

一种求解矩形块布局问题的拟物拟人算法^{*}

黄文奇 陈端兵

(华中科技大学计算机科学与技术学院 武汉 430074)

摘要 在 VLSI 工作中提出了矩形块布局问题,对这一问题,国内外学者提出了诸如模拟退火算法,遗传算法等求解算法。本文以人类上万年以来形成的经验为基础,利用“占角”和“聚类”两个拟物拟人的思想策略,提出了基于最大穴度优先的拟物拟人布局算法。用本文提出的算法,对 MCNC、GSRC 两个典型测试算例的所有实例进行了实算测试,测试结果表明:计算所得布局结果的穴度高,计算时间短。对 MCNC 和 GSRC 测试算例,除 apte 实例外,其它所有实例均得到了最优解,而计算时间都在 10 秒以内。与 CBL 算法、遗传算法和号称当今最好的 CompaSS 算法相比,本文算法所得结果的穴度更高,计算时间更短。进一步的测试表明,本文提出的拟物拟人布局算法为当今的一种高效算法。

关键词 Packing, VLSI 布局规划,拟物拟人算法,占角动作,聚类

An Efficient Quasi-physical and Quasi-human Block-packing Algorithm

HUANG Wen-Qi CHEN Duan-Bing

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

Abstract A block-packing problem is raised in VLSI design. Many algorithms such as simulation annealing and genetic algorithm have been proposed to solve this problem. According to ten-thousand-year experience of human beings and two important quasi-physical and quasi-human strategies, i.e., corner-occupying and clustering, an efficient quasi-physical and quasi-human block-packing algorithm based on maximum cave degree first is proposed in this paper. Some benchmarks, such as MCNC and GSRC, are tested by the proposed algorithm, and high quality results are achieved in short runtime. All MCNC benchmarks, except apte, and all GSRC benchmarks can be packed with zero dead-space, while the runtime is less than 10 seconds. As compared with CBL, genetic algorithm and CompaSS, reputed to be the best block-packing algorithm up to now, the results are much better both in less dead-space and less runtime. Results of further experiments demonstrate that the algorithm proposed in this paper is a highly efficient algorithm nowadays.

Keywords Packing, VLSI floorplanning, Quasi-physical and quasi-human algorithm, Corner-occupying action, Clustering

1 引言

对于能否将若干矩形块互不嵌入地放进一个矩形容器中这一问题,有许多方面的应用,如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。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,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^[1],其核心是设计一种能表示布局结果的编码,如 B*-tree^[2], O-tree^[3], TCG(transitive closure graph)^[4], CS(corner sequence)^[5]和 CBL(corner block list)^[6]等编码表示;对于确定性的构造算法,如分支限界算法^[7,12]、基于最小自由度优先的算法^[8],关键在于确定一种放置规则,使得在一个可以接受的时间内求得问题的近似优化解。J. Cong 等学者在 2004 年提出了 ZDS(zero-dead-space)算法^[9],利用该算法计算得到了 MCNC 五个实例中的两个和 GSRC 所有实例的最优解。对确定性的构造算法,计算时间一般较短,但布局结果的穴度不是太高,为此,一些学者把人类的经验和智慧形式化为算法而得到拟物、拟人算法^[8,10,11],这样,算法不仅执行速度快,计算结果的穴度也较高。

本文利用人类上万年以来形成的经验,以面积最小化为优化目标,提出一种基于最大穴度优先的拟物拟人布局算法。该算法的一个主要思想就是“占角”的思想,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,从而提高布局的穴度;算法的另一个主要思想就是“聚类”的思想,即先将面积相等或相近的若干矩形块组合成更大的矩形块,这样就可以减少矩形块的数量,降低问题的规模,从而提高算法的执行速度。利用本文提出的拟物拟人布局算法对 MCNC 和 GSRC 的所有实例进行了实算测试,除 apte 实例外,其它所有实例均得到了最优解,而计算时间都在 10 秒以内。测试结果表明,本文提出的算法对求解矩形块布局问题十分有效。与 CBL 算法^[6]、遗传算法^[1]和号称当今最好的 CompaSS 算法^[7,12]相比,本文算法计算所得布局结果的穴度更高,计算时间更短。进一步的测试表明,本文提出的拟物拟人布局算法为当今的一种高效布局算法。

2 问题描述

矩形块布局问题可描述为:有一个由 n 块矩形块组成的集合 $A = \{R_1, R_2, \dots, R_n\}$, 其中,矩形块 $R_i (i = 1, 2, 3, \dots, n)$

^{*} 基金项目:国家自然科学基金资助项目 10471051。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,博弈等。陈端兵 博士生,主要研究方向为 NP 难问题高效求解。

的面积给定,高度与宽度可在一定范围内变化。对矩形块 R_i ,可用一个四元组 (S_i, r_i, w_i, h_i) 来表示,其中, S_i 为矩形块的面积, w_i 为矩形块的宽度, h_i 为矩形块的高度, r_i 为矩形块的高度与宽度的比值($r_{\min} \leq r_i \leq r_{\max}$)。现在的问题是,要将这 n 块矩形块按照如下规则全部放置到坐标平面 $o-xy$ 上(x 轴的正向向右, y 轴的正向向上):

① 矩形块的边分别与坐标轴平行;

② 任意两块矩形块互不嵌入(即两矩形块重叠的面积不能大于0)

使得这 n 块矩形块形成的包围矩形的面积尽可能小。

这个问题,紧密地联系着如下的相关问题:任意给定 n 块矩形块 $R_1, R_2, \dots, R_n$ 和一个矩形容器 R_0 (宽为 w_0 ,高为 h_0), R_0 的左下角和坐标原点重合。问能否将这 n 块矩形块放进矩形容器 R_0 中,使得任意两块矩形块互不嵌入,任意一块矩形块 $R_i(i=1, 2, 3, \dots, n)$ 的边和矩形容器 R_0 的边都平行。以后均采用问题的此种相关描述形式。

3 算法基本思想与基本策略

3.1 算法的基本思想

算法的第一个主要思想就是“占角”的思想。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?有这样一句警句:“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”,本文提出的算法就是利用了这种思想,即每次向矩形容器中放置矩形块时,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,减小了对空间的浪费,从而提高了面积的利用率。

算法的第二个思想就是“聚类”的思想。《易经》中说:“物以类聚,人以群分”,这句话的意思是说性质相近的事物往往聚集在一起。对矩形块布局问题中的矩形块,若以面积作为它的一个性质,那么我们很自然地想到,面积相等或相近的矩形块最好能凑在一起,经组合后形成一块更大的矩形块(由于矩形块的高度与宽度之比值可在一定范围内变化,因而这种组合是可行的),从而减小了问题的规模。比如说原来的布局问题有30块矩形块,若将其中的4块面积相近的矩形块组合成一块更大的矩形块(用组合后的那块矩形块代替组合前的4块矩形块),原布局问题就转化为只有27块矩形块的布局问题。经“聚类”处理后,不仅可以提高算法的速度,在一定程度上还可提高布局结果的优劣。

3.2 基本概念

(1)超级矩形块 选择面积最接近且面积相差不大的四块矩形块,将面积最大的矩形块放在左下角,面积次大的矩形块放在右下角,面积最小的矩形块放在右上角,面积次小的矩形块放在左上角而组成一块超级矩形块。超级矩形块的面积等于组成它的四块矩形块的面积之和,高度与宽度之比值的最大值和最小值根据构成此超级矩形块的四块矩形块的高度与宽度之比值的最大值和最小值确定。在构成超级矩形块的四块矩形块中,面积最小的矩形块的面积与面积最大的矩形

块的面积之比值不能低于某一阈值,这个阈值可根据问题规模 and 实际需要而取不同的值,一般可在0.7~0.9之间取值。例如,在图1a中,矩形块A, B, C和D四块矩形块组成一块超级矩形块,若矩形块A, B, C和D的面积分别为10, 9.5, 9, 8.5,高度与宽度之比值的的最小值为0.5,最大值为2,那么由矩形块A, B, C和D组成的超级矩形块的面积为37,高度与宽度之比值的的最小值为0.544,最大值为1.848。在组成超级矩形块的四块矩形块中,其本身也可以是超级矩形块,例如,在图1b中,矩形块A, B, C和D组成一块超级矩形块,而矩形块A(图中阴影矩形块)本身是由四块矩形块组成的超级矩形块。

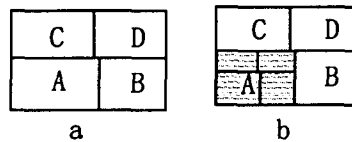


图1 超级矩形块

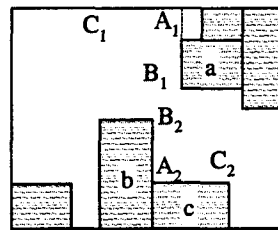


图2 角区

(2)格局 在某一时刻,矩形容器 R_0 中已经互不嵌入地放了 k 块矩形块,还有 $n-k$ 块矩形块在容器外面,这种状态,称为一个格局。若 $k=0$,即容器中还没有放任何矩形块,此时的格局称为初始格局,若 $k=n$,即全部矩形块都互不嵌入地放进了容器,此时的格局称为终止格局。

(3)角区 若矩形容器中已经互不嵌入地放了若干矩形块,则其中某两块矩形块(包括容器,可将容器看成由四块矩形块组成)的不同方向的两条边所形成的角形空旷区域称为角区。例如,在图2中,矩形块a的左面一条边和矩形容器上面一条边形成角区 $B_1 A_1 C_1$,矩形块b的右面一条边和矩形块c的上面一条边形成角区 $B_2 A_2 C_2$ 。

(4)占角动作 在某一格局下,若放进去的矩形块 R 与已有矩形块(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),即矩形块 R 占据了由这两块矩形块形成的一个角区,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3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的动作就为占角动作,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。

(5)占穴动作 若放进去的矩形块不仅占角,而且还与形成这个角以外的某些矩形块贴边,那么与此矩形块贴边的那

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。例如,在图4中,阴影矩形块是已经放进去的矩形块,矩形块A占据了由矩形块a和矩形块b形成的角,与此同时,矩形块A还与矩形块c贴边,因此,矩形块A就占据了一个由矩形块a、b和c构成的穴,放置矩形块A的这个动作就是一个占穴动作。

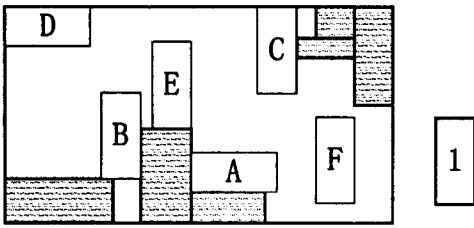


图3 占角动作

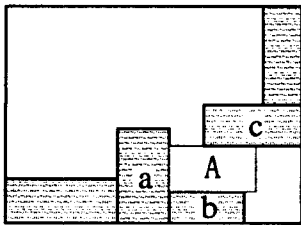


图4 占穴动作

(6)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5a中,两矩形块间的距离 $d=0$,图5b中,两矩形块间的距离 $d=l$,图5c中,两矩形块间的距离 $d=l_1+l_2$ 。

(7)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 H_i 定义为:

$$H_i = 1 - \frac{d_i}{d_i + \sqrt{S_i}}$$

式中, S_i 为矩形块 R_i 的面积。

(8)点的优先序 设坐标平面上有两点 $P_1(x_1, y_1)$ 和 $P_2(x_2, y_2)$,若 $x_1 < x_2$,那么点 P_1 的优先序大于点 P_2 的优先序;若 $x_1 = x_2$ 且 $y_1 < y_2$,那么点 P_1 的优先序大于点 P_2 的优先序。

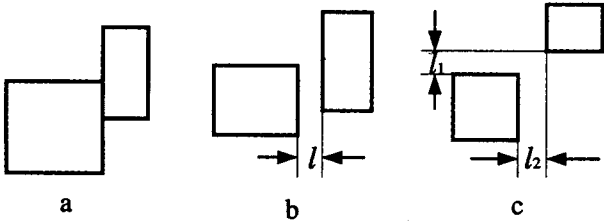


图5 两矩形块间的距离

3.3 算法的基本框架

算法的总体思想和策略如下:首先将面积相等或相近的四块矩形块组合而得到超级矩形块,然后将矩形块(或超级矩形块)按照某种次序一块一块地互不嵌入地向矩形容器中放,直到将所有矩形块(或超级矩形块)都放下而成功停机,或矩

形容器中已放不下任何矩形块(或超级矩形块)而失败停机。

具体放置的方法为,首先在所有的角区进行试放,然后对这个放置动作进行打分(惩罚分数),这个惩罚分数表征了这个动作的好坏,若惩罚分数高,说明这个动作做得不好,反之,若惩罚分数低,说明这个动作做得好,在选取具体的放置动作时,我们就选取惩罚分数最低的占角动作。算法的基本框架如图6所示。

4 算法

4.1 引算法

步骤(1)输入 N 块矩形块的面积 S_i ,矩形块高度与宽度之比的最小值 $r_{\min}^i$ 和最大值 $r_{\max}^i$ ($i=1, 2, \dots, N$)。

步骤(2)选择面积最接近,且最小矩形块的面积与最大矩形块的面积之比值不低于给定阈值(本文取0.7)的四块矩形块,进行组合而得到超级矩形块,并用超级矩形块代替用于组成它的四块矩形块。设经过这样组合处理后的矩形块(超级矩形块与最初的矩形块同等看待)的数量为 n ,对 n 块矩形块进行编号 $1, 2, \dots, n$ 。对组合处理后每一矩形块 R_i ,其面积仍记为 S_i ,高度与宽度之比的最小值和最大值仍分别记为 $r_{\min}^i$ 和 $r_{\max}^i$ ($i=1, 2, \dots, n$)。

步骤(3)进行初始化得到初始格局,令 $k=0$,这里 k 表示已经互不嵌入地放入矩形容器中的矩形块的数目。

步骤(4)在某一时刻,矩形容器中已经按放置规则放入了 k 块矩形块,对还没有放入容器的 $n-k$ 块矩形块,若没有合法占角动作可做,失败停机;否则,对每一未放入的矩形块,在每一个角区枚举穴度最大的合法占角动作。

步骤(5)从所有的合法占角动作中,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, $k=k+1$,转到步骤(8);否则,转到步骤(6)。

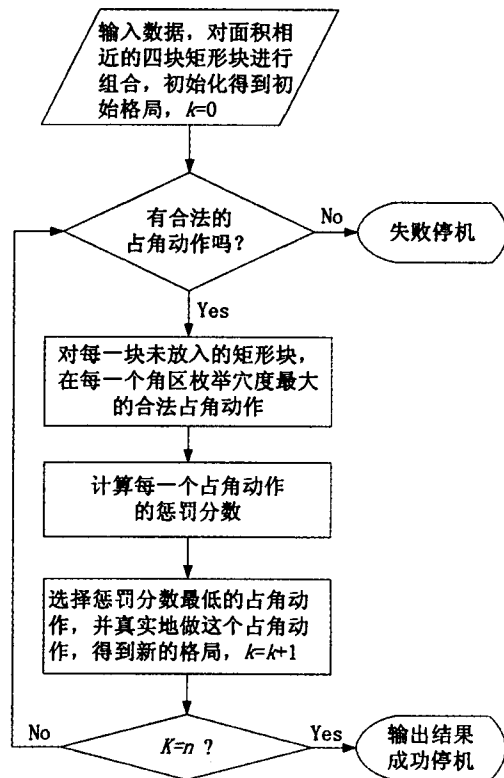


图6 算法的基本框架

步骤(6)选取做占角动作所用矩形块的左下角优先序最

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, $k=k+1$,转到步骤(8);否则,转到步骤(7)。

步骤(7)选取做占角动作所用矩形块的编号最小的占角动作,并真实地做这个占角动作,得到新的格局, $k=k+1$ 。

步骤(8)若 $k=n$,成功停机;否则,转到步骤(4)。

4.2 主算法

步骤(1)~步骤(3)分别对应于引算法中的步骤(1)~步骤(3)。

步骤(4)在某一时刻,矩形容器中已经按放置规则放入了 k 块矩形块,对还没有放入矩形容器的 $n-k$ 块矩形块,若没有合法占角动作可做,失败停机;否则,对每一块还未放入的矩形块,在每一个角区枚举穴度最大的合法占角动作。

步骤(5)对每一个合法占角动作,做完后即可得到一个格局,从此格局出发,用引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,则输出布局结果,成功停机;否则,记录没有放入的矩形块面积之和,作为占角动作的惩罚分数。

步骤(6)选取惩罚分数最低的占角动作,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, $k=k+1$,转到步骤(4);否则,若惩罚分数最低的占角动作有多个,则转到步骤(7)。

步骤(7)选取做占角动作所用矩形块的左下角优先序最大的占角动作,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, $k=k+1$,转到步骤(4);否则,转到步骤(8)。

步骤(8)选取做占角动作所用矩形块的编号最小的占角动作,并真实地做这个占角动作,得到新的格局, $k=k+1$,转到步骤(4)。

5 实算结果

对本文提出的算法,用 C#.net 语言编程实现,并对 MCNC, GSRC 等典型测试算例在 pentium 4(256M/2.0GHz)微型机上进行了计算测试。MCNC 算例的数据可在 <http://www.cse.ucsc.edu/research/surf/GSRC/MCNCbench.html> 上下载,GSRC 算例数据可在 <http://www.cse.ucsc.edu/research/surf/GSRC/GSRCbench.html> 上下载。算例中矩形块的高度与宽度之比的最小值为 0.5,最大值为 2.0。

从实算结果来看,用本文算法求解矩形块布局问题,得到布局结果的面积利用率高,计算时间短,因此,本文提出的算法是一高效的矩形块布局算法。我们用本文提出的算法对 MCNC 和 GSRC 的所有实例进行了实算,如表 1 所示。除 apte 实例的布局结果的面积未利用率为 0.01%外,其它实例均得到了最优解,即面积未利用率为 0.00%,计算时间都在 10 秒以内。图 7 和图 8 给出了我们对 MCNC 和 GSRC 算例中部分实例计算得出的布局结果图。

表 2 给出了用本文算法和 CBL 算法^[6], CompaSS 算法^[7,12]以及遗传算法^[1]求解 MCNC 和 GSRC 算例所得结果的对比情况,CompaSS 算法的计算结果可在 <http://vlsicad.eecs.umich.edu/BK/CompaSS/results/mcnc-soft.html> 和 <http://vlsicad.eecs.umich.edu/BK/CompaSS/results/gsrc-soft.html> 上查到。CBL 算法所用的计算机为 Sun Sparc 20 workstation, CompaSS 算法所用的计算机为 2.8GHz pentium 4 Linux workstation,表中的计算时间均为作者所用机器下的计算时间。从表 2 可看出,用本文算法计算所得结果略好于 CompaSS 算法(表中 1%第 2~5 列所示),对 apte 算例,CompaSS 算法计算所得布局结果的面积未利用率为 0.75%,本文算法计算所得布局结果的面积未利用率只有 0.01%,这个结果已经很接近最优解了,对 MCNC 和 GSRC 的其它实例,本文算法和 CompaSS 算法都可得到问题的最优解,但本文算法的计算时间比 CompaSS 算法的计算时间短(表中第 3 列和第 5 列所示)。从表 2 还可看出,本文算法计算所得布局结果明显好于 CBL 算法和遗传算法(表中第 2 列,第 6 列和第 8 列所示),而计算时间比 CBL 算法和遗传算法短得多(表中第 3 列,第 7 列和第 9 列所示)。

结论 在 VLSI 设计工作中,提出了矩形块布局的问题。本文以人类社会上万年以来形成的经验为基础,利用“占角”和“聚类”两个拟物拟人的思想策略,以面积最小化为优化目标,提出了基于最大穴度优先的拟物拟人布局算法来求解这一问题。用本文所述算法计算得到的布局结果的优度高,计算时间短。对 MCNC 和 GSRC 的所有实例,除 apte 实例的布局结果的面积未利用率为 0.01%外,其它实例均得到了最优解,即面积未利用率为 0.00%,而计算时间都在 10 秒以内。测试结果表明,本文提出的拟物拟人布局算法为当今的一种高效布局算法。

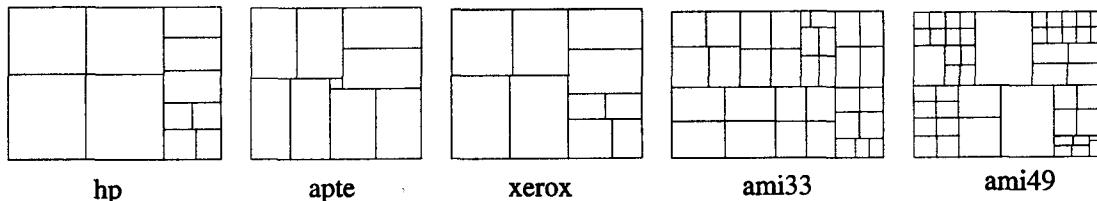


图 7 MCNC 算例的布局图

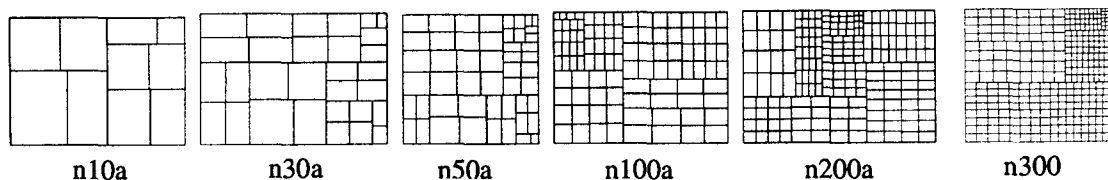


图 8 GSRC 部分实例的布局图

表 1 MCNC 和 GSRC 算例的实算结果

算例	矩形块数 量	矩形块的总面 积	容器面积(w-h)	面积未利用率 (%)	计算时间 (sec)
apte	9	46561628	46566025.49(7300.59999-6378.38336)	0.01	0.15
xerox	10	19350296	19350296.00(5000.00000-3870.05920)	0	0.17
hp	11	8830584	8830584.00(3532.23360-2500.00000)	0	0.28
ami33	33	1156449	1156449.01(1299.72429-889.76487)	0	1.49
ami49	49	35445424	35445424.01(6704.04589-5287.16906)	0	2.63
n10a	10	221679	221679.00(554.19750-400.00000)	0	0.19
n10b	10	221177	221177.00(552.94250-400.00000)	0	0.28
n10c	10	228770	228770.00(580.73066-393.93478)	0	0.03
n30a	30	208591	208591.02(549.34788-379.70660)	0	0.70
n30b	30	197781	197781.98(519.11000-381.00206)	0	0.68
n30c	30	222522	222522.28(518.27592-429.35099)	0	0.34
n50a	50	198579	198579.35(459.72404-431.95337)	0	1.33
n50b	50	203053	203053.12(598.53518-339.25010)	0	2.33
n50c	50	201512	201512.04(549.04770-367.02100)	0	1.88
n100a	100	179501	179501.01(491.50445-365.20729)	0	0.54
n100b	100	160126	160126.06(459.66900-348.35080)	0	1.01
n100c	100	171966	171966.58(503.80000-341.33899)	0	0.36
n200a	200	175696	175696.03(509.46100-344.86650)	0	8.42
n200b	200	174593	174593.72(450.34696-387.68713)	0	0.90
n200c	200	170129	170129.54(548.90235-309.94501)	0	4.42
n300	300	273170	273170.09(547.31242-499.11166)	0	0.18

表 2 本文算法和其它算法的结果对比

算例	本文算法		CompaSS ^[7,12]		CBL ^[6]		遗传算法 ^[1]	
	面积未利用率 (%)	计算时间 (sec)	面积未利用率 (%)	计算时间 (sec)	面积未利用率 (%)	计算时间 (sec)	面积未利用率 (%)	计算时间 (sec)
apte	0.01	0.15	0.75	5.99	3.4	78	2	29
xerox	0	0.17	0	0.73	2.1	76	1.35	30
hp	0	0.28	0	1.82	2.0	75	0.26	30
ami33	0	1.49	0	1.44	1.7	87	NR	NR
ami49	0	2.63	0	2.43	1.8	179	4.2	311
n100a	0	0.54	0	8.75	NR*	NR	NR	NR
n200a	0	8.42	0	30.64	NR	NR	NR	NR
n300	0	0.18	0	14.64	NR	NR	NR	NR

* NR: 原文献中未报告对应实例的结果。

参 考 文 献

- 王小港, 姚林声, 甘骏人. 一种有效的 VLSI 布图规划算法. 微处理机, 2002, 1: 4~7
- Chang Y C, Chang Yaowen, et al. B*-trees: A new representation for non-slicing floorplans. DAC'2000, Los Angeles, California: 458~463
- Guo P N, Cheng C K, Yoshimura T. An O-tree representation of non-slicing floorplan and its application. In: ACM/IEEE Design Automation Conf. 1999, 268~273
- Lin Jaiming, Chang Yaowen. TCG-S: Orthogonal coupling of P*-admissible representations for general floorplans. DAC'2002, 842~847
- Lin Jaiming, Chang Yaowen, Lin S P. Corner sequence-a P-admissible floorplan representation with a worst case linear-time packing scheme. IEEE'2003, 679~686
- Hong Xianlong, Dong Sheqin, et al. Corner block list representation and its application to floorplan optimization. IEEE Transactions on Circuits and Systems-II: Express Briefs, 2004, 51(5): 228~233
- Chan H H, Markov I L. Practical slicing and non-slicing block-packing without simulated annealing. ACM/GLSVLSI' 2004, Boston Massachusetts USA, April 2004
- 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
- Cong J, Nataneli G, Romesis M, Shinnerl J R. An area-optimality study of floorplanning. ACM/ISPD, Phoenix, Arizona, USA. April 2004, 78~83
- 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
- Huang Wenqi, Xu Ruchu. Two personification strategies for solving circles packing problem. Science in China, Ser. E, 1999, 42(6): 595~602
- Chan H H, Markov I L. <http://vlsicad.eecs.umich.edu/BK/CompaSS/>