

基于遗传算法的布局问题求解

刘红, 韦穗

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039)

摘要:布局问题是VLSI版图设计中的重要问题,传统的方法很难得到满意的解答。针对该问题,文中提出了一个基于遗传算法的求解方法,并将它应用到VLSI门阵列模式布局设计中,构造了一个较完备的综合布局目标函数,引入并定义了通道拥挤度。实验结果表明,所提出的方法能在较短的时间内提供优化解,为VLSI布局设计提供了一种新的思路。

关键词:遗传算法;门阵列模式布局;通道拥挤度

中图分类号:TP301.6;TN402

文献标识码:A

文章编号:1673-629X(2007)01-0074-02

A Placement Solution Based on Genetic Algorithm

LIU Hong, WEI Sui

(Ministry of Education Key Laboratory of Intelligence Computing and Signal
Processing, Anhui University, Hefei 230039, China)

Abstract: The placement problem is an important problem in VLSI layout design, but results from traditional methods are not satisfied. This paper proposes a genetic algorithm to solve the gate array placement. A proper objective function is constructed. A congestion degrees of channels is put forward and defined. Experiment result is encouraging. The algorithm appears to be a promising method of solving VLSI placement problems.

Key words: genetic algorithm; gate array placement; congestion degrees of channels

0 引言

版图设计是VLSI(Very Large Scale Integrated)芯片设计的关键环节,由于VLSI版图设计的复杂性,版图设计分为布局设计和布线设计。如果把版图设计结果看作是一些带连线接点的模块,那么布局设计的目标就可以描述为:在满足各项电学和工艺要求的条件下,给定连线表,使用计算机按一定规则在芯片上安置单元,对目标函数实现优化。布局是一类NP完全型问题,亦即随着对象 n 的增大,求得最佳解的时间与 n 呈指数关系。

有关布局问题的文献最早出现于1957年,以后随着集成电路技术的飞速发展,研究布局问题的文献也日渐增多。早期的算法思想,一类是基于图论模型和力学模型,如Steinberg分配法、广义力矢量松弛法;另一类着重考虑器件分布的均匀性,如最小截线法(min-cut)。Steinberg分配法亦称最大独立子集算法,其主要思想是:将器件拓扑成点,线网为边,寻找独立子集。由于Steinberg算法只在独立集之间交换元素,而集与集之间不能交换,所以解的好坏很难保证,此外其计算复杂度也较高,所以一般不直接单独用此法布局。广义力矢量松弛法的主要思想是把芯片上每一个单元看作是一个用若干弹簧或橡皮带连

着的模块,寻找系统最小能量状态。该算法的特点是对形状差异不大的布局问题具有速度快、构形好等优点,但对形状差异大的器件布局过程复杂。最小截线法的主要思想是将单元和载体空间分割成块,不断将各块划分为两个新的块,直至每个块中至多只含有一个可移动单元为止。该算法的优点是其目标函数与可布性有很好的对应关系,不足之处是难以获得总体最佳构形,且布局好坏与切割顺序有关,很难保证最后的结果。以上这些算法存在的共同问题是目标函数在计算过程中不能确保收敛于最优解,其解的优劣程度对问题本身有依赖性^[1]。

近年来,随着前沿交叉学科的迅速发展,科学家开始研究用基于热力学模型、生物模型的方法求解复杂的优化问题,跳出了传统的组合算法或人工智能中的确定型启发式搜索算法的基本指导思想,在人工智能领域、土木工程和电力工程中得到应用,并取得了成功。这些算法的最大特点是能在理论上保证其目标函数收敛于全局最优解。最具有代表性的就是遗传算法(Genetic Algorithm, GA)^[2],它不仅在理论上保证其收敛于全局最优解,而且在实时机器人控制、机器视觉和人工神经网络等领域已取得了令人鼓舞的进展。

1 门阵列模式布局问题描述

门阵列设计模式布局有时也称作母片式,利用预先制造的“母片”来进行版图设计。一般的母片上以一定间距

成行、成列地排列着大小、形状相同的基本单元电路。设已知 m 个单元集 $M = \{e_1, e_2, \dots, e_m\}$, n 个线网集 $N = \{s_1, s_2, \dots, s_n\}$, s_i 是互连单元的集合, $1 \leq i \leq n$ 。母片上单元的位置是固定的, 为 k 个元素的集合 $L = \{c_1, c_2, \dots, c_k\}$, $k \geq m$ 。门阵列模式的布局就是在满足各项约束条件下, 在给定的 k 个位置上安置 m 个单元, 使目标函数达到最优。其中约束条件包括电学和工艺要求、布局要求(必须安置每个单元, 又不能使任意两单元重叠)、布线要求。

实际中, 由于布局设计的复杂性, 多数布局设计采用一些近似的易于度量的设计目标来代替实际目标。考虑到布局的最终评价是由布线给出, 亦即所考虑的最重要的参数是布通率。以往布局的目标函数有带权线总长最短、最小截线等。带权线总长和布通率有着较好的对应关系, 但并不成严格反比关系, 即当线长下降至某一程度后, 布通率反而下降。而最小截线法, 虽然注意了布线的均匀性, 但其效果与切割顺序有关, 不能保证解的质量, 这两者都存在片面性, 所以综合这两者, 选择了比较能全面反映布局效果的目标函数 $F^{[3]}$, 它包含了线网总长和垂直、水平通道的拥挤度。

在布线实际完成之前, 对于一个比较复杂的布局问题要算出其实连线长度是根本不可能的。因此常常需要采用近似的计算方法。常用的估算一条线网 i 的连线长度的方法是: 作覆盖该线网所有接点的最小矩形, 令该矩形长与宽分别为 L 和 W , 则该线网连线长的近似解 $\Delta_i = L + W$, 即为矩形的半周长。如图 1 所示的线网, 计算线网中单元的行号最大值 r_{\max} , 列号最大值 c_{\max} , 行号最小值 r_{\min} , 列号最小值 c_{\min} 。线网长为包围这个线网的最小矩形半周长。则此线网长 $\Delta_i = r_{\max} - r_{\min} + c_{\max} - c_{\min} = 4 - 1 + 5 - 1 = 7$ 。

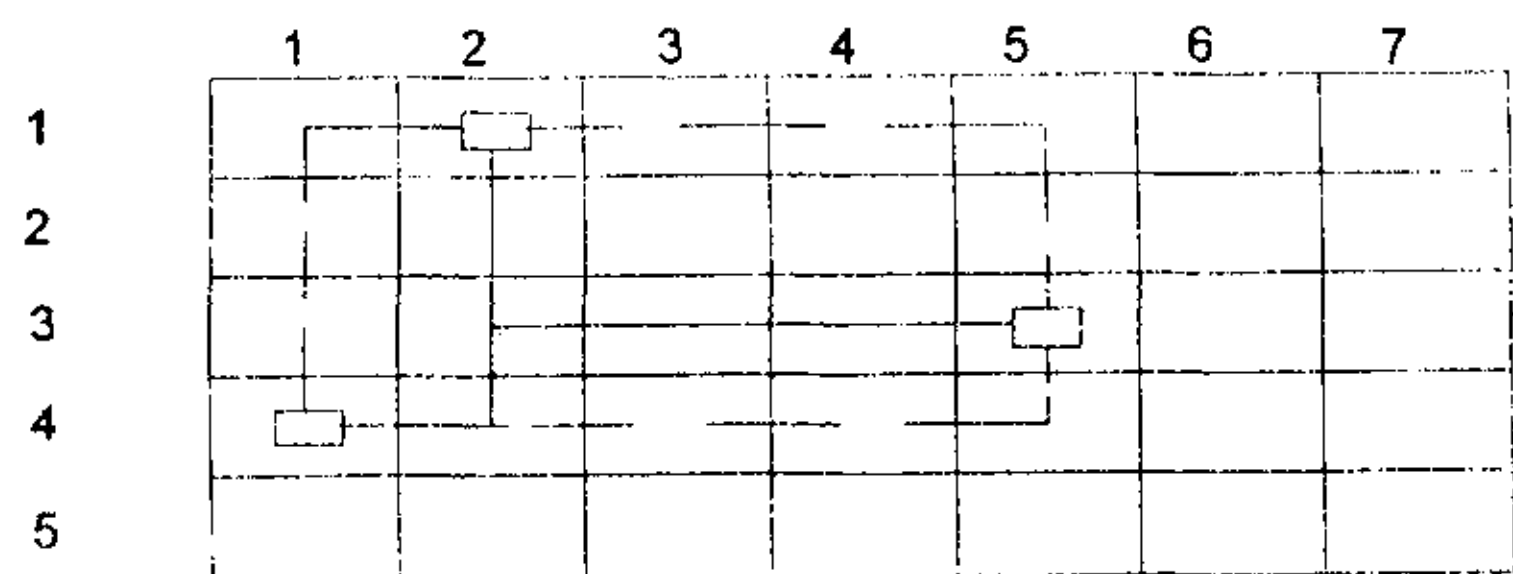


图 1 求线网长度示意图

单元与单元之间的区域是供布线用的水平通道和垂直通道。设 Δ_i 是包围线网 i 的最小矩形的半周长, $1 \leq i \leq n$, 矩形的长和宽分别是它通过的垂直和水平通道数, n 是线网总数。

设 h_j 是线网的最小矩形通过水平通道 j 的线网数, $1 \leq j \leq r - 1$, r 是单元位置的行数。设 v_k 是线网的最小矩形通过垂直通道 k 的线网总数, $1 \leq k \leq c - 1$, c 是单元位置的列数。设 \bar{h} , \bar{v} 是相应于 h_j 和 v_k 的平均值, s_h , s_v 是水平和垂直的标准偏差, 即

$$s_h = \left(\sum_{j=1}^{r-1} \frac{(h_j - \bar{h})^2}{r-2} \right)^{1/2} \quad s_v = \left(\sum_{k=1}^{c-1} \frac{(v_k - \bar{v})^2}{c-2} \right)^{1/2}$$

那么设

$$h'_j = \begin{cases} h_j - \bar{h} - s_h & \text{if } \bar{h} + s_h < h_j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{和 } v'_k = \begin{cases} v_k - \bar{v} - s_v & \text{if } \bar{v} + s_v < v_k \\ 0 & \text{otherwise} \end{cases}$$

定义目标函数 $F = \frac{1}{2} \sum_{i=1}^n \Delta_i + \sum_{j=1}^{r-1} h'^2_j + \sum_{k=1}^{c-1} v'^2_k$, 综合考虑了布线总长和水平通道, 以及垂直通道的拥挤度, 确保布局得到优解。

2 求解门阵列模式布局问题的遗传算法

遗传算法将问题的优化过程与生物进化过程对应起来, 将实际的优化问题编码成字符串, 实际问题的目标函数用字符串的适应函数表示。遗传算法是一个迭代过程, 在最初随机产生一个种群, 按某种指标从中选出一些字符串, 进行交叉和变异运算, 产生新一代的种群。重复此过程, 直至满足某种指标为止。将这些字符串解码还原就可以得到原优化问题的解。当种群中包括的字符串足够多, 迭代次数足够多时, 从理论上讲, 遗传算法可以给出原优化问题的最优解^[4]。用遗传算法求解门阵列模式的布局问题, 首先要把布局构形编码成字符串, 且构造一个能较为精确地反映布局效果的适应函数, 其次要专门设计适用而高效的遗传算子, 如交叉和变异算子^[5]。

设布局问题的解空间为 Π , 构造初始种群 P , P 为 Π 的子空间。根据适应函数 $F: \Pi \rightarrow R^+$, R^+ 为正实数, 计算种群中每个字符串的适应值, 设目标函数是寻找全局最小, 如果 $F(x) < F(y)$, 那么字符串 x 优于 y 。在每一迭代过程中, 选择函数 φ 按交叉率 K_φ 从种群 P 中选出一定数目的字符串作为父串, 进行交叉运算, 产生新字符串集 Offspring。选择函数 ρ 从当前种群 P 和新产生的字符串集 Offspring 中选出一定数目的字符串形成新种群, 每代的种群大小是相等的。新种群中的字符串以概率 K_μ 进行变异。随着迭代次数的增加, 那些优质的字符串在种群中占主导地位, 最后产生出最优解。

遗传算法描述如下所示:

```

P ← construct initial population
while no. of generations less than N do
  Offspring ← Φ
  for k ← 1 to p × Kφ do
    (x, y) ← φ(P)
    Offspring ← Offspring ∪ {φ(x, y)}
  end for
  P ← ρ(P, Offspring)
  For i ← 1 to p × Kμ do
    x ← random(P)
    x ← μ(x)
  end for
end while
return best scoring string in P

```

(下转第 78 页)

属性的匹配对,但 r_1 和 r_2 的距离大于包含 r_1 或 r_2 的其它匹配对的最小距离,则把 (r_1, r_2) 存入 NotReady 集合中。

依据上述思想,得到多数据源匹配算法如图 1 所示。

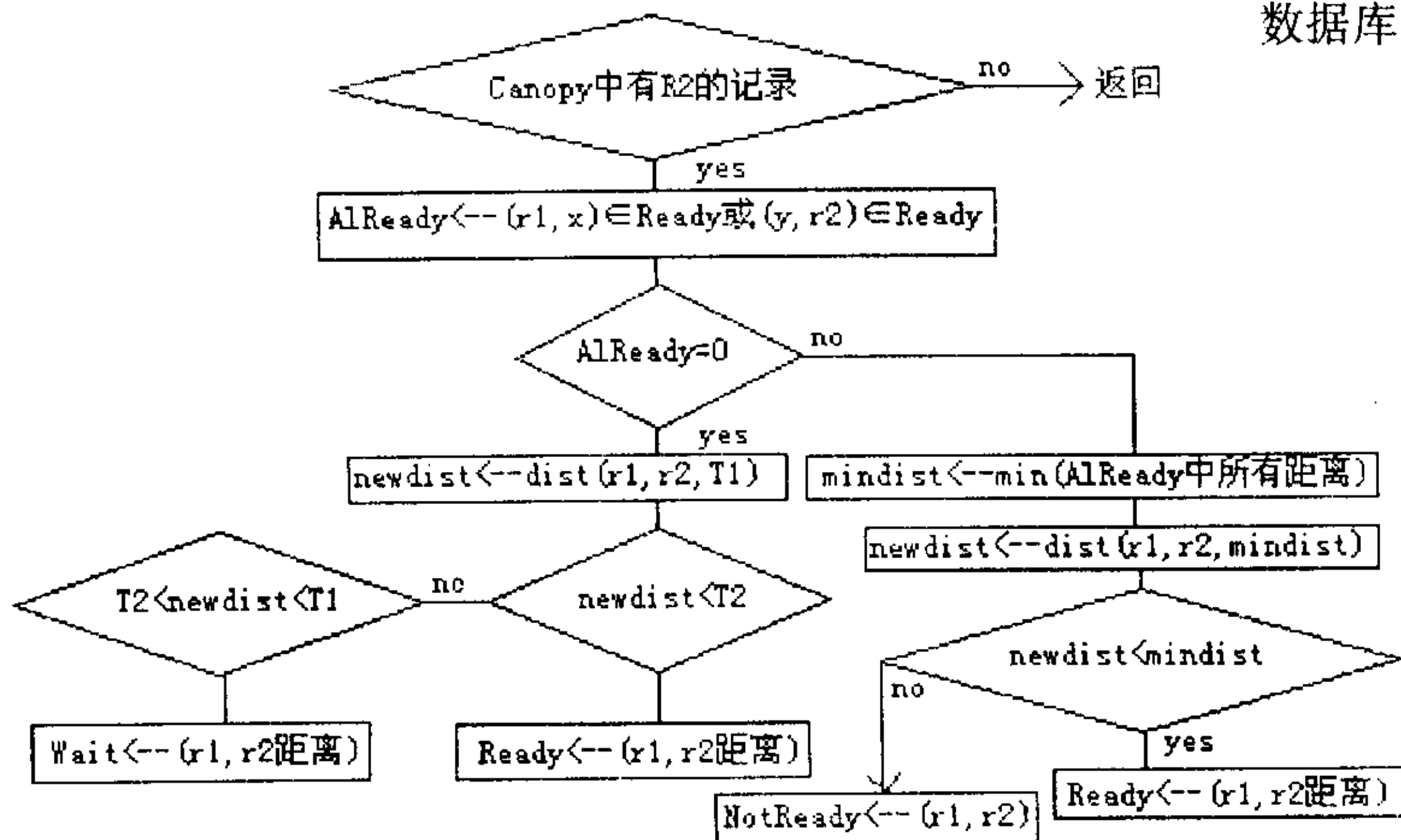


图 1 快速距离阈值法

根据上面的分析,发现匹配的阈值总是改变为与表示同一信息的属性对匹配的最小距离,这就省去了要存储所有的距离然后比较哪个是最小距离的步骤,而且对于编辑距离这种特殊的距离计算,以阈值为界限,一旦距离大于阈值,就不需要再继续计算,而是直接返回 ∞ 。若多数据库系统中,可能存在多个属性都应该与属性 r_1 匹配的情况,那么取以 $\text{mindistance}(r_1, r_2)$ 为中心点,以 ϵ 为邻域的 $\text{dist}(r_1, r') = \text{mindistance}(r_1, r_2) \pm \epsilon$, 为最小匹配距离。算法的这种改进不但保证了原有的准确度,还节省了计算时间和存储空间,提高了匹配的效率。

4 系统实现

采用 CORBA 技术作为多数据库系统运行的底层支持,实现数据库系统的动态加入与注册。当有新的数据库加入到该系统中时,它向运行于系统中的 LDAP 服务器注册本数据库中提供的数据表、字段名以及各字段名的频繁

项的模式,并形成全局数据库模式。

在 LDAP 服务器上运用文中提出的聚类算法输出多数据库系统中所有可能的字段的匹配对,这样每当有新的数据库加入系统中时,系统便会自动识别出与其提供的字段可能的匹配对,客户端便可更新其查询结果。

在实验中,将多个数据库复制多次,并将相同的字段名更改为不同的表示方式,例如,将“地址”这个字段名在不同的复制的数据库中改为“address”、“addr”、“addr1”...“addrn”的形式,再运用分布的查询算法查询有关“地址”的信息时,能够返回“address”、“addr”、“addr1”...“addrn”所在数据库的相关数据集。

5 结束语

运用了针对大型数据聚类的 Canopy 技术对多数据源的字段进行匹配,在保证原有准确程度的前提下,减少了必要的计算量,以及人工介入的过程,提高了字段匹配的效率,实现了数据库的自动发现、自动融合的功能。

参考文献:

- [1] 吕晓华,贾宇波,孙 麒. 一种基于多层模糊模式的频繁项集剪枝算法的优化[J]. 浙江理工大学学报, 2006,23(1): 50-55.
- [2] 胡和平,曾庆锐,路松峰. 中文词聚类研究[J]. 计算机工程与科学, 2006,28(1):122-124.
- [3] 王永成,沈 州,许一震. 改进的多模式匹配算法[J]. 计算机研究与发展, 2002,39(1):55-60.
- [4] 孙晓山,王 强,关 毅,等. 一种改进的 Wu-Manber 多模式匹配算法及应用[J]. 中文信息学报, 2006,20(2):47-52.
- [5] 周 霆,张 伟,张泽洪. 基于关联规则的映射聚类算法[J]. 微电子学与计算机, 2006,23(3):26-33.

(上接第 75 页)

3 实验结果

笔者用软件模拟了一个单元数为 16,线网数为 35,行数为 4,列数为 4 的门阵列模式布局实例。参数的选择和运行结果如表 1 所示。

表 1 运行结果

种群规模	迭代次数	最优值
50	500	81

4 结束语

针对传统布局算法经常收敛于局部最优解的缺点,用遗传算法解决门阵列模式布局问题,提出了一个较完备的综合布局目标函数,改变了传统的单一目标函数,使布局的构形更趋合理。模拟实验表明,应用遗传算法能在较短

的时间内提供优化解,为 VLSI 布局设计提供了一种新的思路。

参考文献:

- [1] 张良震,庄文君. 大规模、超大规模集成电路计算机辅助设计[M]. 北京:电子工业出版社, 1987.
- [2] Holland J. Adaptation in Natural and Artificial Systems[M]. Ann Arbor: University of Michigan Press, 1975.
- [3] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by Simulated Annealing[J]. Science, 1993, (4598):671-680.
- [4] Goldberg D. Genetic Algorithms in Search, Optimization and Machine Learning [J]. Reading, MA: Addison - Wesley, 1989.
- [5] Cohoon J, Paris W. Genetic Placement[J]. IEEE Transactions on Computer - Aided Design, 1987, 6:956-964.