

# 基于 GA-SA 混合算法的 VLSI 门阵列布局设计

操礼程, 杨依忠, 范海秋, 解光军

(合肥工业大学理学院, 合肥 230009)

**摘要:** 布局是 VLSI 布图设计中的关键环节, 通常采用随机优化算法。该文采用遗传算法(GA)与模拟退火法(SA)相结合的搜索算法实现 VLSI 门阵列模式布局, 利用遗传算法进行全局搜索, 模拟退火法进行局部搜索。进化过程中采用精英保留策略, 并对进化结果进行有选择的模拟退火操作, 这样既加强了局部搜索能力又防止陷入局部最优。在复合布局目标函数中引入对最长线网的惩罚, 其收敛速度比以总线长度为单一目标函数的要快。在交叉操作中, 对交叉位置的选择采用了一种新的策略, 增加了交叉的有效性。实验表明, 此算法与简单遗传算法相比, 有效地提高了全局搜索能力。

**关键词:** VLSI 布局; 遗传算法; 模拟退火法

## VLSI Placement Design Based on Genetic Algorithm and Simulated Annealing Algorithm

CAO Licheng, YANG Yizhong, FAN Haiqiu, XIE Guangjun

(School of Science, Hefei University of Technology, Hefei 230009)

**【Abstract】** A new algorithm for regular ICs' placement problem is presented. It is resulted from the combination of the genetic algorithm and simulated annealing algorithm, the genetic algorithm is served as the main flow of the new algorithm for global search, simulated annealing algorithm adjusts the optimization population to avoid trapping in the local optimum. In the genetic algorithm, a new crossover operator involved Gauss probability to determine the crossover slot is described. The new algorithm has been run on a variety of test instances; the result is superior to that of simple GA.

**【Key words】** VLSI placement; Genetic algorithm; Simulated annealing

随着微电子工艺进入甚深亚微米(VDSM)层次, 物理设计对于集成电路设计日趋重要。布局作为物理设计的第1步, 是一种组合优化问题, 这类问题已被证明是 NP 问题。

针对布局问题的困难, 已发表了许多不同算法。其典型代表是基于最小割的图划分算法, 像 Kernighan<sup>[1]</sup> 和 Fiduccia<sup>[2]</sup> 等的最小割算法是依赖于初始划分的迭代优化算法, 其在每次迭代的过程是贪心的, 这类算法的最大缺点是局限于局部最优解。若把基于划分的算法看成是自上而下的, 那么与之相反的则是基于结群的算法。这类算法是根据单元间的联结度把单元组合成一个个群, 然后把小的群组合成更大的群。根据结群关系可构造一颗结群树, 再按结群树进行布局。这类算法和基于划分的算法一样, 易于陷入局部最优解。

还有一些基于数学规划的算法, 如 GORDIAN<sup>[3]</sup> 等。这类算法是在分布约束下求解目标函数的最优解。它有一个好的特性, 即可基于严格的数学分析证明它的求解质量。但需求解一系列方程组, 特别是问题规模较大时, 代价是可观的。

此外, 过去还发表了基于模拟退火、遗传算法等随机优化方法的算法。它们是从概率意义上以随机的方式来寻找最优解。上述两种算法各有特点: 遗传算法具有较强的全局搜索能力, 但局部寻优能力差、且易陷入局部最优解; 模拟退火算法具有较强的局部搜索能力, 并且容易避免局部最优解, 但它不适合全局搜索, 特别是当问题规模较大时, 算法运行将变得非常慢。

本文将遗传算法和模拟退火算法有机地结合起来应用于 VLSI 门阵列模式布局设计。

## 1 门阵列模式布局——GA-SA 混合算法

### 1.1 问题描述

VLSI 布局问题的输入有:  $m$  个单元(或模块)集  $M = \{e_1, e_2, \dots, e_m\}$ ,  $n$  个线网集  $N = \{s_1, s_2, \dots, s_n\}$ , 其中  $s_i (1 \leq i \leq n)$  是互连单元的集合, 这些数据都可从综合后的网表中获得; 行数设为  $r$  及每行最大可放单元数设为  $c$ , 一般来说  $r$  和  $c$  只确定一个, 另一个由算法优化确定, 以优化面积, 但因为假设单元都是等宽、等高的, 所以  $r$  和  $c$  都作已知输入。布局就是在满足各项约束条件下安置这  $m$  个单元, 使得目标函数达到最优。其中约束条件主要包括无重叠约束, 可布性约束等。

### 1.2 编码

采用遗传模拟退火算法解决布局问题, 遗传算法不能直接处理问题空间参数, 必须把它们转换成遗传空间的由基因按一定结构组成的染色体或个体。首先要将布局构型编码成字符串, 设芯片中规定行数是  $r$ , 每行可放  $c$  个单元, 即可放入  $m = r * c$  个单元, 所以染色体的长度就应为  $m$ 。为芯片上的每个位置编上号(如图 1 单元左上角的数字), 按照一定的位置顺序将单元号连接起来组成一个染色体<sup>[4,5]</sup>。图 1 表示布局构型, 图 2 表示此布局的编码(基本单元  $e_1$ 、 $e_4$  分别用 1、4 表

**基金项目:** 国家自然科学基金资助项目(60302014); 博士后基金资助项目(20040350578); 合肥工业大学学生创新基金资助项目(XS0404)

**作者简介:** 操礼程(1981 -), 男, 硕士生, 主研方向: VLSI 布图设计, 模拟集成电路设计; 杨依忠, 硕士; 范海秋, 硕士生; 解光军, 博士后、教授

**收稿日期:** 2006-01-02 **E-mail:** gjxie8005@hfut.edu.cn

示，以下类同)。

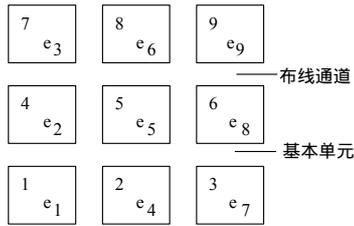


图1 门阵列模式的布局构型

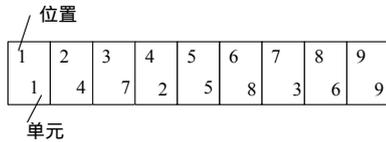


图2 布局编码

### 1.3 适应度函数

遗传算法在进化搜索中基本上不使用外部信息，仅用适应度函数(往往是目标函数)为依据。适应度函数的选取至关重要，直接影响到遗传算法的收敛速度以及能否找到最优解。布局的最终结果是由布线给出，布通率是一重要参数。可以证明线网总长与布通率有较好的对应关系。由于布局在布线之前，那就要在无任何走线的情况下估算线长。估算线长有很多方法，如最小斯坦纳树法、最小生成树法、最小链法、完全图法、二次线长法和半周长法<sup>[6]</sup>等。半周长法估算线网长度，如图3所示，假如一条线网包含图中的4个单元，那么作一个最小的矩形，使它包括这4个单元，就用这个矩形周长的一半来估计这条线网的长度。这种方法十分简单，并且对只有三四个顶点的线网具有较高的估算精度，而集成电路中大部分线网也只有三四个顶点，用此方法具有合理性。这里用复合目标函数

$$Cost = L_{max} * P + \sum_{i=1}^N L_i$$

其中 $N$ 为线网数， $L_i$ 为第 $i$ 条线网的长度， $L_{max}$ 为最长线网的长度， $P$ 为惩罚系数，取其值为单元数。对最长线网的惩罚的引入，其收敛速度比以总线长度为单一目标函数的要快，使得布局构形更趋合理。另外，对于布局问题，若目标函数值越大，对应个体的适应度就越小，这就存在适应度的定标问题，取适应度值  $Fit = -Cost + MaxCost$ 。其中 $MaxCost$ 是一代中的最大目标函数值。

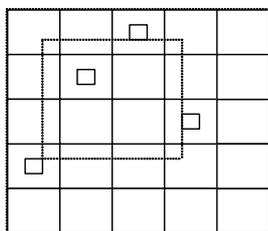


图3 半周长法估算线网长度示意图

### 1.4 选择函数

选择函数对遗传算法性能有很大的影响，它既要使所选的个体具有较高的适应度，又要使个体在解空间中具有较好的分散性。赌轮选择法是GA中最基本也是最常用的选择方法，它也叫适应度比例法或蒙特卡罗选择，在该方法中，各个体的选择概率和其适应度值成比例。我们采用保留最优的个体，使它直接进入下一代，另外 $(n-1)$ 个个体根据赌轮法来选择。

### 1.5 交叉和变异算子

交叉是把两个父个体的部分结构加以替换重组而生成新个体的操作，由于布局问题编码的特殊性，如果直接利用基本遗传算法中的交叉算子，就可能出现非法染色体(如出现了重复单元等)，这在布局中是不允许的。为避免非法染色体问题，利用图4的部分匹配交叉算子<sup>[7]</sup>。

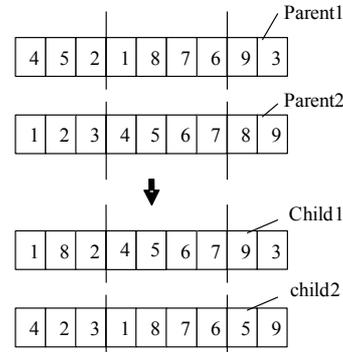
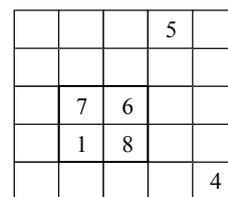
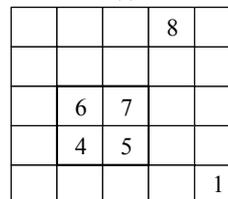


图4 部分匹配交叉策略

首先，将两个交叉点之间的中间段交换，并确定一一对应的映射关系，如  $1 \leftrightarrow 4$ 、 $8 \leftrightarrow 5$ 、 $7 \leftrightarrow 6$ 、 $6 \leftrightarrow 7$ ，然后，首尾两段继承父代，若遇到与中间代码相同的代码，由此代码的映射代替，如中间代码段交换后，parent1中首位的4中间代码段中已含有，用它的映射1代替放入child1的首位，由此类推。最后得到父代parent1和parent2产生的子代child1和child2。在本例中，VLSI布局是在平面内安置单元，如图5所示，确定了交叉块后，由于交叉块中存在不同的单元，若直接交换就会发生非法个体，采用上述的部分匹配交叉策略，实际上交叉的就是图5中标出的单元(其余单元未标)。对于交叉块的确定，采取新的策略，首先确定交叉块(第 $r1$ 行~第 $r2$ 行，第 $c1$ 列~第 $c2$ 列)，让 $r1$ 和 $c1$ 随机确定； $r2$ 和 $c2$ 的选取让它们服从正态分布，使得它们选取中间行和列的概率较大，离中间行和列越远的行和列，被选取的概率就越小。这样选取过大或过小的块的概率变得较小，使得交叉效率得到提高，加快了收敛速度。实验表明这样获得的结果比 $r2$ 和 $c2$ 也随机确定的好。



(a)



(b)

图5 平面交叉策略示意图

进行交叉之后是子代的变异，子个体变量以很小的概率或步长产生转变，变量转变的概率或步长与维数(即变量的个数)成反比，与种群的大小无关。变异本身是一种局部随机搜索，使得GA具有局部搜索能力，另外它还能使GA保持种

群的多样性,防止出现未成熟收敛。为防止变异产生非法个体,采用对交换进行变异,变异概率取为 0.15。

### 1.6 模拟退火操作

遗传算法具有较好的全局搜索能力,但其局部搜索能力较差,易发生早熟,而模拟退火算法具有较好的局部搜索能力,易跳出局部最优解,但全局搜索能力较差,并且它的搜索效果与初始解的情况有很大的关系。我们将遗传算法与模拟退火算法有机地结合起来,对遗传算法每一代进化的最优解(局部最优解)进行一定次数的模拟退火操作,模拟退火的次数与进化代数有关,进化代数越大,模拟退火操作的次数就要越多。算法的伪代码如下:

```

Procedure GA-SA algorithm:
Begin
  Initialize data;
  For i=1 to popsize do
    Randomly generate individual chrom(i);
    Evaluate the fitness:current_cost(i);
    Statistics;
  End for
  While not(terminate condition) do
    For i=1 to popsize
      Select a individual and put it to select(i);
    End for
    For each two selected individual do
      If random<Pcross
        Crossover;
      End if
      If random<Pmutation
        Mutation;
      End if
      Evaluate the fitness:cost;
      Statistics;
    End for
    Applied SA to the local optimized individual;
    Generation:=1;
  End while
End
  
```

## 2 实验结果

我们用 MATLAB 语言在 PC 机上进行程序设计。首先从门级网表提取单元和线网信息,并将其转换成连接矩阵。用到下列数据:单元数 12,线网数 9(其中 5 条线网含 3 个单元,4 条线网含 2 个单元)。图 6 为单元间的连接关系,目标是连线总长最短且其中的最长线也最短。相邻单元间距定为 1。开始随机产生初始种群,图 6(a)为随机产生的一个个体,经过 100 代的进化,得到图 6(b)所示的最优个体。

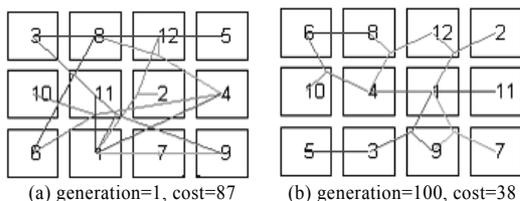


图 6 布局结果状态

图 7 为遗传模拟退火操作进程,从中可看出:大约一直到 50 代时,遗传进程进行得较好,个体的平均代价逐代变小,表明个体按预期进化,在 50 代之后,种群易陷入局部最优解

(早熟),这时模拟退火过程发生了较大的作用,加强了局部搜索。图 8 是基本遗传算法的进程,它较难跳出局部最优解。

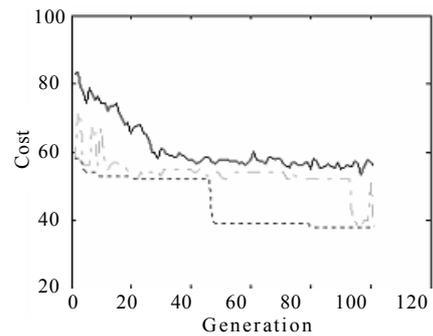


图 7 GA-SA 进程(Aver 实线, Local 点划线, Best 虚线)

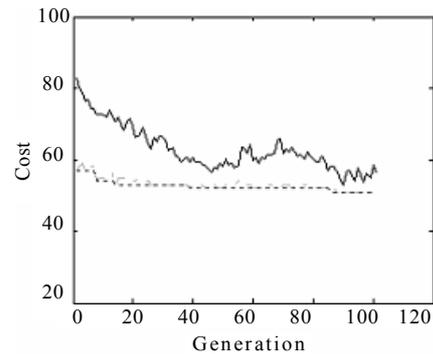


图 8 传统遗传算法进程(Aver 实线, Local 点划线, Best 虚线)

## 3 总结

VLSI 版图设计是一项比较困难的工作,它的复杂度为  $N!$  ( $N$  为单元数),随着  $N$  的增大,其复杂度急剧膨胀,增长速度比指数函数快得多。对于实际问题还要考虑单元大小,那么重叠问题的解决就变得更加复杂,此外还有时序问题,特别是在甚深亚微米阶段,电路对功耗要求特别严格等。所有这些都要求在布局阶段加以考虑。本文只对简单的门阵列模式的布局算法作了初步的研究,得到的结果在布局质量和运行时间等都较传统遗传算法理想。

### 参考文献

- 1 Kernighan B W, Lin S. An Efficient Heuristic Procedure for Partitioning Graphs[J]. The Bell System Technical Journal, 1970, 49(2): 291-307.
- 2 Fiduccia C M, Mattheyses R M. A Linear Time Heuristic for Improving Network Partitions[C]. Proc. of the 19<sup>th</sup> Design Automation Conference, 1982: 175-181.
- 3 Kleinhans J M, Sigl G, Johannes F M, et al. Gordian: VLSI Placement by Quadratic Programming and Slicing Optimization[J]. IEEE Trans. on CAD, 1991, 10(3): 356-365.
- 4 Cohoon J P, Paris W D. Genetic Placement[J]. IEEE Trans. on CAD, 1987, 6(6): 956-964.
- 5 张良震, 刘红, 史亮等. 遗传算法应用于 VLSI 布局的研究[J]. 电路与系统学报, 1999, 4(3): 47-53.
- 6 洪先龙, 严晓浪, 乔长阁. 超大规模集成电路版图理论与算法[M]. 北京: 科学出版社, 1998: 114-118.
- 7 陈国良, 王煦法, 庄镇泉. 遗传算法及其应用[M]. 北京: 人民邮电出版社, 1996: 138-141.