

基于拟人策略的带有预放置模块的布局算法

刘景发^{1,2}, 黄文奇¹

(1. 华中科技大学计算机学院, 武汉 430074; 2. 衡阳师范学院数学系, 衡阳 421008)

摘要: 在超大规模集成电路设计中, 一些特别重要的模块, 像 RAM、ROM 和 CPU 等经常被优先放置, 而其它模块则被两两互不重叠地放置在芯片的剩余区域。此类问题能被形式化为带有预放置模块的布局问题, 关于此问题的求解方法多为随机优化方法。该文基于拟人的思想, 提出了占角和最大穴度优先的放置策略, 为该问题的快速求解提供了一种高效的启发式确定性算法。算法的高效性通过应用于标准电路 MCNC 得到了验证。

关键词: 预放置; 拟人策略; 占角动作; 穴度; 布局算法

Algorithm Based on Quasi-human Strategy for Placement Problem with Pre-placed Modules

LIU Jingfa^{1,2}, HUANG Wenqi¹

(1. College of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan 430074;
2. Department of Mathematics, Hengyang Normal University, Hengyang 421008)

【Abstract】 In very large scale integration (VLSI) design, the locations of some macro cells, such as RAM, ROM and CPU are often required to fix a priority and the other components are subject to be placed in the rest of the chip area without overlapping. These situations are formulated as the module placement with pre-placed modules. For solving the problem, the methods used are usually stochastic optimization placement algorithm. Based on the quasi-human strategy, this paper proposes the so-called corner-occupying and the largest cave degree first placement policy. An effective heuristic deterministic algorithm is presented, and the solution to this problem can be obtained quickly by applying this algorithm. Experimental results on MCNC benchmark circuits demonstrate that the algorithm is quite effective in solving the problem.

【Key words】 Pre-place; Quasi-human strategy; Corner-occupying action; Cave degree; Placement algorithm

在超大规模集成电路(VLSI)设计中, 一些特别重要的模块, 像 RAM、ROM 和 CPU 等经常被优先放置, 而其它模块则被两两互不重叠地放置在芯片的剩余区域。在印刷电路板(PCB)设计中, 也存在类似的问题。可将这类问题形式化为带有预放置模块的布局问题, 也就是说对于一些宽、高事先给定的矩形模块, 要求找到宽、高最小的矩形框, 使得可以将这些模块中的一些特殊块(称为预放置模块)按固定的方向被放置在指定的位置, 其余块(称为自由模块)被两两互不重叠地放置在矩形框内的剩余区域, 且不与预放置模块重叠。

模块布局问题已经被证明是NP难的, 于是带有预放置模块的布局问题也是NP难题。求解NP难题是当今计算机科学面临最严峻的挑战。然而, 近年来的研究表明, 对于这类问题可能根本就不存在最优的多项式算法, 于是人们提出了一些解决此类问题的近似或启发式算法。对于带有预放置模块的布局问题, 许多学者利用不同的思想提出了各种各样的算法, 被使用的方法包括基于序列对(SP)^[1]、变形网格(BSG)^[2]、角模块序列表(CBL)^[3]和B^{*}-tree^[4]等表示的随机优化的布局算法。本文提出了一种基于拟人策略^[5]的确定性启发式算法——占角穴度算法。实验证明, 该方法在解决带有预放置模块的布局问题上是有有效的。

1 问题的形式化描述

我们更形式化地描述带有预放置模块的布局问题如下: 给定一个矩形框ABCD, 以它的左下角顶点A为坐标原点, 水

平向右为x轴正向, 竖直向上为y轴正向建立直角坐标系。矩形框ABCD所围成的区域称为布局区域。假设 $S = \{b_1, b_2, \dots, b_n, b_{n+1}, b_{n+2}, \dots, b_{n+p}\}$ 是 $n+p$ 个事先给定的矩形模块的集合, 其中 b_1, b_2, \dots, b_n 是自由模块, $b_{n+1}, b_{n+2}, \dots, b_{n+p}$ 是预放置模块。每一个模块 $b_i(1 \leq i \leq n+p)$ 用一个四元组 (x_i, y_i, w_i, h_i) 来表示, 其中 x_i, y_i 表示模块 b_i 的左下角的坐标, w_i, h_i 表示模块 b_i 的宽和高。预放置模块的方位固定, 它的宽和高是常数, 不能相互交换(也就是说, 模块不能旋转)。自由模块能被放置在布局区域内的任何没有被占用的位置, 且能旋转 90° , 即自由模块在放置时, 它的宽和高能被交换。 S 的一个合法布局是指在所有预放置模块按指定的方位放好以后, 确定每一个自由模块 $b_i(1 \leq i \leq n)$ 在布局区域内的位置 (x_i, y_i) 和方向(水平或垂直), 使得任何两个不同模块之间都互不重叠。本文考虑问题的目标是找到 S 的一个边界矩形框面积最小的合法布局。

2 基本概念和放置策略

2.1 格局

一个第 k 个格局是指对于 n 个自由模块, 若在某个时刻有其中 $k(k \leq n)$ 个模块已被放进矩形框内所得到的平面上的一个

基金项目: 国家自然科学基金资助项目(10471051); 国家“973”计划基金资助项目(2004CB318000)

作者简介: 刘景发(1972-), 男, 博士生, 主研方向: 优化计算, NP 难题; 黄文奇, 教授、博导

收稿日期: 2006-02-14 **E-mail:** ljf720622@163.com

有序对 $L_k = \langle R, N \rangle$, 其中 R 是 4 个边框模块(即构成矩形框边界的 4 个特殊模块)、 p 个预放置模块和 k 个已被放入矩形框内的自由模块组成的集合; N 是尚未被放入矩形框内的 $n-k$ 个自由模块组成的集合。当 $k=0$ 时, 称 L_0 为初始格局; $k=n$ 时, 称 L_n 为终止格局; 正在处理的第 k 个格局被称为当前格局。在第 k 个格局下, 称 4 个边框模块、 p 个预放置模块和 k 个已被放入矩形框内不再移动的自由模块为固定模块。

2.2 占角动作

占角动作是由占角放置动作(corner occupying packing move)^[5]发展而来的。占角动作比占角放置动作的范围要广, 例如在图 1 中, 如果模块 M 放在位置 3 和 4, 则相应的放置动作是占角动作, 而不是文献[5]中的占角放置动作。

在当前格局下, 如果某个自由模块 M 占据了当前格局中的某一个角, 并且该模块的两边与形成此角的两固定模块的两边重叠长度均大于 0, 则称自由模块 M 做了一个占角动作。若自由模块 M 做了一个占角动作, 并且该模块与格局中的其它任一固定模块都互不重叠, 则称自由模块 M 做了一个合法占角动作, 否则称 M 做了一个非法占角动作。例如, 在图 1 中, 自由模块 M 若被放置在位置 1, 2, 3 或 4, 则 M 做了一个合法占角动作; 若被放置在位置 5, 则 M 做了一个非法占角动作; 若被放置在位置 6, 则根本没做一个占角动作。

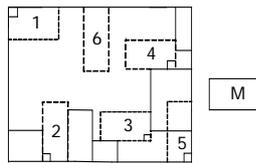


图 1 合法占角动作和非法占角动作

2.3 穴度

在当前格局下, 称

$$C_i = 1 - \frac{d_i}{w_i + h_i}$$

为自由模块 M_i 做了一个合法占角动作后的穴度, 其中 w_i, h_i 分别为 M_i 的宽和高, d_i 为 M_i 与所有已放定的固定模块(除构成角两边的模块外)之间的最小曼哈顿(Manhattan)距离(如图 2)。

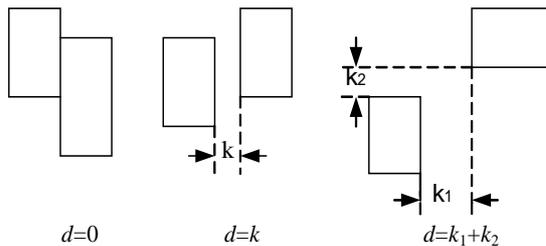


图 2 两模块之间的曼哈顿距离

显然, 一个合法占角动作的穴度不会大于 1, 若某个模块占了一个穴(即该模块不仅做了一个合法占角动作, 而且接触到除构成角两边的固定模块之外的其它固定模块), 则相应的占角动作的穴度正好等于 1。

例如, 在图 3 中, 模块 a, b, c, d, e 和矩形框的宽和高分别为 $(1, 2), (1, 1), (1, 3), (3, 1), (1, 1)$ 和 $(7, 6)$ 。模块 a, b, c, d 是固定模块, 模块 e 是矩形框外的自由模块。如果模块 e 被放在位置 1, 那么这个合法占角动作的穴度 $C_1 = 1 - 1/(1+1)/2 = 0$, 因为它与最近的模块(构成角两边的模块除外) b 或 c 的距离为 1; 如果模块 e 被放在位置 2, 则 $C_2 = 1 - 0/(1+1)/2 = 1$, 因为它占了一个穴; 类似地, 如果模块 e 被分别放在位置 3, 4, 5 或

6, 则 $C_3=0, C_4=1, C_5=-3, C_6=-2$ 。

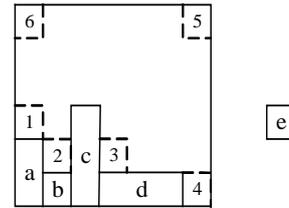


图 3 穴度的计算

2.4 放置策略

最大穴度优先的放置策略是: 在第 $k(0 \leq k \leq n-1)$ 个格局下, 对于每一个自由模块, 首先枚举所有的合法占角动作, 并计算每个合法占角动作的穴度, 然后选择穴度最大的那个合法占角动作, 将相应的模块置入矩形框内, 直到所有自由模块全部置入矩形框内, 或自由模块的个数不为 0 但没有合法占角动作做为止。如果有多个合法占角动作具有相同的最大穴度, 那么就按照下面的优先次序确定一个合法占角动作:

- (1) 选择面积最大的那个模块所对应的合法占角动作;
- (2) 选择所占角的坐标最小的那个合法占角动作, 也就是说先比较所占角的 x 坐标, 再比较 y 坐标, 谁小谁优先被占用;
- (3) 选择标号最小的模块所对应的合法占角动作。

这样的放置策略是基于以下的考虑: 选择穴度最大的合法占角动作做是为了使模块之间放得尽可能“紧凑”, 在穴度一样大的情况下, 优先挑选不容易放得下的模块, 然后注意使待放的模块尽量向左向下靠。例如, 在图 3 中, 根据最大穴度优先的放置策略, 我们优先考虑模块 e 放在位置 2。

3 占角穴度算法

占角穴度算法是占角和最大穴度优先的放置策略的一种具体实现, 主要包含以下 4 个步骤: (1) 在当前格局下, 枚举所有自由模块在矩形框内的合法占角动作; (2) 计算在做每一个合法占角动作后的穴度; (3) 按照最大穴度优先的放置策略挑选出一个合法占角动作, 并将相应的模块放入矩形框内; (4) 更新当前格局, 得到新的格局。这个过程重复进行直到所有自由模块全部置入矩形框内, 或自由模块个数不为 0 但没有合法占角动作做为止。

下面给出算法的具体描述:

输入: 所有自由模块、预放置模块以及矩形框的宽和高, 自由模块的个数 n 和预放置模块的个数 p

输出: 所有已被放进矩形框内的固定模块的坐标以及这些固定模块的个数

Begin

用给定的矩形框的宽和高初始化布局区域, 建立 4 个边框模块和预放置模块所得到的初始格局。 k 为矩形框内固定模块的个数。初始化 $k=p$;

枚举当前格局下的所有自由模块在矩形框内的合法占角动作;

while $k < n+p$ 且在当前格局下有合法占角动作

 在当前格局下计算每个合法占角动作的穴度;

 挑选出穴度最大的那些合法占角动作;

if 有多个具有相同最大穴度的合法占角动作

 根据节 2.4 中的原则(1)~(3)挑选出一个合法占角动作;

End if

 将相应的模块置入矩形框内;

 更新当前格局 L_k , 得到新的格局 L_{k+1} , 令 $k \leftarrow k+1$;

 对于新的格局下的每一个自由模块枚举所有的合法占角动作;

End while

End

注意, 当算法中的预放置模块的个数 $p=0$ 时, 就得到了

相应的无预放置模块的布局算法。

为了得到带有预放置模块的布局问题一个最优解或近似解，我们首先给定一个初始足够大的矩形框，在预放置模块按指定的方位放好以后，使得利用占角穴度算法可以把这 n 个自由模块全部互不重叠地置入矩形框内(这是可能的)，然后逐渐减少矩形框的宽和高，采用占角穴度算法放下所有自由模块，直到不能再减小矩形框为止。此时的矩形框就是面积最小的矩形框，也就是所求问题的一个最优解或近似解。

4 实验结果

我们在 Pentium 4 2.0GHz 256MB 的 RAM 的 PC 机上用 C++ 实现了占角穴度算法，为了检测算法的效率，我们用标准电路 MCNC——Apte, Xerox, Hp, Ami33 和 Ami49——作为应用实例。

首先，假设所有待放的模块都是自由模块，没有预放置的模块。对于每一个测试实例，应用无预放置模块的占角穴度算法，找到一个面积最小的布局作为目标放置。然后，从目标放置中随机地选择 1~5 块作为预放置模块，并分别运行带有预放置模块的占角穴度算法，得到尽可能小的布局面积。我们将最终的计算结果列举在表 1 中。

表 1 占角穴度算法的运行结果

电路	模块总面积 (mm ²)	目标放置面积 (mm ²)	带有预放置模块的布局面积 (mm ²)						平均面积利用率	平均面积增长率	平均时间 (sec)
			1	2	3	4	5	平均值			
Apte	46.56	46.92	46.92	46.92	46.92	46.92	46.92	46.92	99.23	0.00%	0.001
Xerox	19.32	19.79	19.79	19.79	19.79	19.91	19.91	19.84	97.38	0.25%	0.001
Hp	8.92	9.20	9.20	9.20	9.20	9.20	9.20	9.20	96.96	0.00%	0.015
Ami33	1.16	1.21	1.21	1.21	1.26	1.24	1.21	1.23	94.31	1.63%	0.252
Ami49	35.43	37.30	37.30	37.30	37.30	37.30	37.30	37.30	94.99	0.00%	0.780

表 1 中平均面积增长率表示带有预放置模块的平均布局面积相对于目标放置面积的增长率。总的来看，布局面积增长都不大，最大的为 1.63%(电路 Ami33)。特别地，对于电路 Apte、Hp、Ami49 来说，目标放置和带有预放置模块的实际布局面积相等。表中平均时间表示分别运行带有 1~5 个预放置模块的占角穴度算法的运行时间的平均值。图 4 是电路 Ami49 的一个目标放置和在此目标下带有 5 个预放置模块的实际布局结果，它们的布局面积均为 37.30mm²，面积利用率为 94.99%。图中灰色的模块为预放置模块，黑色部分为布局

结果中尚未被占用的空间。

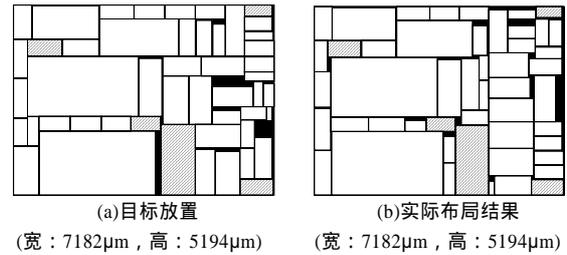


图 4 电路 Ami49 的带有预放置模块的一个布局结果

5 结论

人类在解决 Packing 问题上积累了丰富的经验。我们的工作证明，在解决复杂问题(例如 NP 难题)上，模仿人类的行为是一个求得高效算法的有效途径。对于带有预放置模块的布局问题，我们在拟人的基础上提出了一个占角穴度算法，算法采用了占角和最大穴度优先的放置策略。通过将算法应用于标准电路 MCNC 的仿真结果表明，这种算法是一种有效算法。另外，本文中提出的放置策略对于研究其它多边形 Packing 问题也将具有重要的意义。

参考文献

- Murata H, Fujiyoshi K, Kaneko M. VLSI/PCB Placement with Obstacles Based on Sequence-Pair[J]. IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems, 1998, 17(1): 60-68.
- Nakatake S, Furuya M, Kajitani Y. Module Placement on BSG-structure with Pre-placed Modules and Rectilinear Modules[C]. Proc. of ASP-DAC, 1998: 571-576.
- Dhamdhere S, Zhou Ninyu, Wang Tingchi. Module Placement with Pre-placed Modules Using the Corner Block List Representation[C]. Proc. of ISCAS, 2002: 349-352.
- Jiang Yihe, Lai Jianbang, Wang Tingchi. Module Placement with Pre-placed Modules Using the B*-tree Representation[C]. Proc. of ISCAS, 2001: 347-350.
- Wu Youliang, Huang Wenqi, Lau Siuchung, et al. An Effective Quasi-human Based Heuristic for Solving the Rectangle Packing Problem[J]. European Journal of Operational Research, 2002, (141): 341-358.

(上接第 25 页)

是最优的统计模型，有待于进一步的实验和验证。

(3) 二叉树利用方面

二叉树中左、右子树的深度有时悬殊很大，不作区分而统一处理，不仅会丢失一些有效信息，还会带来噪音。另外，本文单纯考察基于 SLM 生成的二叉树对语音停顿预测的贡献程度，对二叉树的利用比较单一，若将二叉树所提供的信息与其他方法结合起来，就能够取得更好的效果。

本文今后的工作将针对存在的问题，围绕训练语料、语言模型、以及二叉树信息利用等各个方面展开，做进一步的研究和改进。并拟结合其他方法，将二叉树信息与其他方法融合在一起，探寻更有效的语音停顿预测方法。

参考文献

- 赵 晟, 陶建华, 蔡莲红. 基于规则学习的韵律结构预测[J]. 中文信息学报, 2002, 16(5): 30-37.

- 牛正雨, 柴佩琪. 基于边界点词性特征统计的韵律短语切分[J]. 中文信息学报, 2001, 15(5): 19-25.
- 应 宏, 蔡莲红. 基于结构助词驱动韵律短语界定的研究[J]. 中文信息学报, 1999, 13(6): 41-46.
- 曹剑芬. 基于语法信息的汉语韵律结构预测[J]. 中文信息学报, 2003, 17(3): 41-46.
- 李剑锋, 胡国平, 王仁华. 基于最大熵模型的韵律短语边界预测[J]. 中文信息学报, 2004, 18(5): 56-63.
- 关 毅, 张 凯, 付国宏. 基于统计的计算语言模型[J]. 计算机应用研究, 1999, 16(6): 26-28.
- 叶竹钧. 朗读中的停顿探析[J]. 语文教学通讯, 1995, (7): 30-31.
- 汪国胜. 标点符号概说[J]. 高等函授学报(哲学社会科学版), 1999, 12(6): 19-23.
- GB/T 15834-1995 标点符号用法[S]. 1995.