busse J W. Discretization of numerical attributes[M]. Klösgen W, Zytkow J, eds. In Handbook of Data Mining and Knowledge Discovery, Oxford University Press, 2002: 218-225

[18] Golub T R, Slonim D K, Tamayo P, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring[J]. Science, 1999, 286: 531-537

[19] Wang L P, Feng C, Xie X. Accurate cancer classification using expressions of very few genes[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2007, 4: 40-53

[20] Grzymala-Busse J W, Grzymala-Busse W J. Handling missing attribute values[M]. by Maimon O, Rokach L, ed. In Handbook of Data Mining and Knowledge Discovery, 2005: 37-57

(上接第 183 页)

入更多的长方体来提高空间的利用率。

参 考 文 献

[1] Bischoff E E, Ratcliff M S W. Issues in the development of approaches to container loading [J]. International Journal of Mana gement Science, 1995, 23(4): 377-390

[2] Lim A, Rodrigues B, Wang Y. A multi-faced buildup algorithm for three-dimensional packing problems [J]. International Journal of Management Science, 2003, 31(6): 471-481

[3] Gehring H, Bortfeldt A. A genetic algorithm for solving the container loading problem [J]. International Transactions in Operational Research, 1997, 4: 401-418

[4] Bortfeldt A, Gehring H. A tabu search algorithm for weakly heterogeneous container loading problems [J]. OR Spectrum, 1998, 20(4): 237-250

[5] Bortfeldt A, Gehring H. A hybrid genetic algorithm for the container loading problem [J]. European Journal of Operational Research, 2001, 131: 143-161

[6] Gehring H, Bortfeldt A. A parallel genetic algorithm for solving the container loading problem [J]. International Transactions in Operational Research, 2002, 9(4): 497-511

[7] Loh T H, Nee A Y C. A packing algorithm for hexahedral boxes [C]// Proceedings of the Conference of Industrial Automation. Singapore, 1992: 115-126

[8] 张德富，彭煜，朱文兴，等. 求解三维装箱问题的混合模拟退火算法 [J]. 计算机学报，2009，32(11)：2147-2156

[9] Huang Wen-Qi, He Kun. A caving degree approach for the single container loading problem [J]. European Journal of Operational Research, 2009, 196(1): 93-101

[10] Parreño F, Alvarez-Valdes R, Oliveira J F, et al. A maximal-space algorithm for the container loading problem [J]. INFORMS Journal on Computing, 2008, 20(3): 412-422

[11] Beasley J E. OR-Library: Distributing test problems by electronic mail [J]. Journal of the Operational Research Society, 1990, 41 (11): 1069-1072